

**Creating 3D computer objects by integrating
multiview range data**

Gérard Blais

B Sc., (University of Waterloo), 1990

Department of Electrical Engineering

McGill University

Montréal

August, 1993

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

© Gérard Blais, 1993

Abstract

The research presented in this thesis deals with the problem of range image registration for the purpose of building surface models of three-dimensional objects. The registration task consists of finding the translation and rotation parameters which properly align overlapping views of the surface so as to reconstruct from these *partial* surfaces, the *complete* surface representation of the object.

The approach taken is to express the registration task as an optimization problem. We define a function which measures the quality of the alignment, produced by a set of motion parameters, between the partial surfaces contained in two range images. The function computes a sum of Euclidean distances between a set of control points on one of the surfaces to corresponding points on the other surface. The strength of the approach resides in the method used to determine point correspondences across range images. The calibration process of the range camera (rangefinder) is reversed, resulting in a set of equations which can be used to directly compute the location of a point in a range image corresponding to an arbitrary point in three-dimensional space.

Two stochastic optimization techniques, Genetic Algorithms (GA) and Very Fast Simulated Reannealing (VFSR), were investigated to carry out the minimization of the cost function. Better registration results and faster convergences are obtained with the VFSR optimizer.

Dual-view registration experiments conducted with the algorithm displayed extremely good results obtained in a very reasonable time. A multi-view registration experiment was also completed with success, except for the fact that a large processing time was required. A complete surface model of a typical 3D object was then constructed from the integration of its multiple partial views. The effectiveness with which registration of range images can be accomplished makes this method attractive for many practical applications where surface models of 3D objects must be constructed.

Résumé

La recherche présentée dans cette thèse traite du problème de correspondance d'images tridimensionnelles pour la construction de modèles de surface d'objets 3D. Le problème de correspondance consiste à trouver des paramètres de translation et de rotation qui alignent correctement des vues chevauchantes d'une surface afin de reconstruire à partir de ces surfaces *partielles*, une représentation de la surface *complète* de l'objet.

Pour ce faire, nous formulons la tâche de correspondance en un problème d'optimisation. Une fonction est définie afin de mesurer la qualité de l'alignement des surfaces partielles de deux images tridimensionnelles. Cet alignement est produit par une instance donnée des paramètres de déplacement. La fonction calcule une somme des distances Euclidiennes existant entre un groupe de points de contrôle sur une des surfaces et leur point correspondant sur l'autre surface. La force de cette approche réside dans la méthode utilisée pour déterminer les correspondances de points entre les images. Le procédé de calibration de la caméra tridimensionnelle est inversé produisant ainsi une série d'équations pouvant être utilisées pour calculer directement la position d'un point, au sein d'une image tridimensionnelle, correspondant à un point arbitraire dans l'espace 3D.

Deux techniques d'optimisation stochastiques furent étudiées pour réaliser l'optimisation de la fonction de qualité: les Algorithmes Génétiques (AG) et *Very Fast Simulated Reannealing* (VFSR). De meilleurs résultats de correspondance ainsi que des convergences plus rapides furent obtenus avec l'utilisation de VFSR.

Des expériences de correspondance de paires de vues réalisées avec l'algorithme produisent d'excellents résultats dans des temps très raisonnables. Hormis la longue durée de temps de traitement nécessaire, une expérience de correspondance de vues multiples fut également réalisée avec succès. Un modèle complet de la surface d'un objet 3D caractéristique est construit à partir de la correspondance de ses multiples vues partielles. L'efficacité avec laquelle cette méthode réussit à accomplir la correspondance d'images tridimensionnelles rend son utilisation très intéressante dans plusieurs applications pratiques, où la construction de modèles de surface d'objets 3D est requise.

Acknowledgments

I wish to thank my supervisor Martin D. Levine for his assistance and guidance throughout the complete realization of my Masters degree. Dr. Levine has made himself available at any time for discussion and I truly appreciate the swiftness and conscientiousness with which he reviewed and edited this thesis. I would like to thank Gilbert Soucy and Pierre Tremblay who have been a tremendous help during the whole duration of my graduate studies and with whom the sacred afternoon *blakes* were very stimulating. Gilbert has been particularly supportive for his pertinent advice and precious expertise in the area I was working on.

I would also like to thank Marc Bolduc, Samir Shah, Gilbert Soucy (once again) and Kenong Wu for proofreading my thesis and for giving me much judicious advice to improve its readability.

I would like to thank Professor Lester Ingber (Lester Ingber Research) for the use of his Very Fast Simulated Annealing optimization program and Kenong Wu for his help in installing and using it. I also wish to thank Kenong Wu for his modeling program which has been used to create the surface model displayed in the last section of the results chapter.

This research has been made possible by an NSERC scholarship and also by the financial support of my supervisor Martin D. Levine to whom I wish to express my gratitude.

Finally, I would like to give my regards to everybody who has supported my endeavour. Especially my family for their continuous support and encouragement throughout my life. I am forever in their debt. I also want to thank Stéphane, Stooz ($\sqrt{-1}$), Burger (666) and Saaarge (00) for enduring me as a roomate and for making the past few years a most enjoyable experience.

Table of Contents

Chapter 1 Introduction	2
1.1 Problem Description	2
1.2 Motivation for Solving the Problem of Range Image Registration	7
1.3 Contributions of this Research	8
1.4 Organization of the Thesis	8
Chapter 2 Literature Review	10
2.1 Open Loop Registration Techniques	11
2.2 Closed Loop Registration Techniques	14
Chapter 3 Registration via Inverse Camera Calibration	21
3.1 Introduction	21
3.2 Formal Problem Definition	22
3.3 Registration Method Overview	25
3.4 Inversion of the Rangefinder Calibration Process	29
3.4.1 Coordinates Evaluation from Rangefinder Calibration	29
3.4.2 Reverse Calibration Process	35
3.4.3 Summary of Calibration Process Inversion	46
3.5 Formulating the Registration Task as an Optimization Problem	46
3.5.1 Cost Function Definition	48
3.5.2 Control Point Selection	54
3.6 Summary	54
Chapter 4 Optimization Methods	57
4.1 Introduction	57
4.2 Shape of the cost function	57
4.3 Genetic Algorithms	65
4.3.1 Chromosome Representation of the Registration Problem	67
4.3.2 Population Initialization	69

4.3.3	Termination Condition	71
4.3.4	Fitness Evaluation	71
4.3.5	Fitness-Proportionate Reproduction	72
4.3.6	Crossover Operator	74
4.3.7	Mutation Operator	75
4.4	Very Fast Simulated Reannealing	78
4.4.1	Overview of Simulated Annealing	78
4.4.2	Boltzmann Annealing	81
4.4.3	Fast Simulated Annealing	84
4.4.4	Very Fast Simulated Reannealing	85
4.5	Summary	92
Chapter 5	Experiments and Results	93
5.1	Experimental Setup and Data Acquisition	93
5.2	Sampling Error of the Laser Rangefinder	95
5.3	Preprocessing of Range Images	100
5.4	Determination of the Initial Transformation Estimate	102
5.5	Time required for a Cost Function Evaluation	105
5.6	Search Control Parameters	106
5.6.1	Search Space and Cost Function Parameters	106
5.6.2	Genetic Algorithm Parameters	109
5.6.3	Very Fast Simulated Annealing Parameters	114
5.6.4	Genetic Algorithm Search Versus Very Fast Simulated Reannealing	120
5.7	Dual-View Registration Experiments	123
5.7.1	View Registration for an Owl Figurine	125
5.7.2	View Registration for a Teapot	126
5.7.3	View Registration for a Metal Pipe	132
5.7.4	View Registration for a Model Car	134
5.7.5	View Registration for Fruits	138
5.7.6	Multi-View Registration Experiment	143
5.8	Summary	154
Chapter 6	Conclusions	157

Bibliography	160
------------------------	-----

List of Figures

1.1	Building of a 3D model by joining separate range views of an object	4
1.2	3D axes and coordinates of a point	5
1.3	Coordinate frame in a range image	6
3.1	Sampling of a surface with the laser rangefinder	27
3.2	Functional Diagram of the Laser Rangefinder	30
3.3	Geometric Relationship between x , y , z and the Rangefinder Parameters	33
3.4	Graph of the TAN calibration table	34
3.5	Graph of the variation of the X-mirror index i with respect to the actual position of the mirror α	34
3.6	Graph of the inverse tangent value of the TAN calibration table with respect to the index j	36
3.7	Linear model of ALPIIA calibration table	37
3.8	Polynomial fit on the slope data and intercept data	39
3.9	rays in the $x\rho$ plane with domain of <i>INVERSE_MB</i> table	44
3.10	Distance Threshold	50
3.11	Point to point correspondence across range images	53
3.12	Fitness Evaluation Process	55
3.13	Control Points for fitness evaluation	56
4.1	Typical example of two range views to be registered	59
4.2	3D slices of the cost function (1)	60
4.3	3D slices of the cost function (2)	61
4.4	3D slices of the cost function (3)	62
4.5	3D slices of the cost function (4)	63
4.6	Basic structure of genetic algorithm	66
4.7	Crossover Operator	75
4.8	Mutation Operator	76
4.9	Boltzmann distribution	82

4.10 Acceptance Probability Function	83
4.11 Cauchy distribution	86
4.12 VFSR Generating Function	88
4.13 Generating Functions for Boltzmann, Cauchy and VFSR	89
4.14 Annealing Schedules for SA, FA and VFSR	90
5.1 Rangefinder / Robot Arm setup	91
5.2 turntable setup	96
5.3 Multiple range views of an object	97
5.4 Distribution curves of the sampling error for various sampling distances	99
5.5 Sampling error as a function of the sampling distance	100
5.6 Background elimination from raw range images	101
5.7 Calibration tool setup on the turntable	103
5.8 Graph of computing time versus number of cost function evaluation	106
5.9 GA convergence as a function of population size	112
5.10 GA convergence for various mutation probabilities	113
5.11 VFSR convergence as a function of the initial temperature	116
5.12 VFSR optimum cost found as a function of the initial temperature	117
5.13 VFSR convergence as a function of the Temperature Ratio Scale	118
5.14 VFSR optimum cost found as a function of the temperature ratio scale	119
5.15 VFSR convergence as a function of the temperature annealing scale	120
5.16 VFSR optimum cost found as a function of the temperature annealing scale	121
5.17 Convergence of GA and VFSR	122
5.18 Two range views of the owl figurine	126
5.19 Initial and final registration of the owl figurine	128
5.20 Coloration of the owl views with gray levels proportional to distance between views	129
5.21 Two range views of the teapot	130
5.22 Initial and final registration of the teapot	132
5.23 Coloration of the teapot views with gray levels proportional to distance between views	133
5.24 Two range views of the metal pipe	134
5.25 Initial and final registration of the pipe	136

5.26 Coloration of the pipe views with gray levels proportional to distance between views	137
5.27 Two range views of the model car	139
5.28 Initial and final registration of the car	141
5.29 Coloration of the car views with gray levels proportional to distance between views	142
5.30 distance due to line of sight correspondence	143
5.31 Two range views of the fruits	144
5.32 Initial and final registration of the fruit	144
5.33 Coloration of the fruit views with gray levels proportional to distance between views	146
5.34 Multiple view registration without global optimization	151
5.35 Multiple view registration with global optimization	151
5.36 Comparison of registration quality between global and pairwise registration of multiple range views	152
5.37 Coloration of the first and last range views showing superiority of global optimization	153
5.38 Integrated 3D model of the owl	155

1.1 Problem Description

How do humans gather and store information about the structure of the world? This is a puzzling question which to this day still evades the scientific community. Over the past three decades, various regions of the visual cortex have been studied and some knowledge about the organization of the visual information has been gained. The pioneering work of researchers like Hubel and Wiesel [10, 11] who, in the early sixties studied the functional architecture of the visual cortex, has sparked a wide interest in the fields of biological and computational vision. Nevertheless, due to the extreme complexity of most biological visual systems, still comparatively very little is known. When dealing with practical vision problems, some people attempt to find solutions by examining how biology copes with similar types of problems. For example, there is evidence of orientation hypercolumns in the visual cortex which are linked with edge detection in the early stages of vision [11]. Based on this hypercolumn model, Parent and Zucker [17] have developed a new approach for detecting edges in intensity images in a manner similar to what has been observed in biology. There are cases, however, where not much is gained by observing biology because too little is known, or simply because the type of problem at hand is not directly related to typical visual tasks accomplished by biological systems. For example, the problem studied in this thesis is the integration of 3D surface information into a model representation. We are dealing with three-dimensional surfaces represented by sets of 3D points and it seems very unlikely that any biological vision system handles this type of data. In such cases, one often tries to find a practical solution to the problem.

The most common approach in vision used for the generation of models of objects consists of obtaining information (an intensity image for example) about the object from a single viewpoint and using this to derive the desired model. This *static vision* approach to modeling is restricted by the limited information contained in a single view. In most cases, data obtained from a single view point around an object provides insufficient information about its overall structure. It is then necessary to gather more information about the object in order to build a valid model. Much research in *active vision* has been initiated in order

to deal with such problems. Contrary to the static modeling approach, where everything is fixed, in active vision the sensors are moved with respect to the observed scene. For the task of 3D object modeling, this permits the acquisition of multiple views of the object and thus a more detailed definition of its overall structure can be obtained.

Some 3D modeling techniques have been used to build a representation of the volume of objects. The modeling of the human fetus in the womb using ultrasound data, where all the internal organs are represented in the model, is one example [31]. The type of model we wish to create in our work is a surface model. We represent a 3D object by its surface. Thus nothing is known about its interior properties. Obviously, in order to create such a model of an object, surface information about the object must be acquired.

To obtain surface information about a scene, the human visual system makes use of the two eyes separated by a small distance and computes the depth from the discrepancies between the images in both eyes. This is the principle of stereo vision. Stereo vision is a very complex process which is an active subject of research but is still not completely understood. Besides the stereo aspect of vision, humans also use optical information such as shadows in order to determine the shape of objects. This has spanned various areas of research into *shape-from-X* techniques. These try to derive surface structure from other surface information (shape-from-shading, shape-from-motion, etc.). A more practical approach for extracting the surface structure and thereby model an object is to obtain the surface information directly using a *rangefinder*. A rangefinder is a special camera which measures the depth of points in the scene rather than measuring the light intensity of points, as a normal CCD camera does. A rangefinder can be viewed as a very precise "radar system". To measure the distance to a given point in a scene, the rangefinder used in this research emits a laser beam towards that point. The light is reflected back to the rangefinder and the distance is computed from triangulation principles (see Section 3.4). An image obtained with the rangefinder is called a *range image* or a *range view*. A range image consists of a set of 3D points representing the surface of the scene being viewed. With the rangefinder used in this research, a range image consists more specifically of a two-dimensional array of 3D points representing the surface of the object.

The approach we take for modeling 3D objects is to obtain multiple range views of an object from various viewing positions (active vision) and then to piece all these views together to form the model. This is similar to a jigsaw puzzle where the different pieces

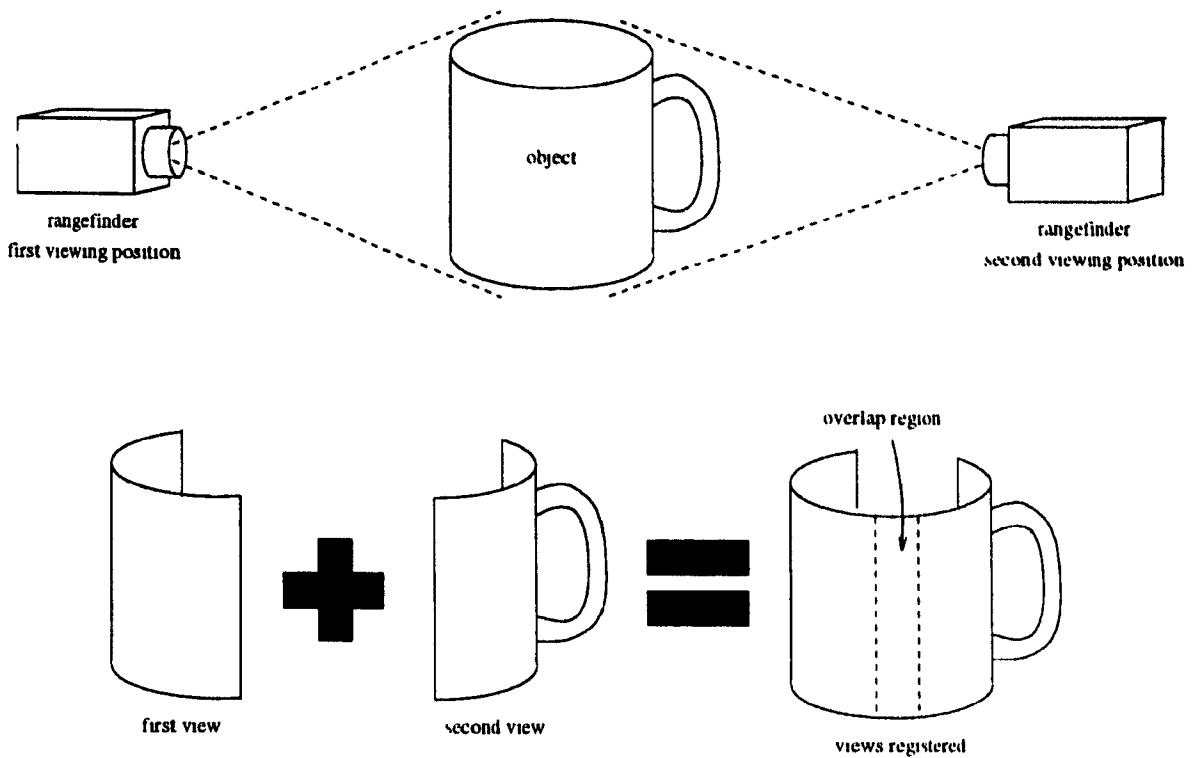


Figure 1.1: This figure illustrates the process of data acquisition and the building of a partial 3D model of a mug by joining two range views. Two separate range views of the object are acquired with a rangefinder. Then, similar to a jigsaw puzzle, the separate pieces of the surface of the mug are put together in order to build the surface model of the object.

are interlocked in order to build a picture, except that here the pieces are tridimensional (Figure 1.1).

Each view acquired with the rangefinder consists of a three-dimensional surface representing a portion of the surface of the object. The task of piecing together two range views is referred to as *registration*. In order to register two range views, one must specify the 3D spatial relationship between them. The relationship establishes how one of the views must be displaced and oriented in order to tightly fit (register) with the other view. This relationship is specified by a transformation. A transformation is usually specified by a translation and rotation; if this is the case it is called a *rigid transformation*. The transformation is applied to all points of one of the two surfaces so that they are translated and rotated to fit with the other surface. A non-rigid transformation is one that does not preserve the original shape of the surface being transformed. Transformations that scale and/or stretch the surface are examples. In this research, the registration of range images is accomplished

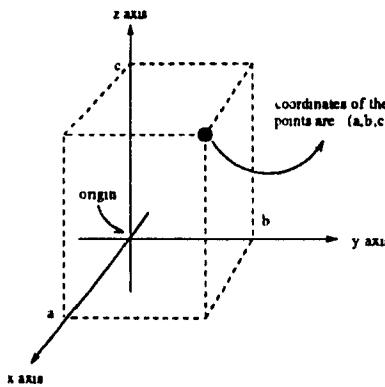


Figure 1.2: The location of a point in 3D space is specified by its coordinates. The coordinates of a point indicate the distance of the point from a fixed origin in each of 3 perpendicular axes. In the figure shown above, the point is at a distance a from the origin along the x axis, b along the y axis and c along the z axis. The coordinates of the point are thus (a, b, c)

by the use of rigid 3D transformations only.

The location of a point in 3D space is specified by its *coordinates*. The coordinates indicate the distance of the point from an arbitrary fixed origin along three axes usually labeled as the x , y and z axes (Figure 1.2). These three axes form what is called a coordinate frame.

When range images are acquired, all points in the range image have coordinates specified with respect to a coordinate frame having its origin at the optical center of the rangefinder and its axes oriented in accordance with the orientation of the rangefinder (Figure 1.3).

Thus, in a range view, the coordinates of the points are expressed with respect to the coordinate frame of the rangefinder. Therefore, when the rangefinder is moved to a new position to acquire a new range image, the location and orientation of this coordinate frame changes. In order to register two range views, the spatial relationship between their respective frames must be specified. This relationship is given by a rigid 3D transformation. In this research, we define rigid 3D transformations by 3 translation parameters and 3 rotation parameters for a total of 6 motion parameters (see Section 3.2). The translation parameters specify by how much one range view must be displaced along each of the three axes of the coordinate frame and the rotation parameters specify by how many degrees it must be rotated around each axis in order to register with the other view.

Usually, the position in space (location and orientation) of the rangefinder with respect to the object is known approximately because a mechanical device is used to move the

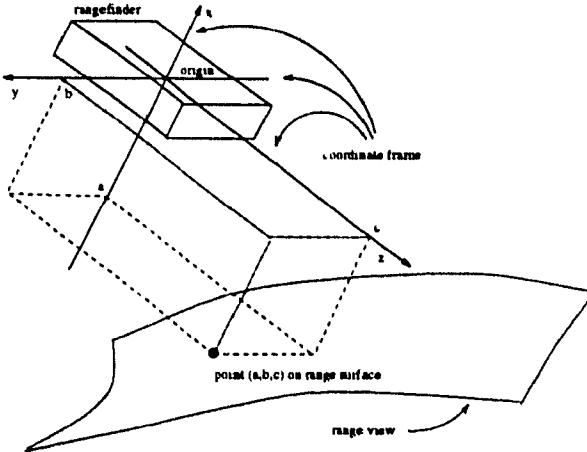


Figure 1.3: The coordinate frame used to specify all points in a range image has as an origin the optical center of the rangefinder and its orientation follows the orientation of the rangefinder. For example, in the figure above point (a, b, c) is a point on the range surface and is thus specified by coordinates related to this frame

rangefinder around the object to acquire different views (or conversely, the object is moved in front of a fixed rangefinder). For example, a robot arm can be used to position the rangefinder. With such a setup, the location and orientation in space of the rangefinder can be determined approximately from the robot joint angles. If the *true* positions of the rangefinder for each view were known, then one could compute the *true* 3D transformation to register them. However, the position of the rangefinder obtained from mechanical devices such as robot arms can sometimes be far from the true position because of mechanical inaccuracies. Consequently, the transformation derived from these estimated positions will be inaccurate. Thus, when registered with this transformation, the range views will not fit properly together and as a result the surface model obtained will not provide a valid representation of the object. It is then necessary to correct the transformation estimate derived during the data acquisition stage.

In order to do this, we make use of the surface information contained in the range images and readjust the transformation estimate until the two surfaces register perfectly. We are in fact correcting the approximate position provided by the positioning devices until the true position of the rangefinder at the time of acquisition is found.

When a transformation relating the frame of a first range view to the frame of a second range view is applied to the points in the first range view, it effectively *transforms* the coordinates of those points to coordinates expressed in the frame of the second view. After transformation the points of both range views are expressed with respect to a unique

coordinate frame, namely the reference frame of the second view.

In order to build the complete surface model of an object, we need to register multiple range views. We therefore need to determine many rigid 3D transformations that will allow us to express all range views in a unique coordinate frame such that all views are perfectly registered. More formally, given N views of an object in a scene, each view describing the 3D structure of the object as seen from a particular point of view, we want to find N rigid motion transformations T_1, T_2, \dots, T_N , that specify the true positions of the rangefinder with respect to a unique frame of reference (arbitrarily chosen and usually the frame of one of the views). Given that each range view i ($i = 1, \dots, N$) consists of a set of 3D points S_i expressed in the coordinate frame of the rangefinder, the transformation T_i transforms the points S_i of range image i into a new set of points $S'_i = T_i(S_i)$ in which the 3D coordinates of the points are expressed in a unique coordinate frame. By transforming the sets of points of all N range views, we can generate a new set of 3D points which are the union of all smaller transformed sets S'_1, S'_2, \dots, S'_N , namely

$$S = \bigcup_{i=1}^N T_i(S_i) = \bigcup_{i=1}^N S'_i$$

This new set of points represents the surface boundary model of the object defined by all the views.

1.2 Motivation for Solving the Problem of Range Image Registration

In many practical applications, it is desirable to generate a three-dimensional model of real world objects. Possible areas in which such a 3D modeling system would be of use are object recognition, robot navigation, CAD input and computer graphics. For object recognition, a 3D modeling system can be used to generate a 3D model from multiple views of an object. Then this model is compared to a database of object models in order to identify it. For robot navigation, one may wish that a mobile robot move about and build up an internal representation of its surrounding environment. A model of the surroundings can then be used by the robot for path planning and collision avoidance. For the purpose of a manufacturing process, it is often desirable to input the shape specifications for an existing part into a CAD system. A 3D modeling system could be used for this task. In certain computer graphics applications a 3D modeling system of the type described in this

thesis could be used for the creation of artificial scenes and objects representing real world objects (i.e. virtual reality).

1.3 Contributions of this Research

The method presented in this thesis provides a novel approach for solving the problem of range image registration. It is a relatively simple method and combines both speed of execution and robustness to noise in the range data and positioning errors in the mechanical apparatus used for data acquisition. The method is suitable for use with an eye-in-hand system where a rangefinder camera is attached to the end of a robot arm.

A cost function indicating the quality of the registration of two range views is defined by a sum of distances between corresponding points in each view. The range views are registered by determining the rigid 3D transformation which minimizes this cost function. The novelty of this registration technique is a method for reversing the calibration process of the rangefinder which allows point correspondences between range views to be computed directly. This results in an extremely fast method for computing the distance between range views as done in the evaluation of the cost function.

A stochastic search technique has been used to find the transformation minimizing the distance between views (minimizing the cost function). This has the advantage of being very robust to noise in the range data and to errors in the initial estimate of the transformation provided by the acquisition system. The minimization of the cost function is accomplished in a very reliable manner, even in the presence of multiple local minima in the cost function.

1.4 Organization of the Thesis

The next chapter (Chapter 2) describes the different approaches which have been developed for solving the range image registration problem. By examining these various approaches, the reader will become more familiar with the registration problem and will gain valuable insight to the subject. This in turn will be useful for appreciating the remaining chapters of this thesis.

In Chapter 3, the essential aspects of our registration method are presented and registration is formulated as an optimization problem. In Chapter 4, two optimization techniques which have been used in the registration are presented. The first of these is based on *Genetic*

Algorithms, and the second is the *Very Fast Simulated Reannealing* optimization technique.

In Chapter 5, the registration method developed in this research is tested. The respective performances of the two optimization techniques presented in Chapter 4 are compared based on their convergence speed and on the quality of the registrations they produce. Registration experiments are then presented.

Finally, in Chapter 6 the positive and negative aspects of the registration method are discussed and areas of possible future improvements are given.

In this chapter, an overview of some of the work that has been done in the fields of range image registration, object modeling and other related topics, will be presented. A review of the research literature is important in the sense that one can gain some insight into the current state of research in the domain of range image registration. An examination of the various registration methods is also necessary for establishing criteria against which our registration method will be compared. This aspect is crucial if one wants to assess the performance of any given method with respect to others. Also, a look at how others have attempted to solve the registration problem allows us to obtain an appreciation for its complexity.

The various methods used for range image registration can be divided in two main categories. The first category consists of methods which avoid the registration problem altogether by relying on precisely calibrated equipment to determine the motion transformation between views. These methods assume that the inter-view transformations provided by the data acquisition apparatus are sufficiently accurate to properly register the range views and do not need to be improved upon. These methods can be viewed as *open loop* systems where the registration transformation provided by the acquisition apparatus is accepted blindly without verifying that it does indeed register the range views correctly. Some of these open loop methods are described in Section 2.1.

The second category consists of methods that derive the registration transformation between range images from the information contained in the range images and other information provided by the acquisition stage. Contrary to the registration methods in the first category, these can be viewed as *closed loop* systems where the transformation parameters are updated and refined gradually until the range views are registered. A feedback function measuring the quality of the registration is used. In most cases, an estimate of the transformation between each pair of range views is part of the available information. However, it is assumed that this transformation is only a coarse approximation of the true registration transformation and that it must be readjusted in order to properly register the range views. These methods are presented in Section 2.2.

2.1 Open Loop Registration Techniques

As indicated previously, many researchers have circumvented the problem of searching for the motion transformation necessary to register two range images by simply relying on accurate and precise hardware. Sakaguchi et al. [24, 25] have used this technique to generate models of 3D object. A fixed rangefinder combined with a turntable is used to acquire range views of objects. Objects are positioned on the turntable and rotated in order to acquire different range views from a fixed viewpoint. By using a turntable to acquire range views, it is quite likely that in some cases surface regions of objects will remain unseen from all angles reached by the turntable. These unseen regions are usually due to self-occlusion. Therefore, in order to complete the surface description of the model, an eye-in-hand acquisition system was also used. The eye-in-hand system consisted of a rangefinder attached to a robot arm. The robot moves the rangefinder to positions where the surface of unseen regions can be acquired. The surface information is integrated and an octree model [5, 8] is generated from the integrated information. In order to unify the data from different range views obtained with the turntable, the range data are transformed according to the rotation angle of the turntable [25]. Thus, it is assumed that the transformation derived from the turntable motion is sufficient for registering the range views. The spatial relationship between the fixed rangefinder and the turntable (and its axis of rotation) must be determined beforehand through calibration. The authors do not discuss how the registration of views obtained with the movable eye-in-hand system are performed. However, similar to the views acquired with the turntable, it is very likely that they used the positional estimate of the robot arm in order to derive the transformation between range views.

Vemuri and Aggarwal [30] have also relied on a calibrated system to obtain inter-frame transformations. They also make use of a turntable (base plane) on which the object to be modeled lies. A pattern is drawn on the base plane which allows the inter-frame transformation to be deduced by observing the orientation of this pattern in an intensity image. Here again, no improvement is made on the transformation obtained from the calibrated instruments. The transformation required to merge range views together is computed directly from the rotation of the turntable and its known spatial relation to the rangefinder. The position and orientation of the turntable and of its axis of rotation with respect to the rangefinder is determined through a system calibration before data acquisition. Multiple

view integration is performed by representing all range views in the same coordinate system and then merging them into a single 3D model. The 3D model is created by expressing the data in a cylindrical coordinate system (θ, r, z) and storing it in a two dimensional array indexed by (θ, z) . The redundant points are averaged. This modeling is only useful for the representation of convex 3D objects. If self-occlusion occurs, some surface information will be missing and as a result holes will be present in the final model.

Potmesil [19] has also done work in the area of 3D object modeling. Similar to Sakaguchi et al., Potmesil used an octree representation for his model. But instead of merging range views, conic volumes generated from silhouettes of the object are merged. The silhouettes are extracted from *intensity* images of the object obtained from different viewpoints. A conic volume is defined by the boundary of a silhouette and the optical center of the camera (which serves as the origin of the conic volume). In order to merge the different conic volumes, their respective position and orientation in space must be known accurately. Potmesil used a camera-calibration technique to find the position of the sensor with respect to some fixed frame of reference. The 3D model of the object, represented in octree form, is constructed by intersecting the conic volumes. The way the volumes are intersected is determined by the motion transformation between the views from which the conic volumes were derived. The silhouettes of the object are acquired using an intensity camera mounted on a robot arm. The perspective projection of the camera maps the 3D homogeneous object coordinate system $O(x, y, z, w)$ into a 2D image plane $I(u, v)$ by a transformation matrix T :

$$\begin{bmatrix} u\epsilon \\ v\epsilon \\ \epsilon \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

The transformation is computed by a camera calibration technique from a set of (at least 6) known 3D points and their corresponding image projections by solving a system of 11 simultaneous linear equations [19]. Given the silhouettes of the object obtained from the intensity images acquired from different viewpoints, the conic volume corresponding to each silhouette is computed using the transformation matrix T . All volumes are then intersected to generate the 3D model of the object. This approach has the same weakness as Venmuri and Aggarwal's with respect to non-convex objects. Because silhouettes are used to generate

the 3D model, inaccurate representations will be obtained if objects with concave regions are modeled.

Sanjay K. Srivasta and Narendra Ahuja [28] have employed a similar method for generating octrees of objects. They also used silhouettes in perspective views of the object to generate 3D volumes which, once intersected together, form the octree of the object. The authors have not addressed the problem of view correspondence, since it was assumed that the absolute position in space, from which a silhouette view is obtained, is known apriori. The authors tested their algorithm with artificial 3D data for which they can directly derive the views from any desired position.

Using artificial data in the same manner, Roth-Tabak and Jain [22] have simulated a system in which a sensor is moved around its environment in order to build an internal representation of its surrounding world. Because they know the exact position and orientation of the sensor with respect to the global coordinate system of the world, they can register all sensor views obtained into a common frame of reference. In a real system, one would have to rely on precise positioning devices or other means for determining the position accurately. So here again, the problem of view registration is avoided by the use of a transformation obtained directly from the system.

The methods presented above can all be considered as simple open loop systems. The motion transformations that are required in order to merge the information obtained about the world to be modeled from different viewpoints are derived directly from calibrated instruments. In many cases the inaccuracy involved in such open loop systems are acceptable for the task at hand, but there are cases where a more precise registration technique is required. One such case is the generation of computer models for CAD systems from existing 3D objects. If further processing or technical analysis is to be performed on a 3D model, this model needs to be very accurate for the analysis to be of any value. As an example, if one were to computationally test the aerodynamic properties of an object whose complete 3D description had been automatically generated by a modeling system, this system would have to render the 3D structure of the object very accurately for the tests to be reliable.

In order to improve on the model accuracy, closed loop systems must be considered. In these types of systems, registration of information between views obtained from different viewpoints is performed by considering information contained in the data gathered with the

sensor. In many cases, calibrated instruments are used to obtain a first approximation to the motion transformation between views. This transformation is then improved upon by techniques utilizing information available about the views and their inter-relation.

2.2 Closed Loop Registration Techniques

Various registration methods make use of the information contained in the views to be registered in order to derive the appropriate motion transformation relating them. These methods can be further divided in two sets. The first consists of registration methods that compare the differences in the structure of the surface across views in order to determine the transformation between them. The comparison is performed over the entire surface or over a set of control points on the surfaces. Typical comparisons between range views are based on the Euclidean distance between the surfaces, the difference and normal directions, curvatures, etc. The second set consists of methods which register range views by matching features from one view with features from the other. Typical features extracted are corners and edges. An overview of some of these techniques is presented in this section.

The approach taken by Potmesil [18] is to compare surface differences between views and to find the rigid 3D transformation which minimizes those differences. A search algorithm is used to improve on an initial estimate of the transformation between two overlapping views. The different views of the object are acquired so as to guarantee that they partially overlap. It is this overlapping region between views that is used to direct the registration search. To construct the model, the surface patches are matched and merged together. The matching algorithm uses a heuristic search to align overlapping surface segments into a common 3D coordinate system. As will be seen, this method is very similar to the one presented in this thesis. The matching process is initiated by selecting a series of control points on the surface of one of the views. A cost function is defined to measure the quality of a registration between views. This cost function, which is assumed to be unimodal, measures the orientation and shape difference between the set of control points in the first view and corresponding points in the second view. The correspondence across views is established by a ray-casting procedure. Starting from a control point, a line is drawn perpendicular to the surface and the intersection point of this line with the surface of the second view defines the corresponding point. The differences computed for the cost function are (1) positional difference, the 3D Euclidean distance from the control point to

the corresponding point on the second view, (2) orientation difference, the angular difference of the surface-normal vectors of the two points, and (3) curvature difference, the magnitude difference of the surface-normal curvatures of the two points. The matching algorithm basically minimizes these differences over all the control points, thus maximizing the shape similarities. Once the inter-frame transformations are found, the range data is represented in a common reference coordinate system and the surfaces are merged. One disadvantage of this approach is that the ray-casting procedure requires that a search be performed on the surface of the second view in order to determine the point correspondence. Although a multiple view registration experiment was conducted, Potmesil does not address the problem of local/global optimization. The views are registered by minimizing a cost function on a one view to one view basis. However, even though the local registration of two views might be good, the overall registration of all the views is usually poor.

Another method for the registration of multiple range images has been devised by Chen and Medioni [4]. Their approach also has many similarities to the one described in this thesis and to the approach taken by Potmesil. Chen and Medioni use an iterative method to register two range images. An initial estimate of the transformation is assumed to be available to guide the registration algorithm. This estimate is used as the starting point of the iterative process. The authors do not explicitly say how they obtain the initial transformation estimate, but mention that it can be obtained from a rotary table or from high level feature matching. The registration algorithm is an iterative process minimizing a presumably unimodal least square error measure. In their approach, Chen and Medioni register range views by minimizing the distance from control points of one view to planes in the other view. The set of control points is selected on the smooth surface regions. For a given control point in view 1, the normal vector to the surface at that point is computed. The control point and the normal define a line. A line-surface intersection algorithm is used to determine the point in the second view which is intersected by the line originating from the first view. This is similar to the way Potmesil [18] establishes correspondences across range views. Once the intersection point is found, its normal vector is computed and the equation of the plane tangent to the surface at that point is determined. The distance measure is defined as the Euclidean distance between the control point in view 1 and the corresponding tangent plane in view 2.

More formally, given two surfaces P and Q , and an initial transformation T^0 , the reg-

istration algorithm attempts to minimize the following error function:

$$e^k = \sum_{i=1}^N d_s^2(T \circ T^{k-1} p_i, S_i^k)$$

where

$$T \circ T^{k-1} = T^k,$$

$S_j^k = \{s | n_{q_j}^k \cdot (q_j^k - s) = 0\}$ is the tangent plane to Q at q_j^k ,

$n_{q_j}^k$ is the normal to surface Q at q_j^k ,

$q_j^k = (T^{k-1} l_i) \cap Q$ is the intersection point of Q with line $T^{k-1} l_i$,

$l_i = \{a | (p_i - a) \times n_{p_i} = 0\}$ is the line normal to P at p_i ,

$p_i \in P$ is a point on P ,

d_s is the signed Euclidean distance from a point to a plane.

The registration algorithm iteratively updates the transformation T to minimize e^k in a least squares sense. The algorithm can be stated as follows [4]:

1. Select a set of control points $p_i \in P (i = 1 \dots N)$ and compute the surface normals n_{p_i} at these points.
2. At each iteration k , repeat the following for each control point p_i , until the process converges:
 - Apply T^{k-1} to both the control point p_i and the normal n_{p_i} to get p'_i and n'_{p_i} .
 - Find the intersection q_i^k of surface Q with the normal line defined by p'_i and n'_{p_i} .
 - Compute the tangent plane S_i^k of Q at q_i^k .
 - Find the transformation T that minimizes e^k with a least squares method.
 - Let $T^k = T \circ T^{k-1}$.

The convergence of the procedure is tested by checking

$$\delta = \frac{\|e^k - e^{k+1}\|}{N'} \leq \epsilon_e, (\epsilon_e > 0)$$

where ϵ_e is a threshold set via experiment. N' ($N' \leq N$) is the number of control points which have an actual point correspondence on surface Q (i.e. their normal line intersects

Q). The authors argue that the nonlinear least squares algorithm used for minimizing e^k can be converted to a linear least squares problem by assuming that the final T^k differs from T^0 by a small amount of rotation.

As indicated earlier, Chen and Medioni's method [4] has strong similarities with the technique described in this thesis. The main difference lies in the fact that a line-surface intersection search algorithm is used in order to find the point in the second view corresponding to a control point in the first view. In our approach, the correspondence is established directly through an inverse calibration function, and instead of finding the point intersected by a perpendicular line to the surface originating at a control point, we find the point intersected by the scan line that sampled that control point during the data acquisition. The advantage of our method is that this intersection point is found extremely fast, computed directly from a set of inverse calibration equations.

Chen and Medioni have used an object-centered representation for their object model. The object is described in a cylindrical or spherical coordinate system, centered in the object. To obtain better overall results when performing global registration, a range image is registered with the merged data from all previously processed views. This way, they avoid the possible error accumulation due to successive registrations with range images of neighboring views.

Szeliski [29] has proposed a registration method which does not require a transformation estimate or knowledge of correspondences between the two views. A motion estimate is obtained by finding the geometric transformation which makes it most likely (in a Bayesian sense) that the points came from the same surface. This method has been designed to work with sparse range data. A spline interpolation of the sparse data is made to complete the surface. The major disadvantage of this approach is that the range data of both views are assumed to be sampled from the same smooth surface. This assumption is very unrealistic from a practical point of view. The method would only work well with very smooth and regular surfaces. This is not the case for the majority of objects in the world, most of which contain edges and many of which have textured surfaces.

The registration method used by Soucy [27] compares surface structures across range views to determine the best registration transformation. The type of registration performed by Soucy is a non-rigid one, meaning that the surfaces are allowed to deform in order to be properly registered. In a typical registration of two range views, a curvature consistency

algorithm is applied to both views separately. The curvature consistency algorithm simply adjusts the 3D position of points in the range image so that the local curvature at each point is consistent to the curvature of the neighbouring points. This makes the curvature field vary smoothly over the whole image and has the effect of smoothing sharp transitions. A smoother range surface is thereby obtained. This is required for a later stage when surfaces patches from one view are fitted to the surface of the other view. The next step in the registration process consists of subdividing the first view into a set of rectangular patches. A few of these patches are then selected for the registration search. The patches that are the central point of the largest regions having the same sign of Gaussian curvature are selected. Usually, 5 to 10 patches are used for the matching process.

The basis of Soucy's approach is the minimization of a functional form that measures the difference between a local neighborhood in one image and a corresponding neighborhood in an adjacent image [27]. Each of the selected patches is then fit onto the surface of the second view. For this purpose, a metric of similarity of surfaces is defined. The augmented Darboux frame is determined for each point of the range surfaces and consists of the point coordinates (\mathbf{p}), the normal vector at that point (N_P), the principal maximum and minimum curvatures (κ_{MP} , κ_{mP}) and their respective direction (M_P , m_P). A difference measurement D_{QT} , between the augmented Darboux frames of a point P_1 in the first view after transformation and a point P_2 in the second view, is defined as follows:

$$D_{QT} = 3 - \langle M_{P'_1}, M_{P_2} \rangle^2 - \langle m_{P'_1}, m_{P_2} \rangle^2 - \langle N_{P'_1}, N_{P_2} \rangle^2 + \frac{|\kappa_{MP'_1} - \kappa_{MP_2}|}{|\kappa_{MP'_1}| + |\kappa_{MP_2}|} + \frac{|\kappa_{mP'_1} - \kappa_{mP_2}|}{|\kappa_{mP'_1}| + |\kappa_{mP_2}|}$$

where $\langle \vec{u}, \vec{v} \rangle$ is the dot product of vector \vec{u} and \vec{v} , P'_1 is point P_1 after transformation, and the augmented Darboux frame of point P_1 is transformed by the rotation matrix \mathbf{Q} and the translation vector \mathbf{T} as follows:

$$\begin{aligned} P'_1 &= \mathbf{Q}P_1 + \mathbf{T} \\ N_{P'_1} &= \mathbf{Q}N_{P_1} \\ M_{P'_1} &= \mathbf{Q}M_{P_1} \\ m_{P'_1} &= \mathbf{Q}m_{P_1} \end{aligned}$$

Note that the maximum and minimum curvatures are not affected by the transformation. A search is performed to find the values of \mathbf{Q} and \mathbf{T} which minimize the functional D and

thus optimally fit the surface patch onto the surface of the second view.

Soucy qualitatively shows that the functional D is rendered convex with a sufficient number of iterations of the curvature consistency algorithm. Then, assuming unimodality and convexity of the functional, a simple gradient descent algorithm is used to find the best motion parameters of each patch. Once all the selected surface patches have been fit to the second view, their motion parameters are propagated to their neighboring patches in the first view. The propagated parameters are adjusted for each patch so that the patch touches the surface of the second view and that the relative rotation minimizes the difference metric. Then these patches propagate their motion parameters to their respective neighbors and so on until all patches have motion parameters assigned to them.

The last stage of the registration consists in the application of a motion consistency algorithm. This algorithm readjusts the relative position and orientations of the patches so that they reflect as closely as possible the relative position and orientations that they had in the original view before the transformation. The positions and orientations of the patches are iteratively updated until the desired level of rigidity in the transformation of the original surface structure is obtained.

This aspect of the method is an advantage in the sense that, by controlling the amount of rigidity in the transformation, one is able to partially compensate for possible distortion introduced by the rangefinder during the acquisition of the range images. The disadvantage of the approach is the amount of computing time necessary to evaluate all extended Darboux frames, and to execute the curvature and motion consistency algorithms.

The methods presented above all compare surface information between views in order to register them. Other methods make use of features extracted from the range surfaces to guide the registration. Cheng and Don [3], for example, use the principle of invariance of distance measures under rigid body motion. They choose a triangle of points $\Delta P_1P_2P_3$ from the first view and then try to find the best matching triangle in the other view. There are other methods which are partially related to the registration problem. The one proposed by Shah and Jain [26] solves the correspondence problem in 2D by matching corners across image frames. Chen [2] has devised a method for determining the pose of an object in a scene based on a known computer model. The sensory data are in the form of lines and the reference model is in the form of planes. To determine the pose of the object, a closed-form solution is found for a set of line-to-plane correspondences. So here again,

2. Literature Review

features are matched in order to determine a motion transformation. The problem with these feature matching techniques is that they must rely on accurate feature extraction prior to the registration and this feature extraction process can be very complex and prone to error.

Sabata and Aggarwal [23] have written a review of the various concepts involved in the problem of estimating motion from a pair of range images. The reader might wish to consult this to examine various registration methods not described here. The method developed as part of this research has the advantages of surface matching algorithms of Potmesil [18] or Chen and Medioni [4] in the sense that no feature extraction step is necessary. However, the point correspondence search problem required in these techniques is eliminated by using inverse camera equations to directly compute the position of corresponding points across range views.

3.1 Introduction

In this chapter, we propose a new approach to deal with the problem of range image registration. The main idea behind this approach is to invert the calibration process of a rangefinder camera in order to establish a point to point correspondence across range views. What is meant by inverting the calibration process will soon be explained in detail. Using this definition of correspondence between range views, the registration task is then formulated as an optimization problem. A cost function measuring the registration quality of individual transformations is defined and an optimization search technique is used to find the 3D rigid motion transformation that minimizes it.

A similar idea has been used by Cheng and Medioni [4]. They find the registration transformation by iteratively minimizing the distance between two range views. They perform the minimization of a sum of distances between control points in one view and corresponding points in the other view. To determine point correspondences, a search procedure is used to find the intersection of a line with the range surface. This is a very lengthy process since a search must be carried out to find each point correspondence. The major advantage of the method presented here is that it eliminates this search by directly computing the location of the most likely correspondence point. This results in an extremely fast, minimal computing time metric for the evaluation of inter-surface distance. This metric is the basis for the cost function which is optimized.

A formal definition of the registration problem is given in Section 3.2. Following this, Section 3.3 gives a brief description of the registration method. The section highlights the major aspects of the method and the reader should gain an overall understanding of the underlying concepts on which this new approach is based. The different aspects of the method are then described in greater detail in the subsequent sections of the chapter.

The first aspect of the method examined is the inversion of the laser rangefinder calibration process. This is presented in Section 3.4. The use of calibration for computing the 3D coordinates of a point in space from a depth measurement is briefly explained. The technique used to perform the inversion of the calibration process is described and we ex-

plain how correspondences between 3D points across range views are established using this inversion procedure. Following this, in Section 3.5, a formulation of the registration task as an optimization problem is given. Section 3.5.2 discusses the control points selection process.

3.2 Formal Problem Definition

The problem of range image registration can be stated as follows: given two range images of a scene with distinct viewing positions and orientations, find the relative camera displacement and rotation in space which relate the two views.

In order to get a range image, a special camera is required. The camera used is called a laser rangefinder. A normal camera is used to measure the intensity of light reflecting from the surface of a scene. The rangefinder camera is used to measure the distance of the surface of a scene to the camera. A laser beam is used to scan the surface of the scene and the distance is measured for different locations on the surface. The 3D coordinates of the sampled points are determined from these distance measurements. The set of sampled points thus acquired represent the scene's surface and form a *range image*.

In a range image, the 3D coordinates of the sampled points are expressed with respect to the camera's reference frame and not with respect to the world's reference frame. This means that the absolute positions of the sampled points in space are unknown. What is known is their relative position with respect to the camera at the instant of acquisition. Another way to think of range image registration is a process by which a relative transformation is found to express 3D points representing the surface of a scene obtained from different camera positions and orientations and hence expressed in different coordinate frames, into a unique coordinate frame.

Some registration techniques allow some distortion of the range images in order to account for distortion introduced in the data acquisition stage. The registration technique developed by Soucy [27] is one example. The method presented here does not distort range images in order to achieve their proper registration. Only rigid motion transformations in 3D space are considered. It is therefore assumed that the rangefinder camera used to acquire data introduces negligible distortion in the images.

We can now define the registration task more formally. Two views of a scene are acquired with the laser rangefinder camera. The views are taken from different positions and

orientations of the camera in space. The 3D points resulting from sampling the surface of the scene in view 1 are all expressed with respect to the local frame of reference of view 1. The same holds for 3D points in view 2 and the local reference frame of view 2. The goal of the registration is to project all 3D points contained in both views into a single reference frame. In order to accomplish this however, the relative motion of the camera from view 1 to view 2 must be known. Since we are only considering rigid 3D transformations, the relative motion can be specified by a 3D translation and a 3D rotation in space. Six parameters are necessary to specify the translation and rotation, three translation parameters t_x , t_y and t_z , and three rotation parameters θ_x , θ_y and θ_z . We can write the transformation T as follows:

$$T = \begin{pmatrix} t_x \\ t_y \\ t_z \\ r_x \\ r_y \\ r_z \end{pmatrix}$$

The translation can be regarded as the distance that the camera had to travel in order to go from its position in view 1 to its new position in view 2. The rotation can be regarded as the amount by which the camera had to turn around each of the three axes in order to change its orientation from view 1 to view 2. We must determine the transformation T so that points in both views can be expressed in a common reference frame.

To do this, we can make frame 2 (arbitrarily chosen) the reference frame. Thus all of the 3D points in view 2 need not be transformed since they are already expressed in the common frame. We must determine how to take a point in view 1 and express it in the new reference frame.

From elementary linear algebra, the 3D translation and rotation of a vector \vec{p} can be done as follows:

$$\vec{p}' = \mathbf{R}\vec{p} + \vec{d} \quad (3.1)$$

where \vec{p}' is the transformed vector, \mathbf{R} is a 3D rotation matrix and \vec{d} is a 3D displacement vector.

In order for Equation (3.1) to represent the transformation of a point expressed in view 1's frame to a point in view 2's frame, we let the translation parameters represent the displacement of the origin from frame 1 to frame 2 and let the rotation parameters represent the change in orientation from frame 1 to frame 2. Hence we have

$$\vec{d} = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} \quad (3.2)$$

We define the rotational components of T , r_x , r_y and r_z , as the Euler angles of rotation around the x , y and z axes, respectively. With angles expressed in radians, we define three rotation matrices \mathbf{R}_x , \mathbf{R}_y and \mathbf{R}_z to perform 3D vector rotations around the x , y and z axes, respectively:

$$\mathbf{R}_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos r_x & -\sin r_x \\ 0 & \sin r_x & \cos r_x \end{pmatrix}$$

$$\mathbf{R}_y = \begin{pmatrix} \cos r_y & 0 & \sin r_y \\ 0 & 1 & 0 \\ -\sin r_y & 0 & \cos r_y \end{pmatrix}$$

$$\mathbf{R}_z = \begin{pmatrix} \cos r_z & -\sin r_z & 0 \\ \sin r_z & \cos r_z & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

We can then define the rotation matrix \mathbf{R} representing the rotation that a vector expressed in view 1's coordinate frame must undergo in order to express it in view 2's frame. The rotations around the z , y and x axes are applied in sequence yielding:

$$\mathbf{R} = \mathbf{R}_x \mathbf{R}_y \mathbf{R}_z = \begin{pmatrix} c_y c_z & -c_y s_z & s_y \\ c_x s_z + s_x s_y c_z & c_x c_z - s_x s_y s_z & -s_x c_y \\ s_x s_z - c_x s_y c_z & s_x c_z + c_x s_y s_z & c_x c_y \end{pmatrix} \quad (3.3)$$

where

$$\begin{aligned}s_x &= \sin r_x & s_y &= \sin r_y & s_z &= \sin r_z \\c_x &= \cos r_x & c_y &= \cos r_y & c_z &= \cos r_z\end{aligned}$$

Now, given a 3D point \vec{p} expressed in view 1's coordinate reference frame, we can properly define the transformation operation $T(\vec{p})$. $T(\vec{p})$ is simply the point \vec{p} expressed in view 2's reference frame, and can be viewed as the vector resulting from rotating \vec{p} by R and translating it by \vec{d} .

$$T(\vec{p}) = R\vec{p} + \vec{d} \quad (3.4)$$

The goal of the registration problem is, therefore, to find the six rigid motion parameters $t_x, t_y, t_z, r_x, r_y, r_z$ which relate the two camera positions in world coordinates. This information is embedded in the structure of the scene's surface in each range view. The relative displacement between view 1 and view 2 results in different regions of the scene being acquired for each range image. If the two views overlap in space, some of the 3D data in each range image will be common to both. Hence, the relative displacement can be derived from this common information. The registration of the range views consists in finding the transformation which, when applied to the 3D data of one range image, makes the data common to both views exactly coincide. The task which remains is to determine which parts of the range views are common to both. Some registration methods proceed by extracting features from each 3D view and comparing features across views. In the method presented in this thesis, we handle this problem by defining a distance measure between the two views. When the proper relative transformation is applied to a view such that the overlapping parts of both views are made to coincide, this distance measure will report a minimum value.

3.3 Registration Method Overview

The rangefinder used for this research allows a two-dimensional sweep of the surface of a scene. When a range view of a scene is acquired, the information given by the rangefinder camera consists of a rectangular array of depth values of points sampled on the surface of the scene. In order to derive the 3D coordinates of each sampled surface point, the rangefinder

is calibrated before data acquisition so that given the index i and j in the rectangular array for a given point, and given the depth measured for that point, the coordinates (x, y, z) of that point with respect to the camera's reference frame can be computed directly. If one thinks of each image point as being sampled by a different laser ray, then the index i and j would specify which ray sampled each point. Figure 3.1 shows how a surface is sampled by the rangefinder. The data contained in the range image array consists of the z coordinate values of all points sampled by the rangefinder.

During range data acquisition, the *exact* relative displacement and rotation of the camera is unknown. This is usually the result of mechanical inaccuracies in the devices used to move the camera with respect to the scene. For example, the camera can be mounted on the end effector of a robot arm. Due to joint backlash and other mechanical errors, the true position and orientation of the camera is only known approximately. Even though the exact relative displacement of the camera between two viewing positions cannot be determined exactly, an estimate can usually be made. It is therefore assumed that an estimate of the transformation T is available as a starting point for the registration process. Consider two range views for which we possess an estimate T_e of the transformation relating them. The goal is to refine this transformation using the data acquired from the surface of the scene, at the two positions and orientations in space.

The principal idea behind this method is to reverse the process by which the coordinates of the point are computed through calibration. This inverse calibration permits us to match points across range views. Given a transformation T from range image 1 to range image 2, a 3D point (x, y, z) in range image 1 is transformed to (x', y', z') into image 2's reference frame. Using the inverse calibration we are able to determine directly the indices (i, j) of the ray in image 2 closest (Euclidean distance) to that transformed 3D point. Every point in a range image is obtained by sampling the surface with a different ray; and the point in image 2 associated with the (i, j) ray is thus taken as the correspondence.

To perform the registration, control points are selected from the first range view. These control points are mapped by a rigid 3D transformation T into the second view's reference frame. Each transformed control point is then associated to a point in the other view. This point to point correspondence is directly established through the inverse calibration process. A distance measure based on a sum of Euclidean distances between the transformed control points of the first view and their respective corresponding points in the second view

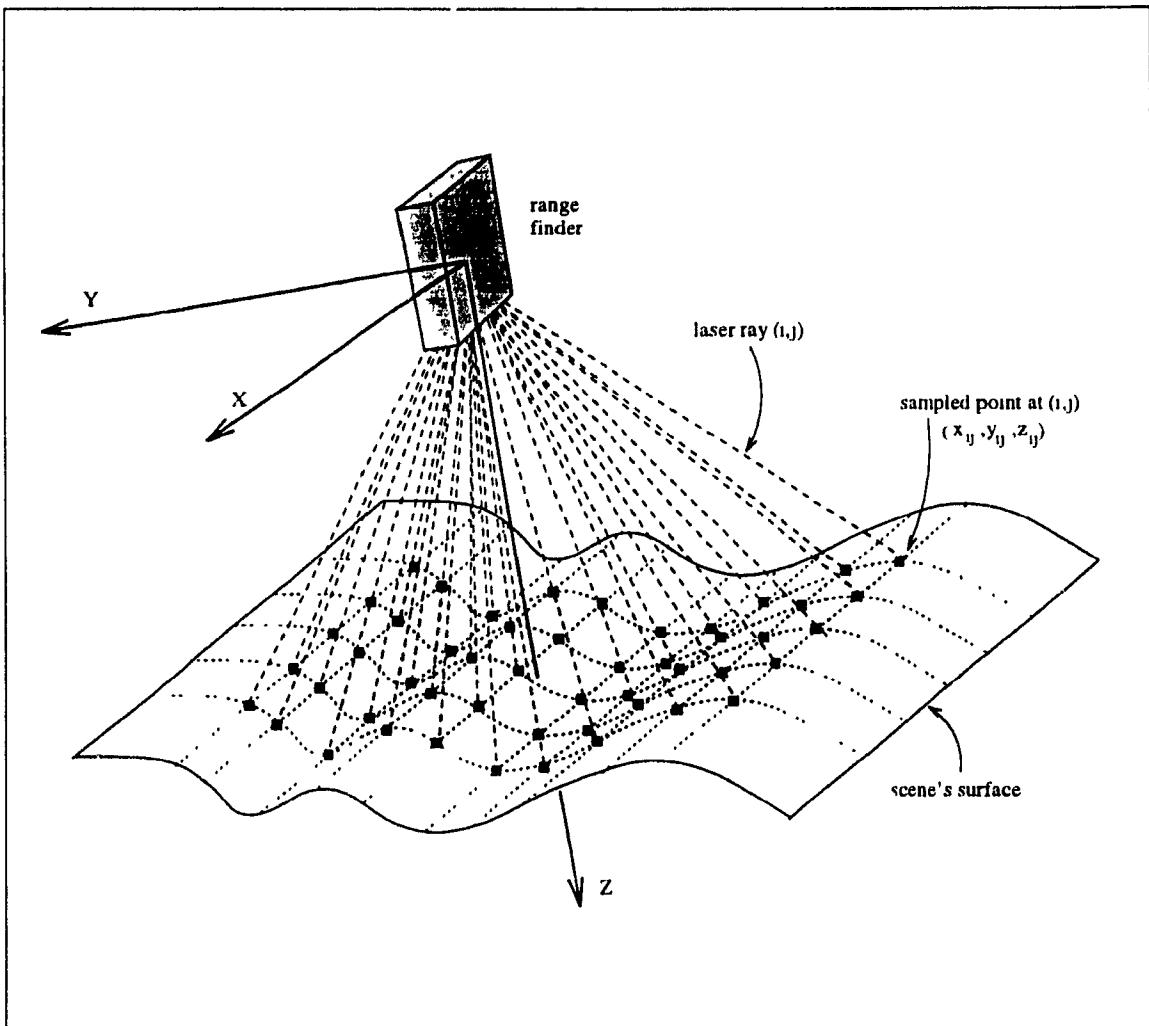


Figure 3.1: This figure shows how the rangefinder camera gathers information about the surface of a scene. The surface is sampled in a grid-like fashion. The data obtained is stored in a two-dimensional array. The values stored in the array are the z coordinate values of all sampled points. The indices of the array represent the particular ray in the 2D scanning grid which sampled the given point. The value z_{ij} would thus be stored in location (i, j) of the range image array.

is computed. The goal of the registration is then to find the transformation T which minimizes this distance measure. The transformation estimate T_e is used to constrain the number of possible transformations. A finite search space is delimited around the estimate T_e and only transformations inside this search space are considered as potential solutions for the registration. The inverse calibration process is detailed in Section 3.4 and control point selection in Section 3.5.2.

By minimizing the sum of Euclidean distances between all control points in one view and their respective corresponding points in the other view, the distance between the views is minimized. Since the sum of distances is at a minimum when surface regions that are common to both views coincide, we can conclude that the views are registered. The sum of Euclidean distances is the basis for a cost function used by an optimization algorithm. This cost function will be described in detail in Section 3.5.

Note that in order for a proper registration of range views to take place it is essential that there be an overlapping region between the two views.

We now summarize the various aspects of this method:

- The object of the view correspondence search is to find a rigid motion transformation $T = (t_x, t_y, t_z, r_x, r_y, r_z)$ which best registers the views.
- The rangefinder used for this research produces range images consisting of two-dimensional arrays of depth values.
- The rangefinder is calibrated so that, given the indices i and j in the array of sampled points and the depth measured for the sampled point, the x , y and z coordinates of the corresponding 3D point can be derived.
- The calibration process can be reversed. That is, given the coordinates x , y and z of a 3D point, the index i and j in the range image can be found.
- This reverse calibration process is a means of establishing point-to-point correspondences across views.
- The transformation is found by doing a search in parameter space for a transformation T in the vicinity of a transformation estimate T_e .
- The goal of the search is to find the transformation T which minimizes the sum of

Euclidean distances between transformed control points of one view to the corresponding points in the other view, the correspondence being established with the inverse calibration.

- The sum of distances between transformed control points in one view and corresponding points in the other view is used as a cost function in an optimization algorithm.
- The algorithm searches for the transformation that minimizes the distance between the views and thus registers the views in the best way possible.

3.4 Inversion of the Rangefinder Calibration Process

The rangefinder used for data acquisition in this research utilizes the principle of triangulation in order to determine the distance of a given point on a surface being sampled. The rangefinder was developed through a joint effort between McGill University and the National Research Council of Canada (NRCC). Many rangefinders are limited in the sense that they can only sample the distance to a surface at a single point or a single line of points on the surface. This rangefinder uses a set of mirrors to allow it to sample a surface along a two-dimensional grid of points. The advantage is clear. A description of the surface under observation can be obtained directly with a single scan, without the need to displace the surface or the rangefinder. Figure 3.2 illustrates the basic construction of the rangefinder. More information on this type of camera can be found in [15].

In this section, the use of the rangefinder calibration is briefly explained. The process by which the actual calibration of the rangefinder is accomplished is not discussed. A description of the rangefinder calibration process is given in [1]. How the coordinates (x, y, z) of sampled points are determined using calibration equations and calibration tables will be briefly examined. The inversion of the calibration equations and calibration tables will then be explained. A brief summary of the calibration process inversion is given at the end of the section.

3.4.1 Coordinates Evaluation from Rangefinder Calibration

In order to determine the actual 3D coordinates (x, y, z) in space of a sampled point, the rangefinder is first calibrated. How this calibration is performed will not be discussed in this thesis (see [1] for details). The essential components resulting from this calibration are

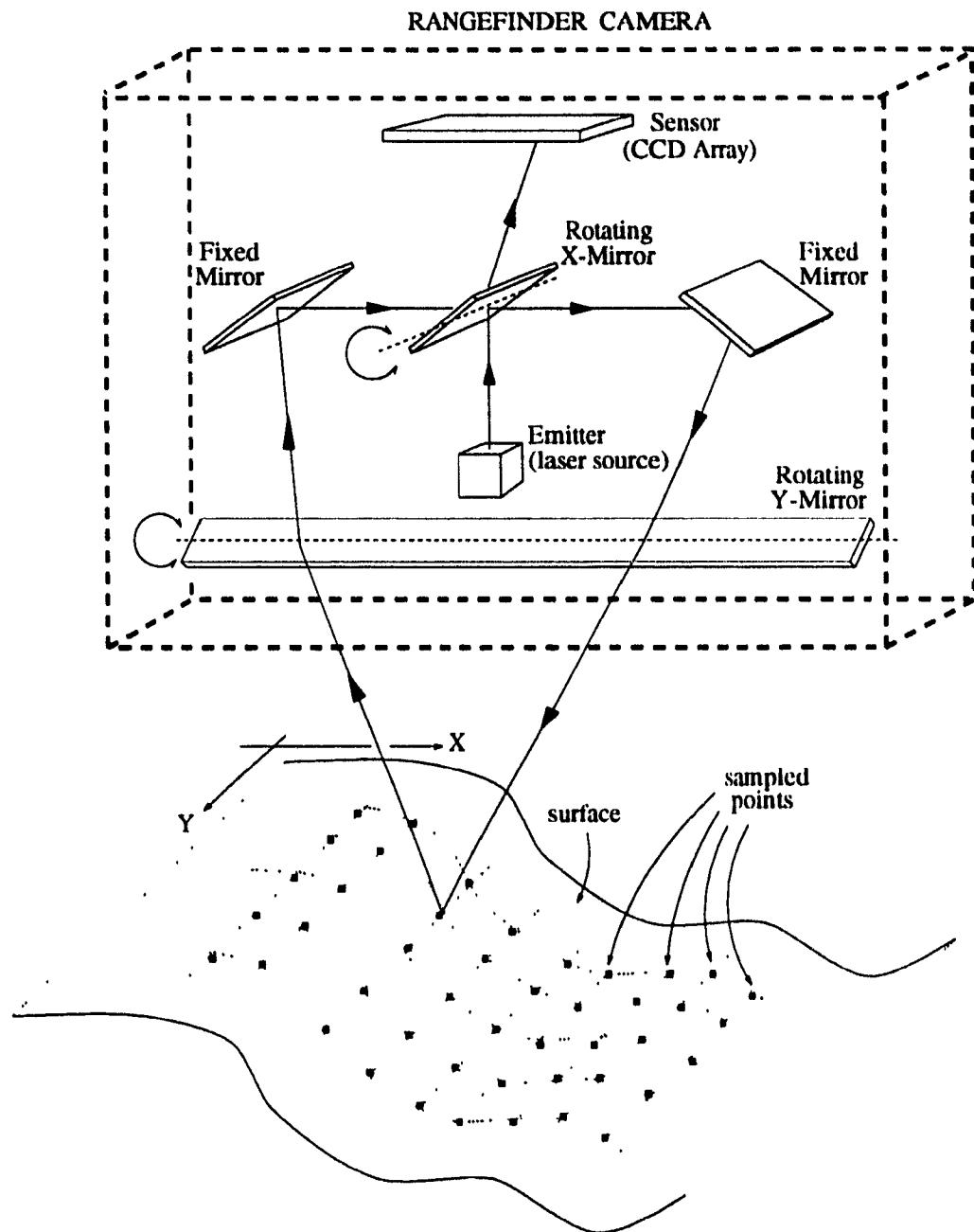


Figure 3.2: This diagram shows the basic components that make up the NRCC/McGill laser rangefinder. The setup consists of two fixed and two rotating mirrors. The rotating mirrors determine the position of the laser beam which hits the surface being scanned. The beam hits the surface and is reflected back through the camera's mirrors into a light sensor which consists of a simple CCD array. Depending on the distance of the camera to the surface at a sample point, the laser beam hits a different location on the sensor. The position of the rotating mirrors as well as the impact location on the sensor determines, through triangulation principles, the distance between the sampled point on the surface and the camera. By changing the orientation of the rotating mirrors, a two-dimensional grid of points can be sampled on the surface.

3. Registration via Inverse Camera Calibration

a number of look-up tables and a set of equations. These are used to compute the x , y and z coordinates of a sampled point expressed in the rangefinder's frame of reference. The coordinates are obtained from the measured depth z of the point to the rangefinder and from the positions i and j of the two rotating mirrors (X-mirror and Y-mirror).

In order to compute the (x, y, z) coordinates of a sampled point, four look-up tables and four equations are required. The information available from the rangefinder camera is: the distance z in the camera's reference frame computed through triangulation and the rotational position of the X-mirror and Y-mirror. The two-dimensional array of distances that constitute a range image may contain a maximum of 256 by 256 sampled points. In other words, both the X-mirror and the Y-mirror can be positioned at a total of 256 discrete positions in their range of possible orientations. An index is assigned to describe each mirror's discrete position. Index i is used to describe the position of the X-mirror and index j is used to describe the position of the Y-mirror. The indices can take the values from 1 to 256.

The steps taken to compute the coordinates (x, y, z) of a point, given the value of z and of the indices i and j are as follows:

1.

$$y = z \ TAN[j] \quad (3.5)$$

2.

$$\rho = \sqrt{y^2 + z^2} \quad (3.6)$$

3.

$$\alpha = ALPHA[i] \quad (3.7)$$

4.

$$x = \rho M[\alpha] + B[\alpha] \quad (3.8)$$

In the above equations, TAN , $ALPHA$, M and B are all calibration tables. Each is a look-up table containing 256 entries. These tables are generated only once during the calibration of the rangefinder camera. Figure 3.3 illustrates the relationship between the values of x , y , z , ρ , i , j and the TAN , M and B look-up tables. The TAN table used in Equation (3.5), as its name indicates, contains the tangent value for the orientation angle of the Y-

mirror. Entries in the table correspond to each of the 256 possible orientations of the mirror specified by the j index. Figure 3.4 shows a plot of the table entries versus the Y-mirror index j .

From Equation (3.6) and Figure 3.3 it can be seen that ρ is simply the length of the line segment generated when the ray that sampled the point is projected onto the yz plane.

The *ALPHA* table reflects small mechanical inaccuracies associated with the galvometer actuating the X-mirror. When the mirror is instructed to rotate to its next discrete position, the mirror index i is adjusted, but sometimes the mirror skips and advances by more than one position or sometimes does not move at all and remains at the current position. As a consequence, the i index value does not always correspond to the actual position of the X-mirror. The *ALPHA* table is built by performing many scans by rotating the X-mirror and determining where the mirror skips and stalls on average. The rangefinder function for the X-mirror defined by the *ALPHA* table can be seen in figure 3.5. From this graph, one can see for example that the mirror gets stuck in a fixed position during the first and last few steps that it executes. The *ALPHA* table is used to compensate for this mechanical problem.

Once α , the true index position of the mirror, is determined with the *ALPHA* table, Equation (3.8) yields the value of the x coordinate of the sampled point. This equation represents a line in the $x\rho$ plane having $B[\alpha]$ as its x-intercept and $M[\alpha]$ as its slope. This line actually corresponds to the ray that the laser beam followed when it sampled the point on the surface. For different positions of the X-mirror, hence for different values of α , the slope and intercept corresponding to the ray vary. In the ideal case, all rays in the $x\rho$ plane would originate from a unique point inside the rangefinder camera. Also, all rays would be evenly distributed so that the angle between consecutive rays would be the same. If this were the case, computing x from the X-mirror index i could be accomplished with a straightforward equation. The M and B calibration tables compensate for the fact that the rangefinder is not ideal. All of the rays do not originate from a unique virtual point in the rangefinder, and the rays are not distributed at exactly equal angle intervals. Knowing which ray sampled a given point (ie. knowing α) and knowing the value of ρ , we use the equation of that ray specified by the M and B tables to solve for x . We find the value of x such that (x, ρ) satisfies the equation of the line with slope $M[\alpha]$ and intercept $B[\alpha]$.

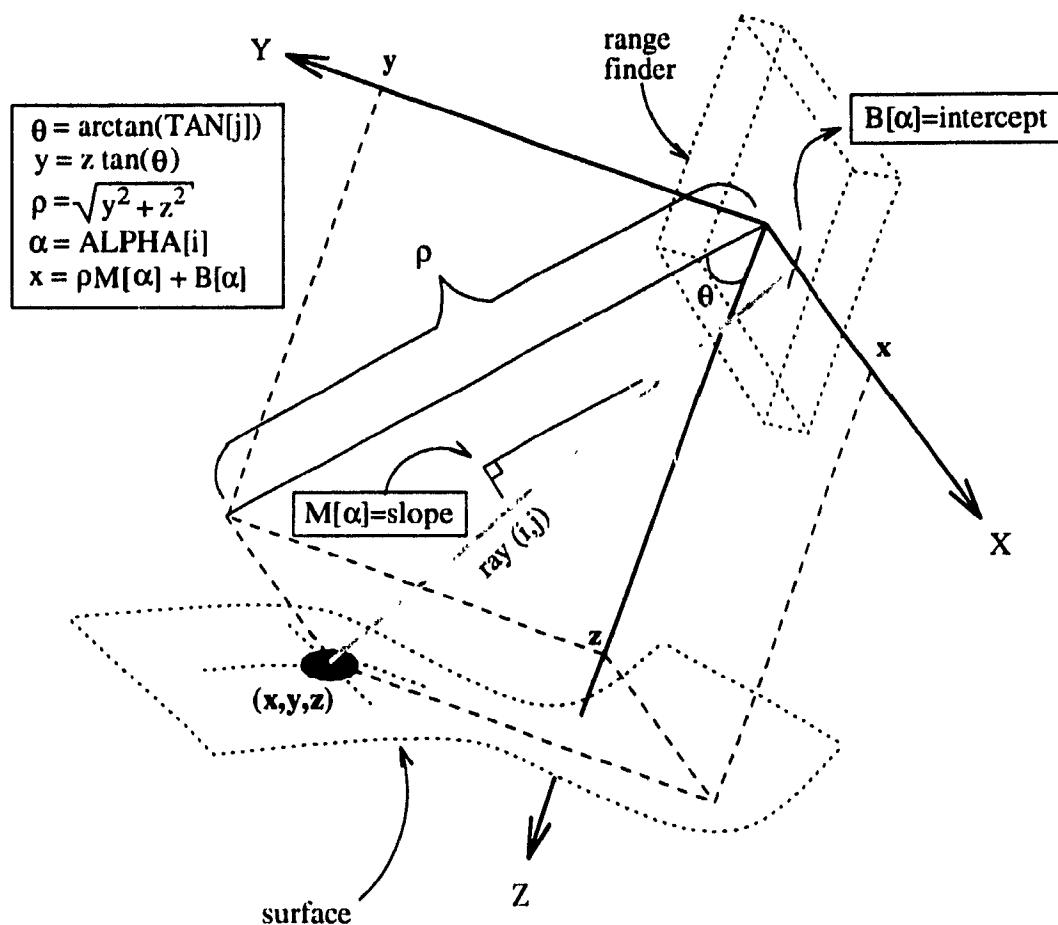


Figure 3.3: The geometric relationship between the 3D coordinates (x, y, z) and the different parameters of the rangefinder are shown in this figure. The coordinate reference frame is local to the rangefinder camera. The X-mirror is in its i^{th} discrete position and the Y-mirror in its j^{th} discrete position. The orientation of the laser ray is defined by the TAN, ALPHA, M and B calibration tables. The angle of the line with respect to the z-axis is determined from the TAN table. The value of ρ is the length of the projection of the ray onto the yz plane. The actual X-mirror position is determined by the ALPHA table. From this actual X-mirror position, the position and orientation of the ray away from the yz plane is determined by the M and B tables. These define the slope and intercept of the ray in the $x\rho$ plane, respectively

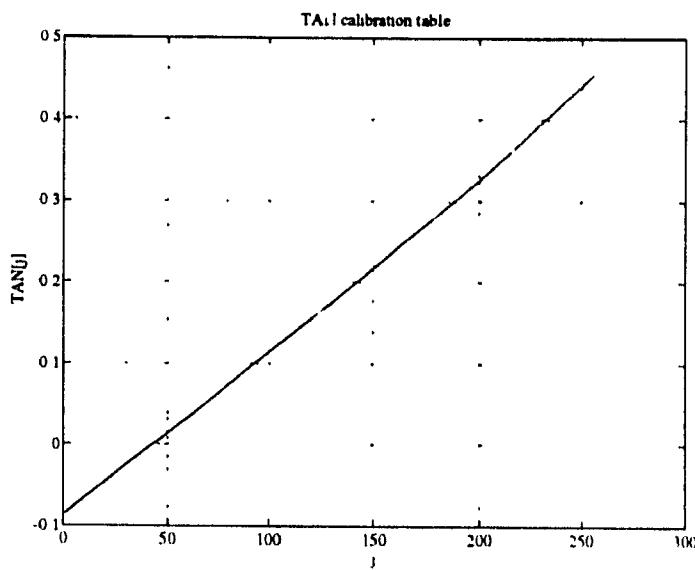


Figure 3.4: plot of the TAN calibration table

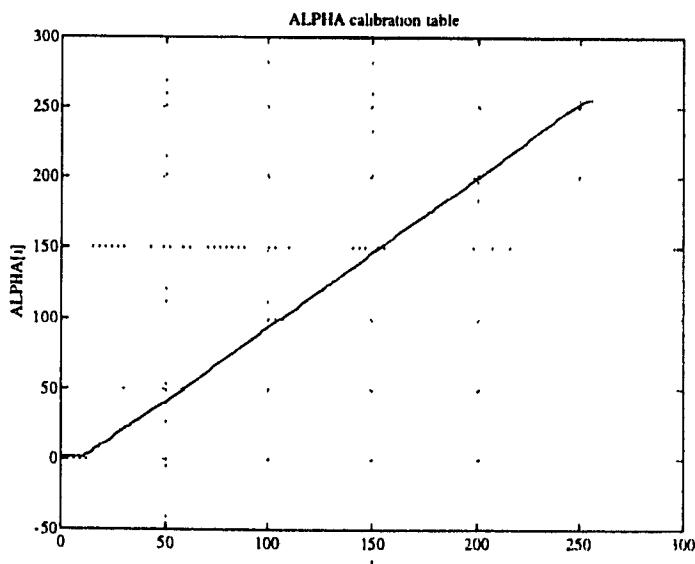


Figure 3.5: This graph shows how the actual position of the X-mirror differs from the position specified by the mirror index i . Sometimes the mirror does not change its position even though the index increases by 1 thus indicating that the mirror has moved to its next position. This results in the flat regions of the function. Sometimes the mirror skips an intermediate position even though the index only increases by 1, to indicate that the mirror should be at that intermediate position. This shows up as regions of the function with a slope greater than 1.

3.4.2 Reverse Calibration Process

By reversing the calibration process, we wish to compute the indices i and j from the coordinates (x, y, z) of a point in space. In other words, we want to find which ray would be the most likely to sample a point at an arbitrary position in space. For each ray (i, j) there is a corresponding sampled point in the (i, j) location of the 2D array of values forming a range image. Therefore, if one can associate a ray to any 3D point, we can establish a point to point correspondence between the given 3D point and the (i, j) point in the range image.

For each i, j combination, there is a specific ray with a given orientation in 3D space. One possible way of reversing the calibration process would be to compute the Euclidean distance of the specified point to every ray. The ray closest to the point is the desired one. Since a range image can have at most 256 by 256 sample points there are a total of $256 \times 256 = 65536$ rays to consider. Thus, it would be inefficient to compute the distance of the point to every ray to find the closest one. A better approach is to somehow reverse the process described in Equations (3.5) to (3.8) so that the indices i and j corresponding to the closest ray can be computed directly from the (x, y, z) coordinates.

a) Computing the Y-mirror index j

The inversion of Equation (3.5) allows us to determine the value of j from the values of y and z . Given the coordinate values y and z we want to find the index value j such that Equation (3.5) is satisfied. By dividing both sides of the equation by z and taking the inverse tangent we get

$$\arctan(TAN[j]) = \arctan\left(\frac{y}{z}\right) \quad (3.9)$$

The TAN table simply contains the tangent value for a series of 256 angles each corresponding to the position of the Y-mirror. By taking the inverse tangent value for each value in the table and plotting the resulting angle values with respect to the index values, a linear pattern emerges. Figure 3.6 shows a graph of the inverse tangent values of the table with respect to the value of the j index.

From the graph, we observe that the TAN calibration table models an underlying linear relationship. Namely, the inverse tangent of the values in the table varies linearly with the table index. This can be exploited to simply express the table as the tangent of a linear

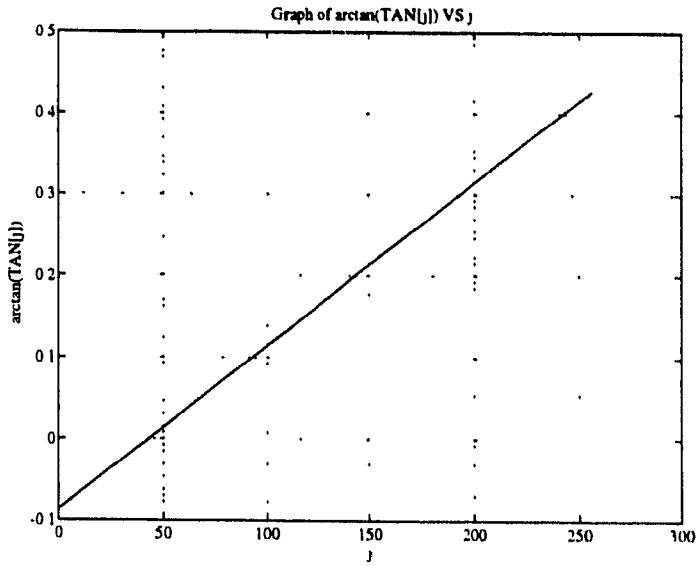


Figure 3.6: This graph shows the relationship between the inverse tangent of the values in the *TAN* calibration table with respect to the value of the *j* index. The graph clearly indicates a linear relationship between the two.

equation. After fitting a line to the data show in Figure 3.6, we can express the *TAN* table as

$$TAN[j] = \tan(\theta_0 + \theta_1 j) \quad (3.10)$$

where θ_0 and θ_1 are the parameters of the line that was fit to the inverse tangent values of the *TAN* table.

Inverting the table is now straightforward. From Equations (3.5) and (3.10) we derive

$$j = \frac{\arctan(\frac{y}{z}) - \theta_0}{\theta_1} \quad (3.11)$$

In order to guarantee that the result obtained for *j* is always an integer, Equation (3.11) is rounded to the closest integer value. We rewrite Equation (3.11) as follows

$$j = \left\lceil \frac{\arctan(\frac{y}{z}) - \theta_0}{\theta_1} + \frac{1}{2} \right\rceil \quad (3.12)$$

Equation (3.12) establishes a way to directly compute the *j* index from the (x, y, z) coordinates. The same procedure must now be accomplished for the *i* index.

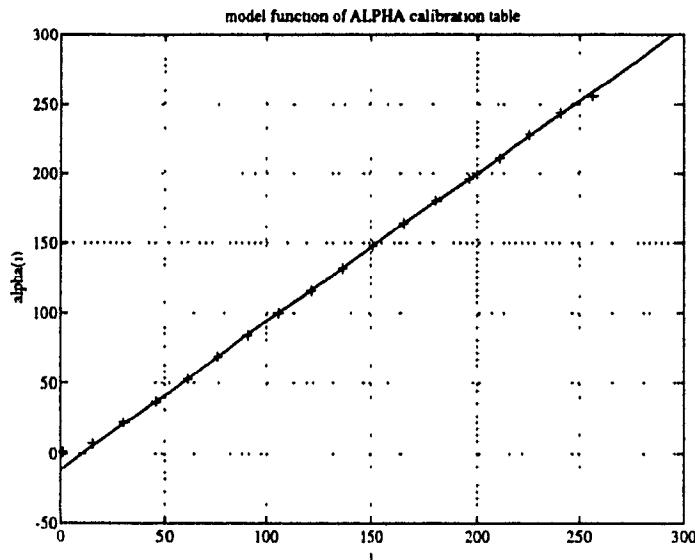


Figure 3.7: This graph shows the linear approximation of the ALPIIA calibration table. The equation of the line was found through a least-squares fit of the data contained in the table (see Figure 3.5). A few point of the table have been superimposed on the model (+ signs). The graph indicates a good fit of the model to the data of the ALPHA calibration table.

b) Computing the X-mirror index i

From Equation (3.7) and (3.8) we observe that in order to compute the index i , three calibration tables must be inverted. To invert Equation (3.7) the *ALPHA* calibration table must be inverted and to invert Equation (3.8), the *M* and *B* calibration tables must be inverted. In the following, the inversion process is explained for each of these two equations.

b.1) Inverting the *ALPHA* Calibration Table

The i index is used in conjunction with three calibration tables, the *ALPHA*, *M* and *B* tables. The relationship between *ALPHA*[i] and i can be approximated by a linear equation. A least-squares fit is performed on the points in the table as a function of the index. As can be seen from the graph of the *ALPHA* table, the first few and the last few points in the table deviate from the line. Therefore, to obtain a better approximation to the table, these points are considered as outliers and are not included in the least-squares fit. The result of the line fit is shown in Figure 3.7. Some points of the *ALPHA* table have been superimposed on the model in order to show how the linear model agrees with the actual data.

From Figure 3.7 we can see that the *ALPHA* table can be approximated by the linear equation:

$$\text{ALPHA}[i] = \alpha_0 + \alpha_1 i \quad (3.13)$$

Obtaining i from α is then achieved directly with the inverse line equation

$$i = \frac{\alpha - \alpha_0}{\alpha_1} \quad (3.14)$$

b.2) Inverting the *M* and *B* Calibration Tables

We can now consider the problem of inverting Equation (3.8). The goal is to determine the index value α of the line with slope $M[\alpha]$ and intercept $B[\alpha]$ for which the point (x, ρ) is closest in terms of Euclidean distance. The value of ρ is obtained directly from Equation (3.6).

We will present two methods for doing this. The first method approximates the *M* and *B* calibration tables by polynomial functions. The second generates an inverse look-up table. The advantages and disadvantages of each are briefly discussed.

b.2.1) Polynomial Approximations of *M* and *B*

Figure 3.8 shows graphs of the *M* and *B* calibration tables. By inspection we can determine that the tables behave similar to a third order polynomial function. A third order polynomial fit in the least-square sense can be performed. From Figure 3.8 we see that the fits agree well with the actual data in the tables.

The *M* and *B* calibration tables can therefore be approximated as follows:

$$M[\alpha] \simeq m_0 + m_1\alpha + m_2\alpha^2 + m_3\alpha^3 \quad (3.15)$$

$$B[\alpha] \simeq b_0 + b_1\alpha + b_2\alpha^2 + b_3\alpha^3 \quad (3.16)$$

Equations (3.15) and (3.16) can be substituted into Equation (3.8) which can then be rewritten as follows:

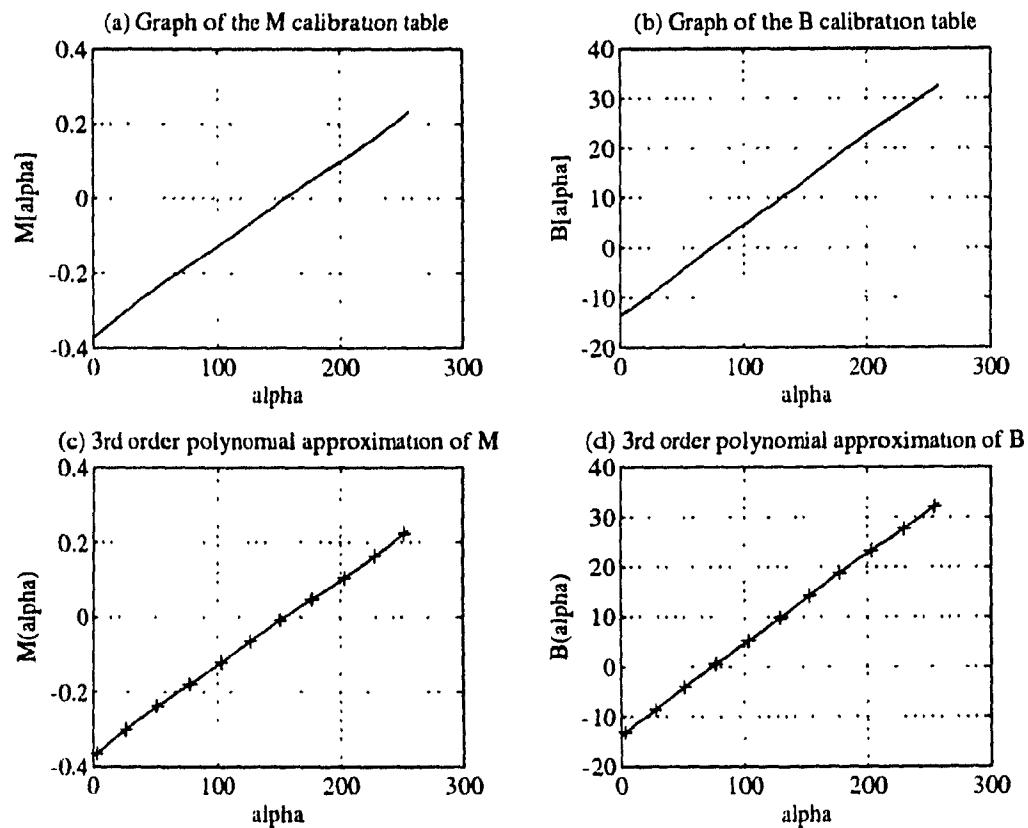


Figure 3.8: This figure shows a graph of the slope and intercept data contained in the calibration tables $M[\alpha]$ and $B[\alpha]$ as a function of the index α (graphs (a) and (b)). Graphs (c) and (d) are a plot of the third order polynomial function derived from the data. A few points of the original data are indicated with + symbols to demonstrate the closeness of the fit.

$$\begin{aligned} \alpha^3 + \left(\frac{\rho m_2 + b_2}{\rho m_3 + b_3} \right) \alpha^2 + \left(\frac{\rho m_1 + b_1}{\rho m_3 + b_3} \right) \alpha + \left(\frac{\rho m_0 + b_0 - x}{\rho m_3 + b_3} \right) &= 0 \\ \alpha^3 + a_2 \alpha^2 + a_1 \alpha + a_0 &= 0 \end{aligned} \quad (3.17)$$

This is a third order polynomial equation in α . This equation can be solved explicitly for α as a function of x and ρ . There are three roots, but over the range of possible values for x and ρ only one is real. We also note that Equation (3.17) is undefined for

$$\rho m_3 + b_3 = 0$$

or

$$\rho = \frac{-b_3}{m_3}$$

and for values close to this, we will run into numerical instability problems. However, this will not cause a problem if ρ is never equal to this value, which is the case for the rangefinder used in this research.

When solving Equation (3.17) for α , the three following solutions are obtained:

$$\alpha = \psi + \xi - \frac{a_2}{3} \quad (3.18)$$

$$\alpha = \left(-\frac{\psi}{2} - \frac{\xi}{2} - \frac{a_2}{3} \right) + \left(\frac{\sqrt{3}(\psi - \xi)}{2} \right) i \quad (3.19)$$

$$\alpha = \left(-\frac{\psi}{2} - \frac{\xi}{2} - \frac{a_2}{3} \right) - \left(\frac{\sqrt{3}(\psi - \xi)}{2} \right) i \quad (3.20)$$

where

$$\begin{aligned} \xi &= \sqrt[3]{\frac{a_2 a_1}{6} - \frac{a_0}{2} - \frac{a_2^3}{27}} + \frac{\gamma}{18} \\ \psi &= \sqrt[3]{\frac{a_2 a_1}{6} - \frac{a_0}{2} - \frac{a_2^3}{27}} - \frac{\gamma}{18} \\ \gamma &= \sqrt{4 a_1^3 - a_1^2 a_2^2 - 18 a_2 a_1 a_0 + 27 a_0^2 + 4 a_0 a_2^3} \sqrt{3} \\ i &= \sqrt{-1} \end{aligned}$$

Because we are only concerned with real physical solutions, Equation (3.18) is the only useful solution.

Since the inversion process must be performed repeatedly, evaluating Equation (3.18) to

compute α from x and ρ is not computationally efficient. Because of this, it is interesting to consider different ways of computing α . One way is to fit a linear equation to the M and B calibration tables instead of a 3rd order polynomial equation. Therefore, we could also approximate M and B as

$$M[\alpha] \simeq m_0 + m_1\alpha \quad (3.21)$$

$$B[\alpha] \simeq b_0 + b_1\alpha \quad (3.22)$$

Substituting Equations (3.21) and (3.22) into Equation (3.8) and solving for α we get

$$\alpha = \frac{\rho m_0 + b_0 - x}{\rho m_1 + b_1} \quad (3.23)$$

This is clearly computationally more efficient. However, some accuracy is lost.

b.2.2) Creating an Inverse Look-up Table

Another option is to perform the inversion of Equation (3.8) by using a look-up table. This option can lead to a fairly accurate inversion if the table is made large enough, and it is certainly efficient since only the index in the tables must be computed from x and ρ .

We create a two-dimensional look-up table *INVERSE_MB* that uses discretized values of x and ρ as its indices, and returns the appropriate α value for that particular (x, ρ) point. The size of the table and the discretization parameters are set on the basis of the rangefinder's limitations and of the desired accuracy of the inversion process. To discretize the x and ρ indices, a minimum and maximum value must be determined for each variable. The rangefinder used has a range of about 10 to 90 centimeters in depth. From this we can set

$$z_{min} = 100mm$$

$$z_{max} = 900mm$$

Using Equation (3.5) along with the minimum and maximum values of the *TAN* calibration table, we derive the minimum and maximum values for y . Note that if the minimum value of the *TAN* table is negative, the minimum value of y is obtained by multiplying the

maximum z value with that value:

$$y_{min} = \begin{cases} z_{max} \times TAN_{min} & \text{if } TAN_{min} < 0 \\ z_{min} \times TAN_{min} & \text{if } TAN_{min} \geq 0 \end{cases} \quad (3.24)$$

$$y_{max} = \begin{cases} z_{min} \times TAN_{max} & \text{if } TAN_{max} < 0 \\ z_{max} \times TAN_{max} & \text{if } TAN_{max} \geq 0 \end{cases} \quad (3.25)$$

Using Equation (3.6) and the min/max values of y and z as defined above, the min/max values of ρ are determined as follows:

$$\rho_{min} = \begin{cases} \sqrt{y_{max}^2 + z_{min}^2} & \text{if } y_{max} < 0 \\ \sqrt{0 + z_{min}^2} = z_{min} & \text{if } y_{min} < 0 \text{ and } y_{max} \geq 0 \\ \sqrt{y_{min}^2 + z_{min}^2} & \text{if } y_{min} \geq 0 \end{cases} \quad (3.26)$$

$$\rho_{max} = \begin{cases} \sqrt{y_{min}^2 + z_{max}^2} & \text{if } |y_{min}| > |y_{max}| \\ \sqrt{y_{max}^2 + z_{min}^2} & \text{if } |y_{min}| \leq |y_{max}| \end{cases} \quad (3.27)$$

Given ρ_{min} and ρ_{max} , we can then compute x_{min} and x_{max} using Equation (3.8) and the min/max values of the M and B calibration tables:

$$x_{min} = \begin{cases} \rho_{max} \times M_{min} + B_{min} & \text{if } M_{min} < 0 \\ \rho_{min} \times M_{min} + B_{min} & \text{if } M_{min} \geq 0 \end{cases} \quad (3.28)$$

$$x_{max} = \begin{cases} \rho_{min} \times M_{max} + B_{max} & \text{if } M_{max} < 0 \\ \rho_{max} \times M_{max} + B_{max} & \text{if } M_{max} \geq 0 \end{cases} \quad (3.29)$$

Knowing the minimum and maximum values, the range of possible values for x and ρ is well defined. With this information, the *INVERSE_MB* table can now be constructed. The size of the table determines its accuracy. The larger the table, the more finely discretized the range of possible values for x and ρ will be, and hence, the more precise the table will be. The table is accessed by two indices x_d and ρ_d which range from 0 to $x_{dim} - 1$ and $\rho_{dim} - 1$, where x_{dim} and ρ_{dim} are the size of the look-up table. The values of the indices x_d , ρ_d nearest to x and ρ are obtained directly in the following way:

$$x_d = \left\lfloor \frac{(x - x_{min})(x_{dim} - 1)}{x_{max} - x_{min}} + \frac{1}{2} \right\rfloor \quad (3.30)$$

$$\rho_d = \left\lfloor \frac{(\rho - \rho_{min})(\rho_{dim} - 1)}{\rho_{max} - \rho_{min}} + \frac{1}{2} \right\rfloor \quad (3.31)$$

The (x, ρ) point corresponding to a given *INVERSE_MB* table entry (x_d, ρ_d) can be simply determined from the inverse equations:

$$x = \frac{x_d}{(x_{max} - x_{min})}(x_{dim} - 1) + x_{min} \quad (3.32)$$

$$\rho = \frac{\rho_d}{(\rho_{max} - \rho_{min})}(\rho_{dim} - 1) + \rho_{min} \quad (3.33)$$

To construct the table, the closest ray, one out of 256 possible ones defined by the *M* and *B* calibration tables, to each discrete point (x, ρ) in the table is found. The index α of that ray is stored in the *INVERSE_MB* table at that given position. Figure 3.9 illustrates the relationship between the values in the *INVERSE_MB* table and the 256 rays defined by the slopes and intercept in the *M* and *B* calibration tables. The lines all lie in the x - ρ plane. The rectangular region defined by x_{min} , x_{max} , ρ_{min} and ρ_{max} represents the extent of the *INVERSE_MB* look-up table. An entry in the table is set to the index of the ray closest in Euclidean distance to the discrete (x_d, ρ_d) point corresponding to that entry.

There are entries in the table for which the (x_d, ρ_d) values correspond to points in space which could not possibly have been scanned by any ray. These are the points which fall outside the viewing range of the rangefinder, but are nevertheless within the range of the table. In Figure 3.9 these table entries correspond to the shaded areas inside the table region where no rays are present. In the *INVERSE_MB* table, an entry with a value of 0 indicates that no ray is close enough to account for it. The other entries are set to an integer value between 1 and 256 representing the value of the α index of the closest ray.

In order to delimit the shaded areas properly, we define two new rays with slopes and intercepts extrapolated from the extremities of the *M* and *B* tables. Since the α index for these tables varies from 1 to 256, we call the two extrapolated rays, 0 and 257. The slope and intercept values for these lines are determined directly by evaluating the third order polynomials derived for *M* and *B* at 0 and 257. Figure 3.8 shows plots of the third order polynomial fits to *M* and *B*:

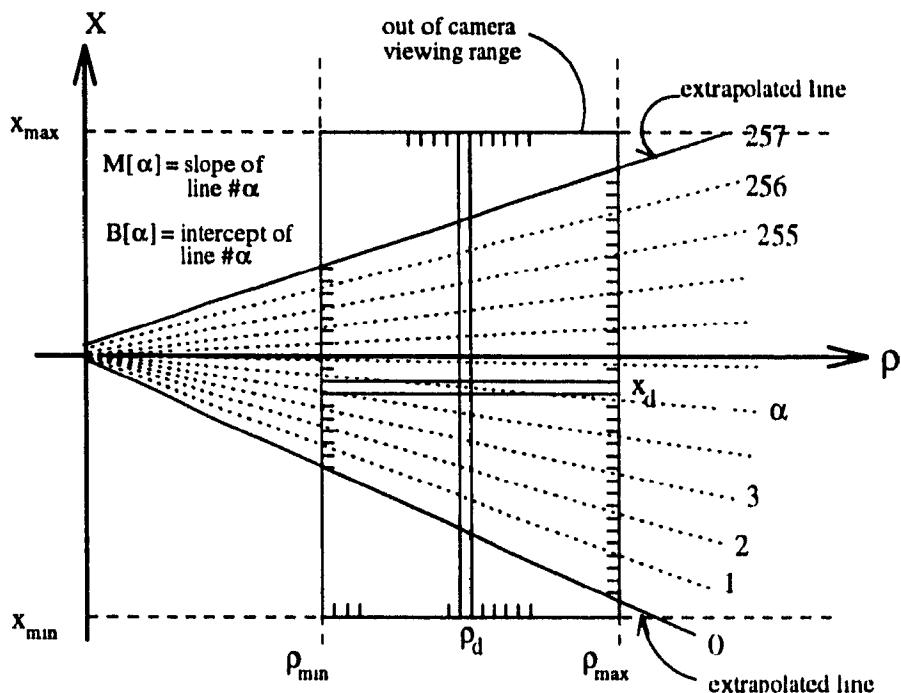


Figure 3.9: This figure shows the 256 rays defined by the slopes and intercept values contained in the M and B calibration tables. The region delimited by x_{min} , x_{max} , ρ_{min} and ρ_{max} defines the extent of the *INVERSE_MB* table. Each entry in the table corresponds to the α value of the line closest to the (x_d, ρ_d) point matching that entry. The areas inside the rectangular region where no rays are present, represent points that are outside the viewing range of the rangefinder camera. The table entries in these areas are set to 0 to reflect this fact.

$$\begin{aligned}
 M_0 &= m_0 \\
 M_{257} &= m_0 + m_1(257) + m_2(257)^2 + m_3(257)^3 \\
 B_0 &= b_0 \\
 B_{257} &= b_0 + b_1(257) + b_2(257)^2 + b_3(257)^3
 \end{aligned}$$

Now, all the elements required to build the table are available. The table is constructed as follows:

1. For each entry position in the table, compute the (x, ρ) coordinate corresponding to the indices x_d and ρ_d of that table entry using Equations (3.32) and (3.33).
2. Compute the Euclidean distance from point (x, ρ) to all 256 scan lines and to the two extrapolated rays.
3. If the closest line is either one of the 2 extrapolated lines (line 0 or 257) then write a 0 in the table entry at position x_d and ρ_d .
4. If the line closest to the point (x, ρ) is the line with slope $M[\alpha]$ and intercept $B[\alpha]$ then write α in the table entry at position x_d and ρ_d .

b.2.3) Polynomial Approximation vs. Look-up Table

The choice of how to compute α is based on a few factors. One aspect is the time required to generate the polynomial fits to the calibration tables as opposed to the time required to create the inverse look-up table. The polynomial fits can be accomplished very simply and efficiently; using any statistical analysis package one can automate this process very easily. Creating the look-up table on the other hand necessitates a rather long computing time, orders of magnitude greater than the polynomial fits. This can be of concern if the calibration of the rangefinder is performed often. However, because both the polynomial fit and the look-up table creation are performed off line, the time required for this operation does not actually affect the performance of the registration procedure in any way.

What does affect the registration procedure is the time required to evaluate the inverse calibration equations. If computing time is not an issue in the task being performed, and high accuracy is required, then Equation (3.18) is the best way of computing α . If, however,

speed of execution is important, then computing α from Equation (3.23) or obtaining it from the *INVERSE_MB* look-up table are better choices. The difference between the two is that the look-up table can be made as accurate as desired by increasing its size. This however necessitates a system with a large memory capacity. The accuracy of Equation (3.23) is only limited by how closely it models the rangefinder, but does not require large amounts of memory space. Hence, the choice between the two is a compromise between accuracy and memory space.

3.4.3 Summary of Calibration Process Inversion

The reason for inverting the calibration process is to have a means of establishing a correspondence between views. The forward process of the calibration consists of computing the (x, y, z) coordinates of a point from the depth value z of that point along with the indices i and j of its location in the array of points sampled with the laser rangefinder. Equations (3.5) to (3.8) describe how this process is accomplished. The reverse process consists in computing the indices i and j from the (x, y, z) coordinates of a point. This is equivalent to finding the closest ray to the 3D point defined by the coordinates. Thus, a 3D point whose closest ray is the (i, j) ray is put in correspondence with the point in a range image array at location (i, j) .

The reverse calibration process is done as follows:

1. Compute j from y and z using Equation (3.12).
2. Compute ρ using Equation (3.6).
3. Compute α from ρ and x using Equation (3.18) or (3.23) or by using the *INVERSE_MB* look-up table.
4. Compute i with Equation (3.7).

3.5 Formulating the Registration Task as an Optimization Problem

The object of the registration task is to find a transformation T' which best represents the relative displacement and orientation between two range views. An optimization search algorithm is used to search for the best transformation. Therefore, we must define some kind

3. Registration via Inverse Camera Calibration

of measure by which transformations can be ranked based on the quality of the registrations they produce.

It is assumed that there is a region of overlap between the two range views for which the correspondence transformation must be determined. What this means is that parts of the scene viewed in one view are also seen in the other view. If the exact correspondence between the two views were known, the 3D data contained in them could be expressed in a single coordinate frame of reference. This would result in 3D surfaces in the scene which are seen from both views to exactly coincide when the views are expressed in the same reference frame. In terms of range images, this means that the 3D data points in each of the views which were obtained by sampling the same 3D surface point in the scene, should have the same coordinates values when expressed in the same reference frame. Since it is very unlikely that a location in a scene is sampled at the exact same physical position from two different view points, the points would not coincide in practice. However, the 3D points would be very close to one another since they are samples of the same region of a surface in the scene. Therefore, the distance between sampled points, once expressed in the same coordinate frame, is an indication of the quality of the transformation between the views. A good transformation would make all points of one view which are part of the overlap region in the scene, very close to a corresponding point in the other view. A poor transformation would result in a large distance between points of one view to points of the other view. This idea is used for ranking the transformations. A good transformation yields a small distance between corresponding points in the two views and a poor transformation yields large distances between corresponding points.

The search algorithm tries to find the transformation that minimizes the distances between points of one image after transformation to corresponding points of the other image. As we have seen in the previous section, with each sensor point sampled on the surface of a scene, there is an associated ray (with index i and j). A point in one image after transformation is put into correspondence with the point in the other image whose associated ray is closest. The Euclidean distance between the transformed point and the rays is used to define the closest ray.

The goal of the search is therefore to find the transformation which minimizes the overall distances between the views in the direction of the rays of one of the views. Cheng and Medioni [4] have used a similar idea. In their work, they minimized the distance between

the surfaces of each view in the direction of the normal to the surface. This distance, the perpendicular from one surface to another, seems intuitively better. However, as was mentionned in the introduction, it has the disadvantage that finding the corresponding point in the other view, requires a search algorithm in itself. By minimizing the distance between views along the ray, the corresponding point in the other view is found directly from the inverse camera equations defined in the previous section. This eliminates the search required by Cheng and Medioni. Hence, this correspondence method can be used to define a cost function which can be evaluated very fast.

3.5.1 Cost Function Definition

We will now define the cost function needed for the optimization task more formally. Let S_c be a set of control points taken from the total set of points on the first view. S_c is a subset of all sampled points in that view. How these points are chosen will be discussed in Section 3.5.2. Let T be the transformation which takes a point in the first view and expresses it in the coordinate reference frame of the second view. If \vec{p} is a point in the first view, then $T(\vec{p})$ is the same point expressed in the second view's coordinate frame. With respect to the definition of T given in Section 3.2, this is simply

$$T(\vec{p}) = \mathbf{R}\vec{p} + \vec{d}$$

where \vec{d} is derived directly from the translation parameters t_x , t_y and t_z (Equation (3.2)); and \mathbf{R} is derived directly from the rotation parameters r_x , r_y and r_z (Equation (3.3)).

Let $C()$ be the correspondence function defined by the camera's inverse calibration equations defined in Section 3.4. If \vec{q} is a point in the second view's coordinate frame, then $C(\vec{q})$ is the point in the second view whose associated ray is closest to point \vec{q} . The input to the function are the coordinates (x, y, z) of a 3D point. From these coordinates, the indices i and j of the closest ray are found using the inverse calibration equations. The result of the function is the 3D point of the range image at location (i, j) . It is possible that the indices i and j found by the inverse calibration equations do not represent any valid point in the range image. In such cases, $C()$ would return *undefined* as a result to indicate that no correspondence has been found. Given a transformation T , we define a cost function for T as follows:

$$\text{cost}(T) = \sum_{\vec{p} \in S_c} d(T(\vec{p}), C(T(\vec{p}))) \quad (3.34)$$

where $d()$ is the 3D Euclidean distance (L_2 norm) between two points. The cost function is an indication of the registration quality of the transformation T . The greater the accumulated distance between points in the views due to a transformation, the higher the cost value of this transformation will be. Therefore, the transformation yielding the best registration of the range images will be the one with the lowest cost. An optimization search can then be applied to find the transformation T which minimizes this cost function.

Because range images are obtained from different viewing positions, there are regions of the images which will be unique to each image and other regions which will be common to them. As a consequence, when registering two range images, some control points in one view will not have a correspondence in the other view because they are unique to one view. As a consequence, the inverse calibration function $C()$ is not guaranteed to return a correspondence to all control points in S_c , and in such a case the L_2 norm is undefined. Therefore, the cost function defined by Equation (3.34) is inadequate. We must define the distance function $d()$ to deal with the case where $C(T(p))$ is undefined. One solution is to set the distance value to 0 when no correspondence is found.

Another aspect which must be taken into consideration is the inherent noise present in range data. For example, boundaries of object introduce large steps in the range surface and shadows can affect the laser sampling process and introduce spurious data points. As shown in Figure 3.10, improper point correspondences between two range views can occur at the edges of the object. If the distance between the points in such a correspondence is large, it will affect the value of the cost function.

Therefore, in order to limit the effect of the Euclidean distance on the cost function, a threshold τ is introduced. The value of the distance threshold limits the negative effects on the cost value of a good transformation that improper correspondences would have. Figure 3.10 shows a typical case where a good transformation could be negatively affected by large distances between control points and corresponding points.

If a distance is greater than the threshold, the distance is set to the threshold value. This is done so that a single point error which is large does not mask a good transformation by increasing its cost arbitrarily. Thus the distance measure is defined as follows:

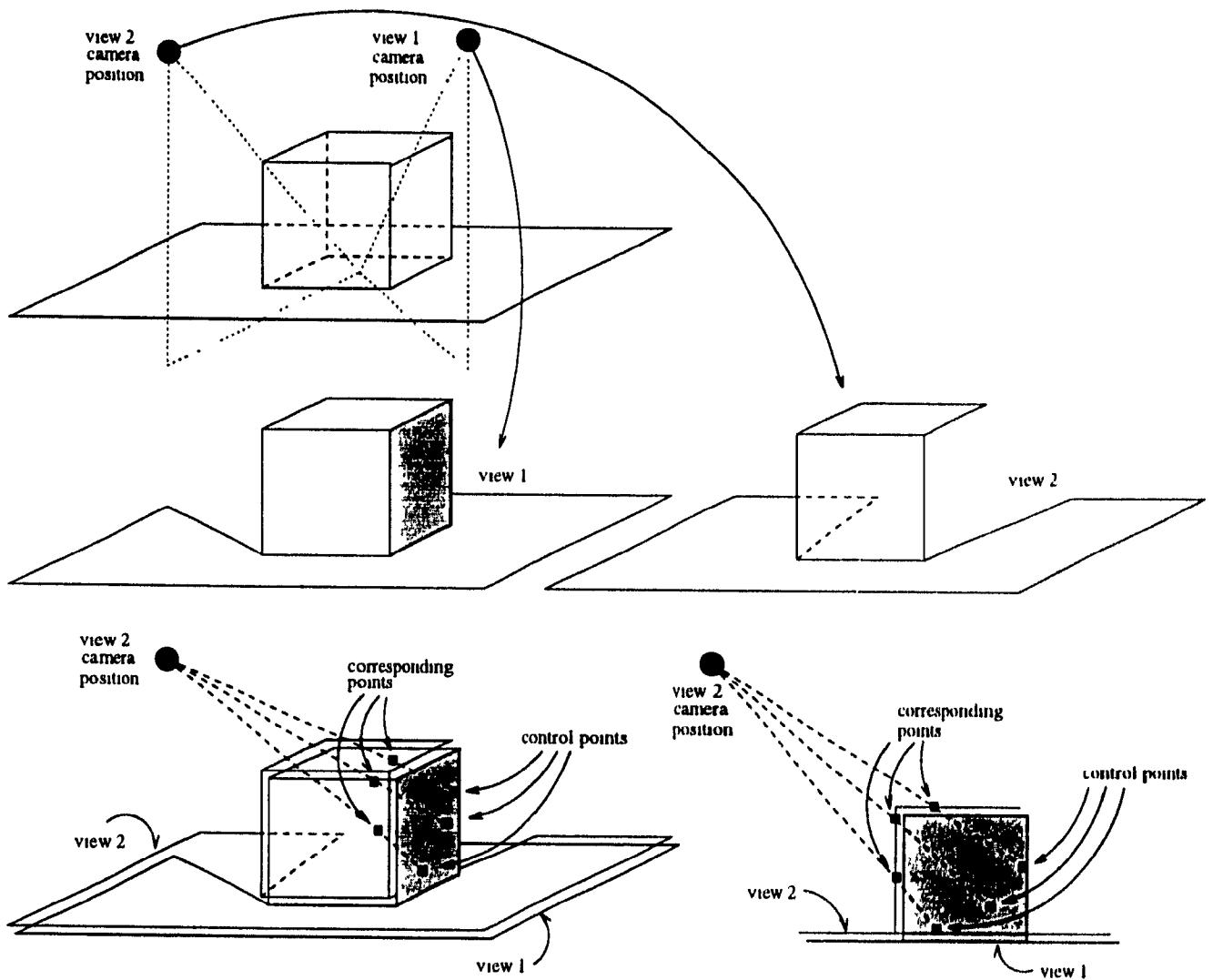


Figure 3.10: This figure illustrates the use for the distance threshold. Two range images are acquired from the same scene (a cube on a plane). The points sampled from the side of the cube (shown by a shaded region) in the first view are unique to view 1. View 2 is a straight on view of the cube and thus does not contain any points sampled on the side of the cube. The fitness of a transformation is derived from the sum of distance between all control points in view 1 and corresponding points in view 2. The control points from the side of the cube should not have any correspondence in view 2 since they are unique to view 1. However, because the points fall within the viewing range of view 1, each control point has a closest scan line and by definition a corresponding point in view 1. The distance between these false corresponding points is large and thus could have a negative effect on the cost function value of a good transformation. The distance threshold limits the effect of such false correspondence by limiting the distance values added to the cost function.

$$d(\vec{p}_1, \vec{p}_2) = \begin{cases} \|\vec{p}_1 - \vec{p}_2\| & ; \text{ if } \vec{p}_2 \text{ is defined and } \|\vec{p}_1 - \vec{p}_2\| \leq \tau \\ \tau & ; \text{ if } \vec{p}_2 \text{ is defined and } \|\vec{p}_1 - \vec{p}_2\| > \tau \\ 0 & ; \text{ if } \vec{p}_2 \text{ is undefined} \end{cases} \quad (3.35)$$

Because the distance measure returns a 0 value when no correspondence point is found, a transformation minimizing the number of correspondences would yield a minimum cost value. This is not desirable since very poor transformations will likely yield very few correspondences by definition. To alleviate this problem, the sum of the distances can be normalized by the number of correspondences. Let $S_c(T)$ be the set of all control points for which a correspondence exists under the transformation T . We redefine the cost function as follows:

$$\text{cost}(T) = \frac{\sum_{\vec{p} \in S_c} d(T(\vec{p}), C(T(\vec{p})))}{\|S_c(T)\|} \quad (3.36)$$

There is still a problem with the cost function defined in this way. Because no penalty is assigned to transformations yielding few correspondences, the cost function will still be a minimum when no correspondences are established between views. This problem can be handled by enforcing a certain overlap between the views. Any transformation yielding less than the specified overlap is then discarded by giving it the highest cost value. All transformations making the views overlap by the specified factor or more are then based on the normal cost function defined by equation (3.36). Specifying a minimum required overlap also has the advantage that one can use prior knowledge about the overlap between the views to guide the optimization search by imposing a constraint on the transformation search space. For example, if we know from the data acquisition stage that the range images overlap by at least 40%, then this information can be used to discard all transformations yielding less than 40% overlap.

Let Ω be the overlap factor. For example $\Omega = 0.3$ means that at least 30% of overlap is required between the range views. The overlap generated by a transformation T is simply the total number of correspondences achieved $\|S_c(T)\|$ divided by the total number of control points $\|S_c\|$. Because of the threshold τ and because control points without correspondence result in a distance value for $d()$ of 0, it is clear that no transformation can yield a cost value greater than the cardinality of S_c times the threshold value τ . This idea is used to

set the maximum value of the cost function. With this, we redefine the cost function as follows:

$$cost(T) = \begin{cases} \frac{\sum_{\vec{p} \in S_c} d(T(\vec{p}), C(T(\vec{p})))}{\|S_c(T)\|} & ; \text{ if } \frac{\|S_c(T)\|}{\|S_c\|} \geq \Omega \\ \tau \|S_c\| & ; \text{ if } \frac{\|S_c(T)\|}{\|S_c\|} < \Omega \end{cases} \quad (3.37)$$

The following algorithm illustrates how the cost function is computed for a given transformation.

```

Cost = 0
NumCorrespondences = 0
For all control points  $\vec{p}$  in first range image [ $\forall \vec{p} \in S_c$ ]
    Transform  $\vec{p}$  using transformation to get  $\vec{p}'$  [ $\vec{p}' = T(\vec{p})$ ]
    Find point  $\vec{q}$  in second range image corresponding to  $\vec{p}'$  using
    inverse camera equations [ $\vec{q} = C(\vec{p}')$ ]
    If  $\vec{p}'$  has no correspondence  $\vec{q}$  [if  $\vec{q} = \text{undefined}$ ]
        distance = 0 [ $d = 0$ ]
    Else
        NumCorrespondences = NumCorrespondence + 1
        distance = Euclidean distance between  $\vec{p}'$  and  $\vec{q}$  [ $d = \|\vec{p}' - \vec{q}\|$ ]
        If distance > Distance Threshold [if  $d > r$ ]
            distance = Distance Threshold [ $d = r$ ]
        EndIf
    EndIf
    Cost = Cost + distance [ $Cost = Cost + d$ ]
EndFor
If NumCorrespondences/NumControlPoints < Overlap Factor [if  $\|S_c(T)\|/\|S_c\| < \Omega$ ]
    Cost = NumControlPoints × Threshold [ $Cost = \tau \|S_c\|$ ]
EndIf

```

Note that by establishing the point to point correspondence across range images by using the inverse camera calibration equations, we in fact compute the sum of the distances between the two range views in the direction of the rays. This is different from the intuitive way of evaluating distance, where the distance between a point on a surface to the other surface

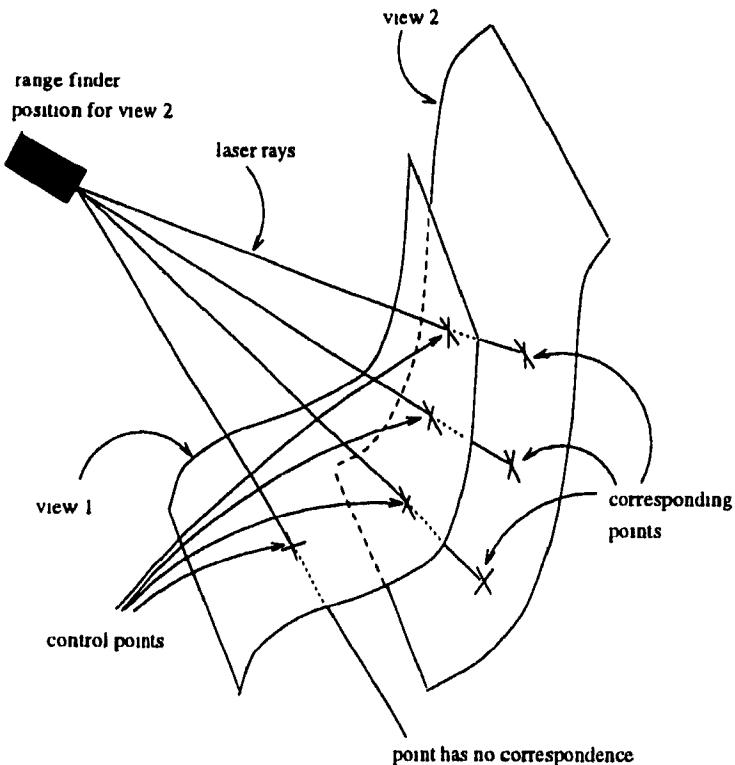


Figure 3.11: This figure shows how the point to point correspondence is established across range images. A control point in view 1 after transformation is put in correspondence with the point in view 2 having the closest ray.

is either the perpendicular distance or the distance to the closest point. One may argue that such distance measurements give a better indication of the distance between the two views than a distance in the direction of the rays. However, as the optimization progresses and the registration between the two views improves, the distance along the scan lines will approach the perpendicular distance measure. Figure 3.11 illustrates how the point to point correspondence is established between two range images. By using the inverse camera calibration equations, we are able to determine the closest ray to any control point in space. The sample point in the second view belonging to this ray is the desired corresponding point. We can clearly see that computing the distance between corresponding points is equivalent to computing the distance between the two surfaces in the direction of the scan lines.

Figure 3.12 shows the process involved for the fitness evaluation of a transformation. In the first figure, we see a typical range data acquisition. Two range images of a scene are obtained from two different viewing positions. The second figure shows how the fitness of transformation is evaluated in the optimization search. The first step consists in the selec-

tion of control points in the first range image. These control points are then expressed in the coordinate frame of the second range image. This is done by applying the motion transformation we wish to evaluate to all control points. Once the control points are mapped in view 2's reference frame, the corresponding points in the second range image are determined (through the inverse camera calibration equations). The Euclidean distances between all control points and their corresponding points are summed. This sum represents the cost of the transformation. Two transformation examples are shown. In the first example, the transformation provides a good registration of the range images. In the second example, the transformation provides a poor registration. A good registration yields small distances between control points and their correspondences and hence results in a low cost value. A poor registration yields larger distances between the two views and therefore results in a large cost value.

3.5.2 Control Point Selection

The reason for selecting only a subset of the total number of points contained in one view is the increase in speed of execution. If all points were used, it would mean that in order to evaluate the cost function for a given transformation, all the points in one view would have to be transformed, the point in the other view corresponding to the transformed point found and the Euclidean distance between the two points computed. Since a typical range image consists of a two-dimensional array of 256 by 256 sample points, a cost function evaluation would require 65536 point computations. Because the cost function can potentially be evaluated a large number of times during the optimization stage, it is desirable to keep the time required to perform an evaluation as small as possible. The selection of the control points on one of the views is done in a very simple way. The range images acquired are simply two-dimensional arrays of sampled points. We choose an interval by which points are selected. This results in a regular grid of points covering the total extent of the surface but including only a subset of the sample points defining the surface. Figure 3.13 illustrates this idea.

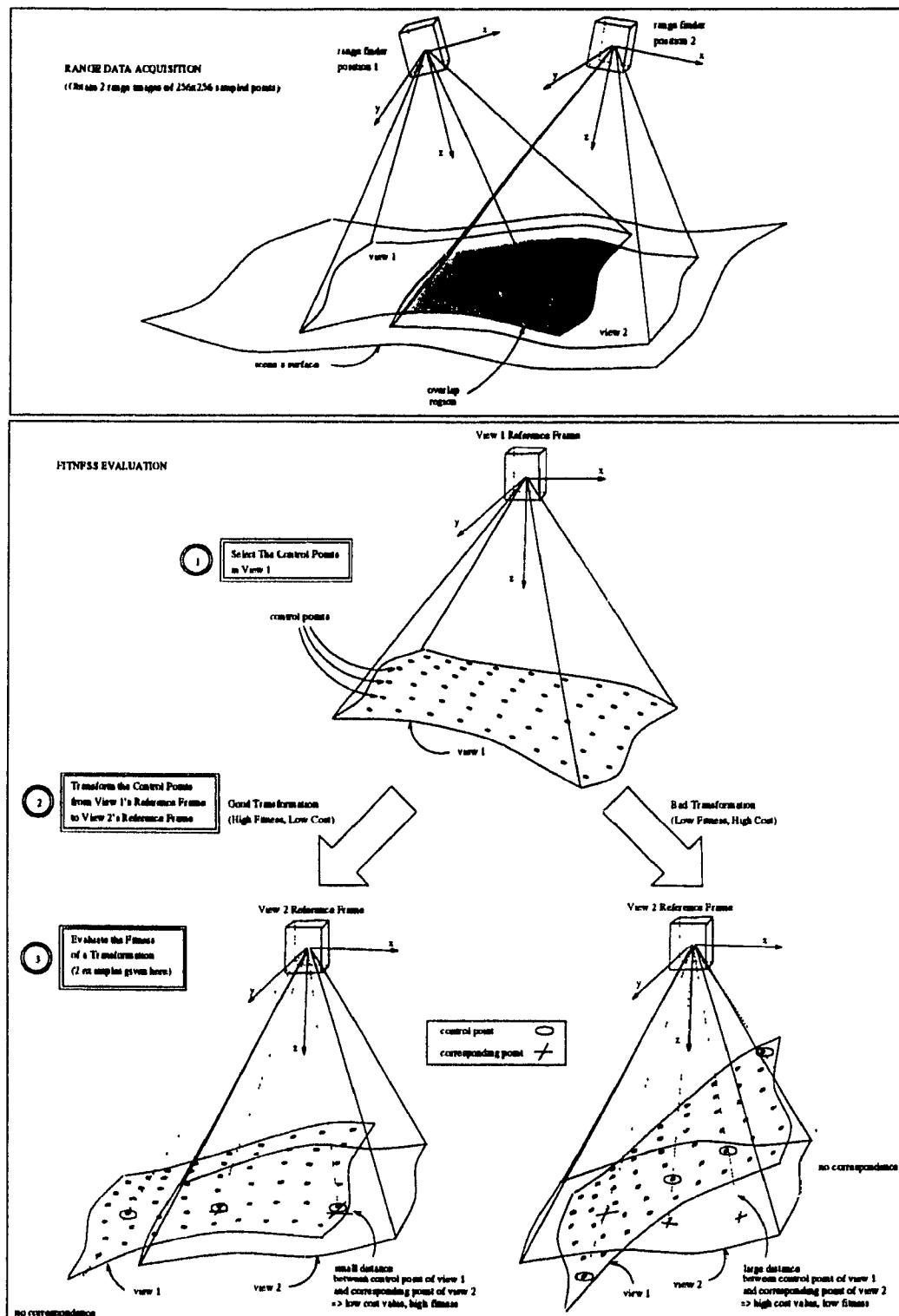


Figure 3.12: This figure illustrates the process of fitness evaluation. The first figure illustrates a typical range data acquisition. The second figure shows how the fitness of a transformation is evaluated. Two cases are shown: A good (left) and a bad transformation (right).

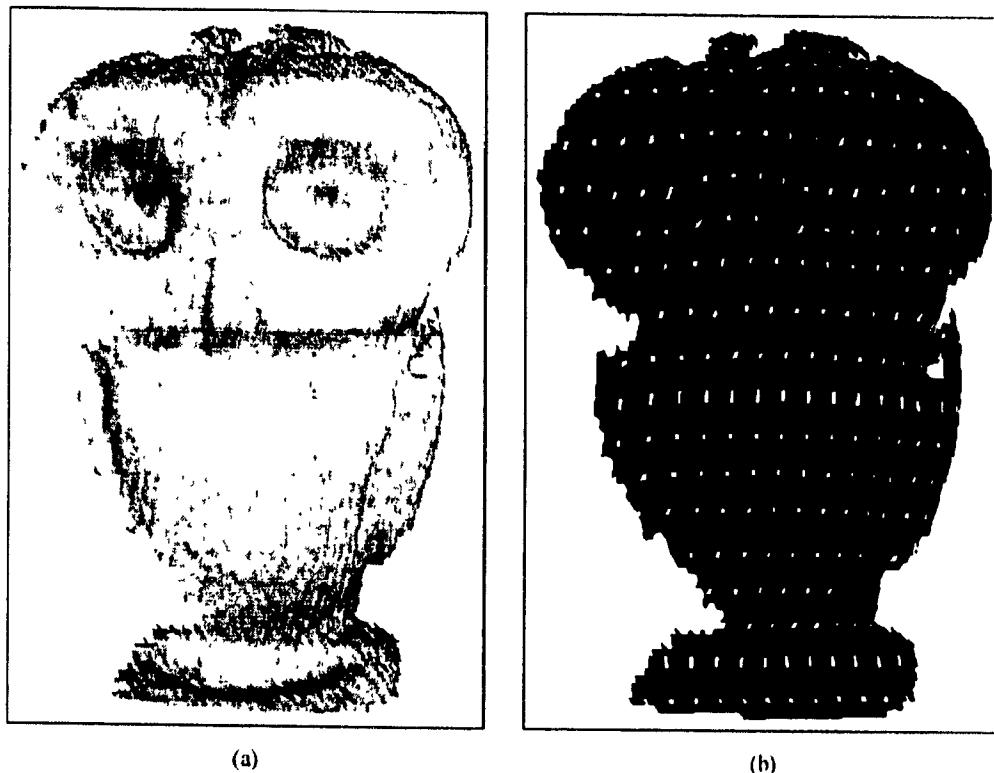


Figure 3.13: This figure shows how the control points are selected in a range image. Figure (a) shows the original range view from which the control points are taken. The scanned object is a small owl figurine and is shown as a shaded surface. The range image contains 256x256 sampled points, but has been segmented to keep only the data belonging to the figurine. Figure (b) shows the control points superimposed on the black colored surface. One can see the grid formed by the control points on the surface. The control points were taken at an interval of 5 sample points in the two-dimensional grid of sampled points, resulting in 252 points.

4.1 Introduction

This chapter presents two optimization search methods that have been used for the range image registration task. The first optimization method to be examined is based on genetic algorithms (GA). The second method is Very Fast Simulated Reannealing (VFSR). These search methods were chosen for their robustness in finding a global minimum in the presence of multiple local minima.

In Section 3.5.1, a cost function was defined. The variable parameters of this cost function characterizes the 3D rigid transformation T . This transformation is composed of six motion parameters: three translation parameters t_x , t_y and t_z and three rotation parameters r_x , r_y and r_z . The minimization is therefore conducted in a 6D search space and the cost function can be viewed as a 7D surface whose domain is the motion parameter search space. The goal of the search is to find the values of the transformation parameters necessary for the cost function to be at its global minimum value. In Section 4.2, the general shape of the cost function is examined. Based on this, the choice of search method is briefly discussed.

Section 4.3 presents an overview of the GA search method and Section 4.4 presents the VFSR method.

4.2 Shape of the cost function

Before deciding on an approach to performing the minimization of the cost function defined in Chapter 3, it is necessary to examine how the cost function behaves. This behavior can indicate an appropriate optimization strategy. If the cost function varies monotonically as the motion parameters are varied and a single minimum exists, then a simple gradient descent procedure could be used to find the optimum transformation. On the other hand, if many local minima are present and the cost function varies very chaotically as the parameters are varied, then a more robust minimization method must be considered.

Because the cost function is a measure of distance between two range views, its shape will vary for different pairs of range views being registered. If the range data to be registered is

very smooth and uniform (range views of a smooth vase for example), then the cost function will vary smoothly as the transformation parameters vary. If the range data is not smooth (range views of a pile of rocks for example), then the cost function will have many sharp transitions and many local minima. As a consequence, to ensure that the minimization of the cost function is successful for many different types of object surfaces, we must rely on a robust search method.

Plotting the cost function helps us understand its shape. The cost function to be minimized is basically a surface in seven dimensions. In order to visualize it, 3D slices are plotted. The domain of the function is the real 6D space formed by the motion parameters t_x , t_y , t_z , r_x , r_y and r_z . To plot a 3D slice, two parameters are varied while the others are kept constant. With six motion parameters, we can thus generate fifteen different slices of the cost function.

To generate plots of the cost function, a typical set of range image views are used (see Figure 4.1). The motion parameters of the transformation yielding the minimum value of the cost function are known exactly. The motion parameters are taken two at a time and varied around this optimum setting while the others are kept constant. The optimum transformation values for the images of Figure 4.1 are:

$$\begin{aligned} t_x &= -46.547913 \\ t_y &= -149.324738 \\ t_z &= 82.531540 \\ r_x &= -33.647762 \\ r_y &= 5.385911 \\ r_z &= -19.028326 \end{aligned}$$

The translation parameters t_x , t_y and t_z were varied by ± 40 millimeters around their optimum value and the rotation parameters, r_x , r_y and r_z were varied by ± 5 degrees. This defines the range over which the cost function is plotted. The cost function is evaluated at regular intervals within this range and a plot of the cost value as a function of the two chosen parameters is created. Figures 4.2, 4.3, 4.4 and 4.5 show the slices of the cost function surface obtained.

In Figure 4.2, 4.3, 4.4 and 4.5 we observe that in general the shape of the cost function as the motion parameters are varied is more or less convex. There are usually many small local

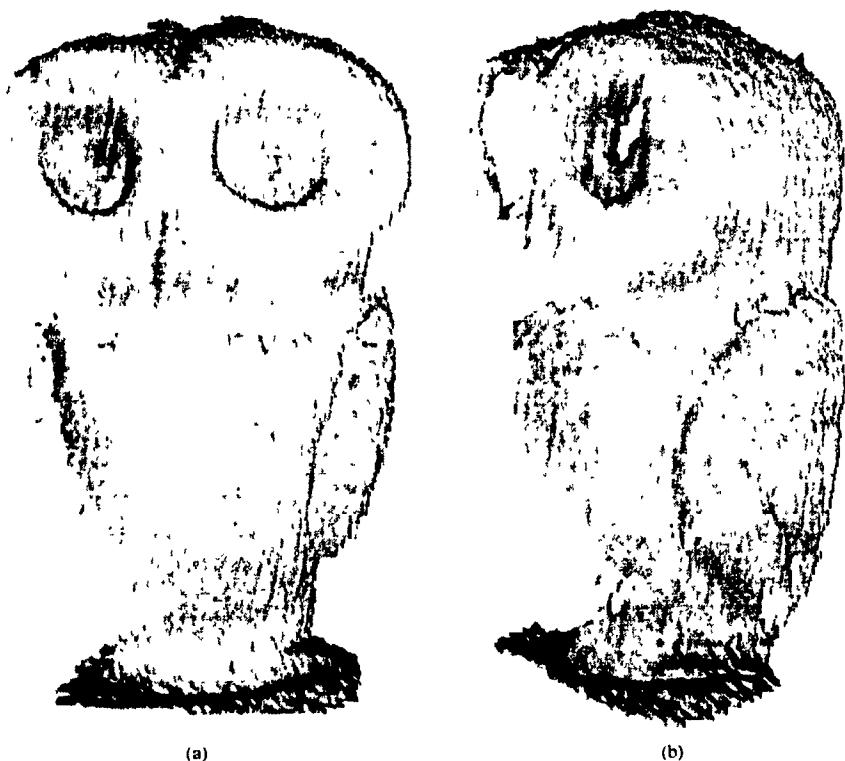


Figure 4.1: This figure shows an example of two typical range views that can be registered by the registration algorithm. The object sampled by the rangefinder camera is a small owl figurine. The data acquisition was performed with the use of a precision turntable and the views were obtained at a 40 degree rotation interval (view (a) at 0 degrees, (b) at 40 degrees). The background surfaces as well as spurious data points were removed to retain the owl's surface only. Both views are shown as a shaded surface.

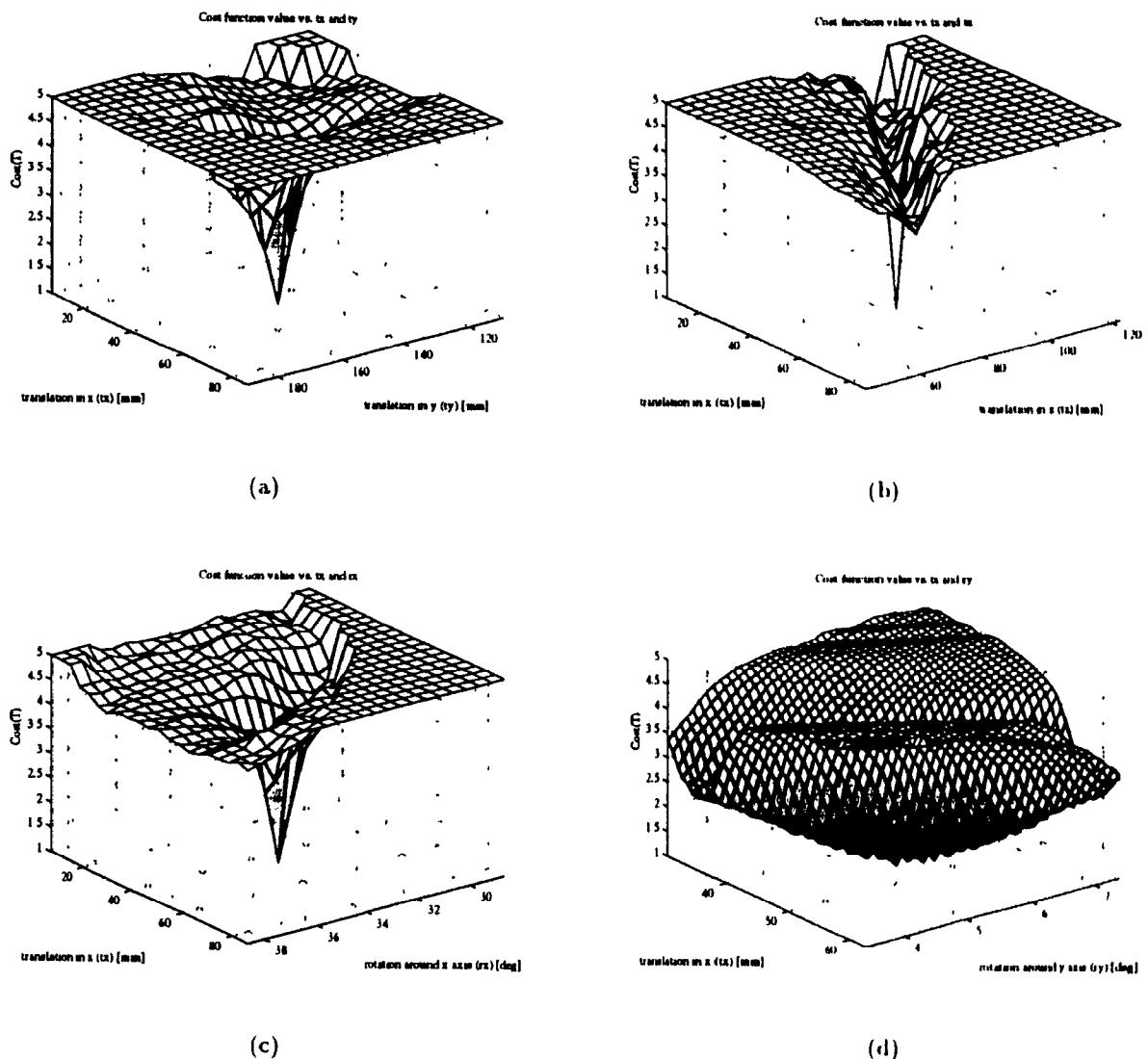


Figure 4.2: This figure shows various 3D slices of the cost function obtained for the two range views of Figure 4.1. Figure (a), (b) and (c) all exhibit a very steep global minimum in the center with multiple local minima present in a neighborhood around it. The flat portions of these slices correspond to the maximum cost value obtained when the two views are so badly registered that the distance between all control points and their respective correspondence exceeds the distance threshold, or that the overlap between the views is less than the specified overlap factor. Figure (d) exhibits a smooth transition to the global minimum along one diagonal of the graph and a sharp transition along the other diagonal. This indicates that a proper translation along the x axis can partially compensate for a poor rotation around the y axis and vice versa. Note that the range of the plot for Figure (d) was slightly reduced to prevent plotting artifacts which were present when a larger range was used.

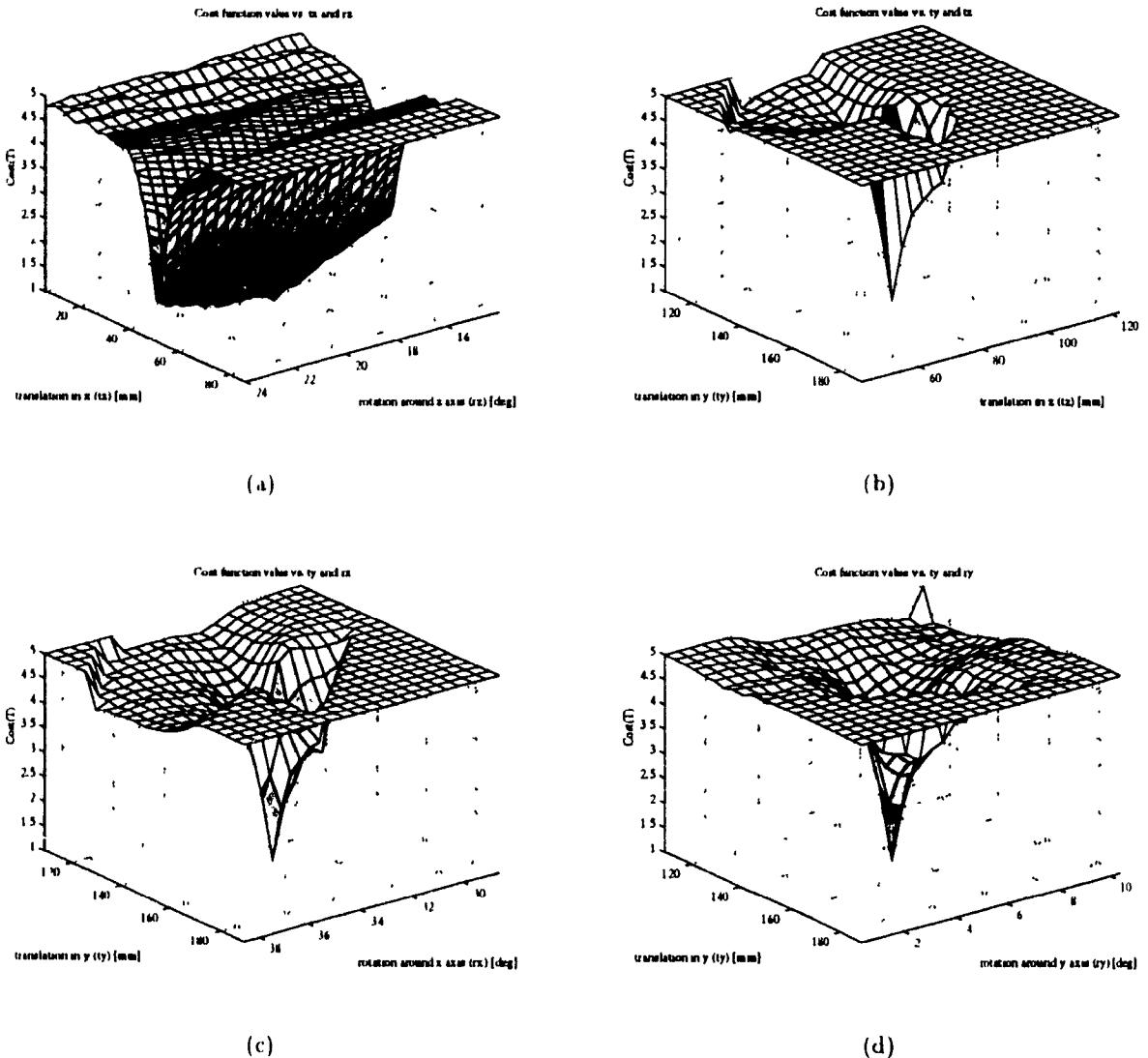


Figure 4.3: This figure shows various 3D slices of the cost function obtained for the two range views of Figure 4.1. Figure (a) exhibits a smooth transition to the global minimum for a variation of the z rotation. This is understandable since for a well centered range image, the z axis of the camera will go through the center of the image. Therefore, when the two range views are overlapping, if one of them is rotated around the z axis, the relative orientation of the views changes but an overlap between the views always remain. Figure (b), (c) and (d) all exhibit a sharp transition to the global minimum. For all plots, multiple local minima are present in a neighborhood around the global minimum. The flat sections of the plots correspond to the maximum value of the cost function (see legend of Figure 4.2)

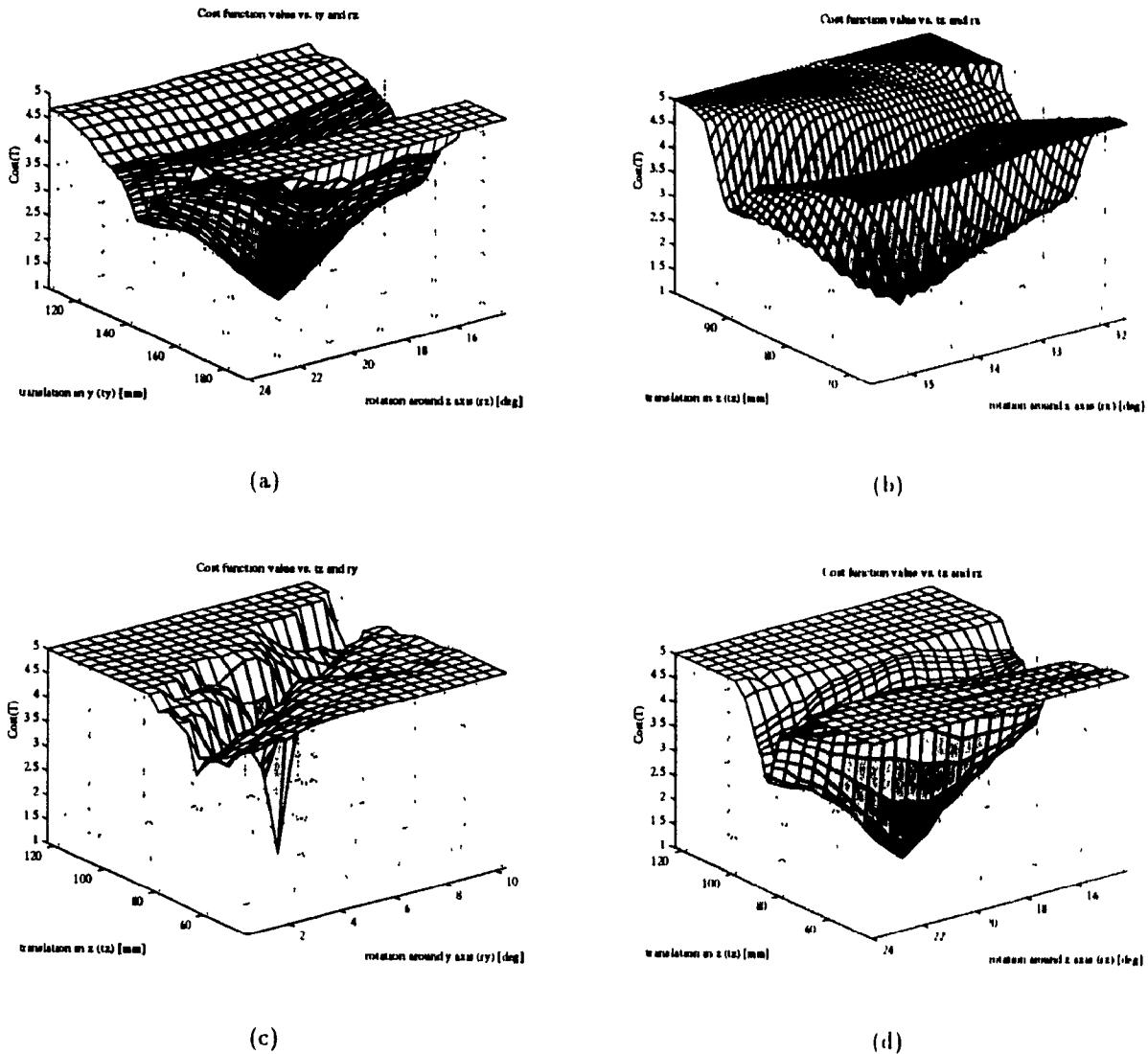


Figure 4.4: This figure shows various 3D slices of the cost function obtained for the two range views of Figure 4.1. In Figure (a) and (d), a smooth transition to the global minimum is present along the rotation around the z axis (see legend of Figure 4.3). Figure (b) exhibits a smooth variation to the global minimum along the diagonal of the graph. This indicates that a translation in z compensates for a rotation around the x axis and vice versa. Here again, the range of the plot was slightly reduced to avoid artifacts present when the larger range is used. Figure (c) has a sharp transition to the global minimum with many small local minima present in its vicinity. The flat sections of the plots correspond to the maximum value of the cost function (see legend of Figure 4.2).

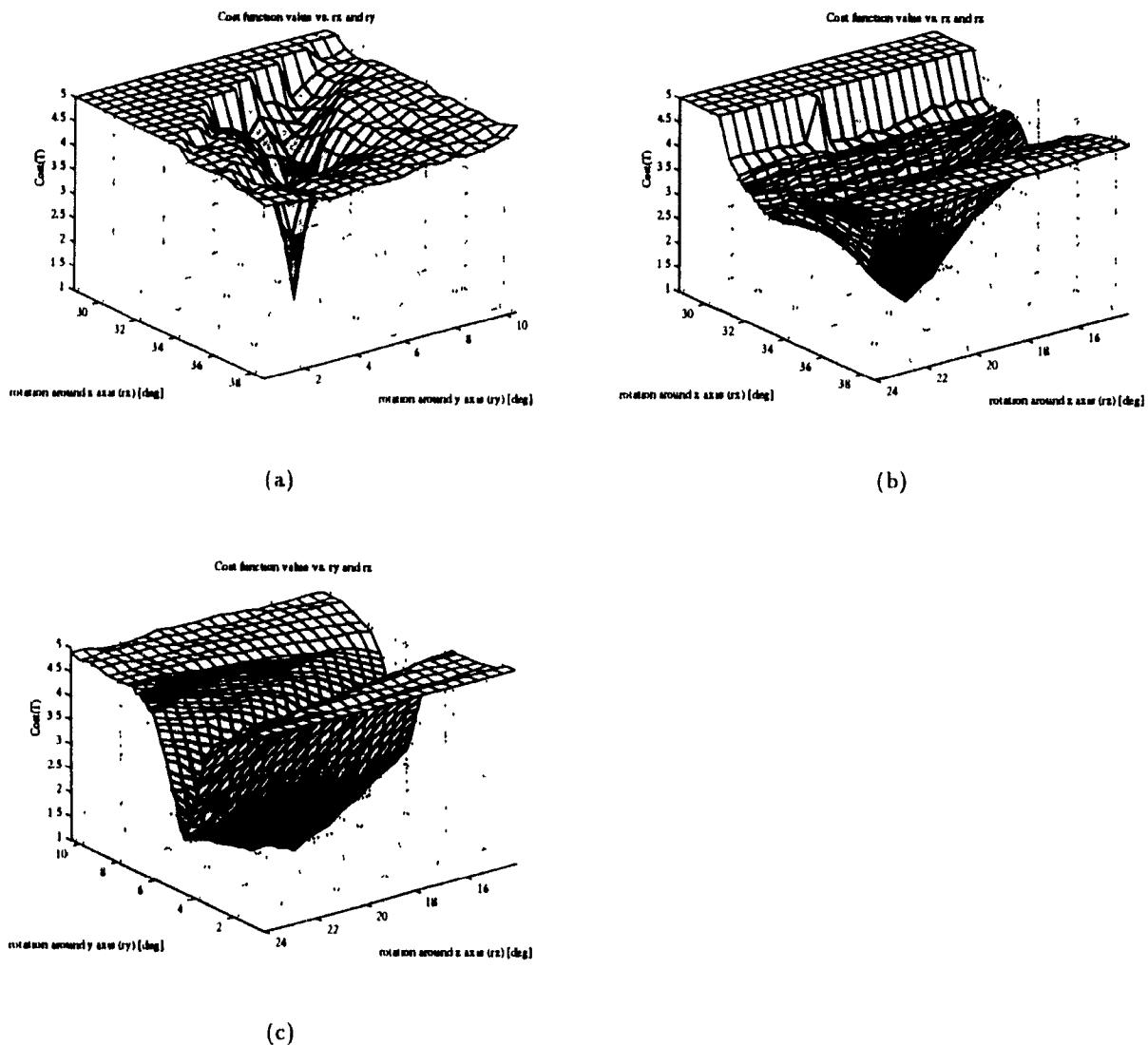


Figure 4.5: This figure shows various 3D slices of the cost function obtained for the two range views of Figure 4.1. Figure (a) exhibits a sharp transition to the global minimum with multiple local minima in its vicinity. This indicates that the registration is very sensible to a variation in the x or y rotation. As Figures (b) and (c) indicate, a rotation around the z axis produces a smooth variation to the global minimum (see legend of Figure 4.3). The flat sections of the plots correspond to the maximum value of the cost function (see legend of Figure 4.2).

minima and one large global minimum. In some cases (Figure 4.2 (a), (b), (c), 4.3 (b), (c), (d), 4.4 (c) and 4.5 (a)) a very steep minimum depression is present meaning that a slight variation of a parameter away from its optimum value results in a drastic deterioration of the registration fit. This is observed for all translation parameters t_x , t_y and t_z as well as the rotation parameters r_x and r_y . The only parameter whose variation results in a more gradual change in cost is r_z . This is understandable since the z axis of rotation passes directly through the center of the range images being matched. Because of this, varying r_z only causes the views to be oriented differently while still being superimposed. This behavior of the cost function can be observed in Figures 4.3 (a), 4.4 (a), 4.4 (d) and 4.5 (b), (c). In some cases, the variation of one parameter can be partially compensated for by the variation of another one. For example a rotation around the y axis can compensate for a translation in x . This can be seen in Figures 4.2 (d) and 4.4 (b) where a valley is present and diagonally oriented with respect to the rectangular plot range.

Because of discretization problems, some of the surfaces when plotted appeared to have multiple local minima when in fact they had one. In order to avoid this, more points were used to plot the cost function and the range of the parameters was reduced to keep the number of points in the graph at a reasonable level. This is the case for Figures 4.2 (d) and 4.4 (b); notice that the plot range is smaller in these two plots than in the others.

It can be noted that in all plots shown, the flat portion of the cost function corresponds to the maximum cost, meaning that all control points of the first view are more than the threshold distance away from their correspondence in the second view or that the overlap between the two views is less than the required overlap (See Section 3.5.1 for details).

The important aspect to observe about the different 3D slices of the cost function is that they are relatively convex in a neighborhood around the optimum point but many *bumps* or irregularities are present. As one gets away from the optimum point, the cost function eventually reaches a flat region corresponding to the maximum cost. The *bumps* in the cost function are mainly due to the irregular surface of the objects from which the range images are obtained and from the inherent noise introduced by the rangefinder during acquisition of the views. To a lesser degree, the discretization present in the range images and the subsampling performed to select control points also affect the shape of the cost function.

Because of these irregularities in the cost function, many small local minima will be present in most cases. Therefore, conventional search methods such as gradient descent

would fail to find the best transformation. More so if the initial transformation estimate obtained from the data acquisition stage is not close enough to the optimum transformation to start the descent in the convex region of the cost function. For example, if the transformation estimate is located in a region of the search space with maximum cost (the flat regions), then no gradient descent can take place because all the derivatives would be approximately zero. We must therefore rely on more robust search algorithms. Stochastic search methods such as genetic algorithms and simulated annealing have the property of being very robust in the presence of multiple local minima or maxima. They should therefore be suitable for performing the minimization of our cost function.

4.3 Genetic Algorithms

Genetic algorithms are optimization and search techniques based on evolution. They are extremely effective algorithms for solving very complex multivariable optimization problems and have generated interest in a wide range of engineering fields.

Genetic algorithms mimic the evolutionary process of life based on the Darwinian principles of reproduction and survival of the fittest. A Genetic Algorithm operates on a population of chromosomes where each chromosome represents a possible solution to the problem. New generations are created in succession starting from an initial population. A new generation of chromosomes is created by evaluating the fitness of each chromosome in the current generation and performing reproduction based on this fitness. Reproduction is proportional to fitness and is followed by crossover and mutation operations. Evolution proceeds until an adequate solution to the problem is found in the chromosome population. Figure 4.6 illustrates how these fundamental building blocks form the basis of genetic algorithms.

This section gives an overview of genetic algorithms and describes how they are used to perform the cost minimization search for our range image registration problem. We are not concerned about giving an in-depth description of the theory of genetic algorithms as their study is not the objective sought by this research. For more information on genetic algorithms, the reader may wish to refer to the specific literature available on the subject. For a good introduction to the different aspects of genetic algorithms see [6].

For the purposes of this research, a simple implementation of a genetic algorithm was used. Only the basic elements required in such an algorithm were implemented. As is described below, these basic elements consist of fitness-proportionate reproduction, the

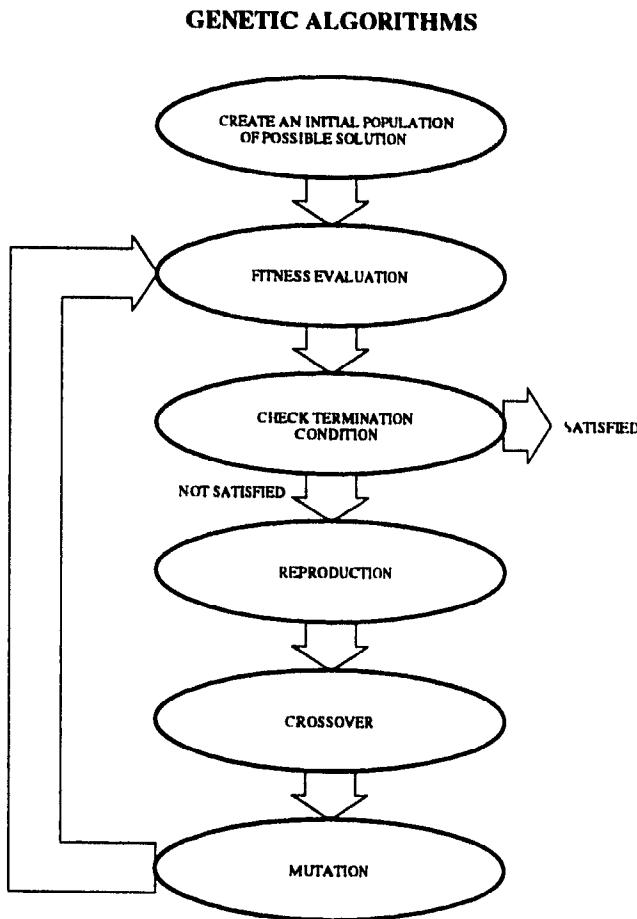


Figure 4.6: This figure shows how the various building blocks of a genetic algorithm are interrelated. The first stage of any genetic algorithm is the initialization of a population of candidate solutions to the problem one is trying to solve. After the evaluation of the fitness of every candidate solution in the population, a termination condition is verified. If this condition is satisfied in the current state of the population then the algorithm ends with the highest fitness transformation as the solution. If the termination condition is not satisfied, the algorithm continues to generate a new generation of candidate solutions. The following steps: reproduction, crossover and mutation are responsible for the evolution of the population. These steps are responsible for the creation of a new generation of solutions by manipulating the solutions in the current generation. The fitness evaluation ranks the different solutions in the population based on their ability to solve the problem. The reproduction step decides which solutions within the current population are fit enough to survive in the next generation and which ones should perish. The crossover and mutation steps are responsible for the exploration of new solutions to the problem. These steps create new solutions by mixing characteristics of different solutions in the population and by randomly altering some of these characteristics. This is done in order to eventually create a better solution to the problem.

crossover operation and the mutation operation. There are many variations of these, such as fitness scaling, multiple point crossover, etc. Goldberg [6] discusses some of them.

4.3.1 Chromosome Representation of the Registration Problem

In a genetic algorithm, chromosomes are used to represent candidate solutions of the problem to be solved. In the case of range image registration, a candidate solution is simply a rigid 3D transformation T and the problem we wish to solve is the minimization of the cost function defined in Chapter 3.

A binary encoding scheme is used to represent transformations as chromosomes. A transformation consists of 3 translation parameters: t_x, t_y, t_z and 3 rotation parameters: r_x, r_y, r_z . To make a chromosome, the motion parameters are binary encoded and concatenated to form a single string of 1's and 0's. Therefore, a chromosome is simply a string of 1's and 0's representing a 3D rigid transformation.

The motion parameters are real numbers and in order to represent them by a binary number, they must be quantized. The number of bits used to represent a parameter and the minimum and maximum values allowed for a parameter are the factors controlling how a parameter is binary encoded. Given a real parameter P we wish to produce its binary encoded version P_b . Given that we allow the parameter to take values between P_{min} and P_{max} and given that we wish to encode it using n bits we have

$$P_b = \left(\left\lfloor \left(P - P_{min} \right) \frac{2^n}{P_{max} - P_{min}} \right\rfloor \right)_2 \quad (4.1)$$

where $(\cdot)_2$ denotes base 2 (binary) representation and $\lfloor x \rfloor$ is the floor function returning the integer portion of argument x . Note that P_b is an integer number between 0 and $2^n - 1$. Also note that P_{min} and P_{max} delimit the search space for parameter P . The genetic algorithm will attempt to find the optimum value of P_b . Since all possible values of P_b correspond to a parameter value P between P_{min} and P_{max} , the optimum value of P found by the genetic algorithm will be within these boundaries.

A few observations can be made from Equation (4.1). The larger the number of bits n used to represent a parameter, the more finely quantized the search space for that parameter, and hence the more accurate the optimum parameter found by the genetic algorithm can be. By encoding a parameter using n bits, the search space for that parameter is subdivided

into 2^n equal increments corresponding to the quantization error. The quantization error is the largest error possible resulting from the parameter encoding. It is defined as follows:

$$\epsilon_{quantization} = \frac{|P_{max} - P_{min}|}{2^n} \quad (4.2)$$

Equation (4.2) can also be viewed as the step size between successive values of the discrete parameter P_b . In our case, knowing the size of the search space for all 6 motion parameters and knowing the desired accuracy for each parameter, Equation (4.2) can be used to determine the number of bits which should be used to encode each of the 6 parameters. The number of bits necessary to guarantee the accuracy of the search to be within ϵ units of the true optimum parameter value, given that the search space is Δ units in size, is defined by

$$n = \left\lceil \frac{\log \left(\frac{\Delta}{\epsilon} \right)}{\log 2} \right\rceil \quad (4.3)$$

For example, consider the binary encoding of the translation parameter t_x . Assuming that we know the optimum value of t_x to be somewhere between 34 and 47 millimeters, the size of the search space Δ_{t_x} for that parameter is therefore $47 - 34 = 13$ millimeters. Also, we want the optimum value of t_x found by the genetic algorithm to be within $\epsilon = 0.5$ millimeters of the true optimum value of t_x . Therefore, parameter t_x must be encoded in the chromosome using $n = \lceil \log(13/0.5)/\log 2 \rceil = 5$ bits. By using 5 bits, we can compute the quantization error defined by Equation (4.2) to be 0.40625 which is below 0.5 as desired.

Another observation which can be made from Equation (4.1) is that for a fixed number of bits n , the smaller the search space, the more accurate the solution can be. It is therefore important that the initial estimate T_e for the motion transformation be relatively accurate. In the case of an eye-in-hand system where the rangefinder camera is mounted on the end effector of a robot arm, the transformation estimate is obtained from the robot arm motion between the views. If the motion of the robot arm is not precise, the error in the estimate will be large and the size of the search space will have to be large enough to accommodate this. If the robot is very precise in its movements, the search space will be smaller. This will affect the time required for the genetic search. The smaller the search space, the less number of bits have to be used to obtain the same accuracy and therefore the faster the search can be performed.

4.3.2 Population Initialization

The first step involved in a genetic algorithm is the creation of an initial population. A population is a set of chromosomes, each representing a possible solution to the problem. The algorithm manipulates a population of chromosomes in order to search for the optimum registration transformation. To initiate the GA search, an initial population is created. This population consists of a number of different chromosomes representing different 3D transformations within the transformation search space.

The size of the population (i.e., the number of chromosomes in it) depends on a few factors. One factor is the extent of the search space. The size of the search space is fixed by the number of bits in a chromosome. If the chromosomes are composed of n bits, the search space has by definition a size of 2^n . For range image registration, the number of bits in a chromosome is determined from the search range of each motion parameter encoded in the chromosome and the desired accuracy for each parameter. The size of the population should reflect the size of the search space. For example, to be effective a larger search space may require a larger population.

Another factor which must be taken into consideration when determining the size of the initial population is the smoothness of the fitness function over the search space. If one suspects the fitness function to be very noisy and thus to have very many local maxima, then the size of the population should be large enough to accommodate this. If on the other hand the fitness function is very smooth over the search space, with only a few local maxima, then the size of the population can be made smaller without sacrificing search efficiency.

Computation time must also be taken into consideration when deciding on the size of the population. At each generation, the fitness function must be evaluated for every chromosome in the population and the reproduction, crossover and mutation operations must be performed. Therefore, the larger the population the more time the search will require.

The optimum size of initial population is usually determined by experimentation since it is usually very difficult to predict how the fitness function will behave over the search space. This will be discussed in greater detail in section 5.6.

The way the initial population of chromosomes is distributed over the search space can vary. The most common approach is to select the chromosomes randomly with a uniform

probability distribution over the search space. This can be done in a very simple manner. Given that the chromosomes are n bits in length and that we create an initial population of size P , we simply generate P uniformly distributed random numbers between 0 and $2^n - 1$.

For the image registration task, an initial transformation estimate is already available for the two range images that we wish to register. This estimate is only as accurate as the device used for moving the object or the camera during data acquisition. For example, if a turntable is used to rotate the object in front of a fixed rangefinder camera, then the accuracy of the transformation estimate is determined by the accuracy of the turntable and by the accuracy of the calibration of the camera position and orientation with respect to the turn table. If the object is fixed in space and the camera is manipulated by a robot arm, then the accuracy of the transformation estimate depends on the accuracy of the robot arm's displacements.

We assume a certain error for each motion parameter in the transformation estimate. This error in the transformation estimate determines the search range for the registration problem. The transformation estimate $T_e = (t_{xe}, t_{ye}, t_{ze}, r_{xe}, r_{ye}, r_{ze})$ is defined by 3 translation parameters and 3 rotation parameters. The error defined around this estimate can be written as $(\Delta t_x, \Delta t_y, \Delta t_z, \Delta r_x, \Delta r_y, \Delta r_z)$. This gives the following search range of valid transformations $T = (t_x, t_y, t_z, r_x, r_y, r_z)$:

$$\begin{aligned} t_{xe} - \Delta t_x &\leq t_x \leq t_{xe} + \Delta t_x \\ t_{ye} - \Delta t_y &\leq t_y \leq t_{ye} + \Delta t_y \\ t_{ze} - \Delta t_z &\leq t_z \leq t_{ze} + \Delta t_z \\ r_{xe} - \Delta r_x &\leq r_x \leq r_{xe} + \Delta r_x \\ r_{ye} - \Delta r_y &\leq r_y \leq r_{ye} + \Delta r_y \\ r_{ze} - \Delta r_z &\leq r_z \leq r_{ze} + \Delta r_z \end{aligned} \quad (4.4)$$

A precise estimate of the transformation restricts the size of the search range and therefore permits a faster search. If no transformation estimate were available, the search range for the rotation parameters would have to be large enough to make sure that the solution transformation is contained in it. The more accurate the estimate of the transformation, the faster the search will converge to the optimum solution.

4.3.3 Termination Condition

Once the fitness of all chromosomes in the population has been evaluated, we must determine whether the goal of the search has been reached. In certain applications, it is possible to measure how close to the goal the search has progressed. If the fitness value of the optimum solution is known, then the decision to stop the optimization process is simple to make: the process is stopped when it reaches a solution whose fitness value is close enough (within some predetermined range) to the optimum fitness value.

When the optimum goal of the search is unknown however, deciding when to stop the search can be a very complex problem. The goal of the registration process for example, is to find a transformation which perfectly registers the range images. With the genetic search, we are looking for a transformation which minimizes a certain cost function. We know what the end result should be, namely two registered range images. However, it is impossible to know beforehand what the value of the cost function is for the optimum transformation. We know that it should be the lowest cost value possible, but since this value is unknown we cannot determine how far from the optimum solution we are. Also, the optimum cost/fitness value will vary with the range images being registered. We therefore cannot base the termination condition solely on the fitness value.

Nevertheless, we must define a criterion for stopping the genetic search. Through experimentation, it can be observed that an acceptable solution is reached after a certain number G of generations has elapsed. The genetic search can thus be terminated after G generations with reasonable certainty that the best solution reached up to that point is acceptable. The choice of G will be discussed further in chapter 5.

The termination condition box of Figure 4.6 then, simply does the following verification to decide whether or not the search should continue:

If Number of Generations so far > G Then

Stop Genetic Search Process

Endif

4.3.4 Fitness Evaluation

Genetic algorithms cannot be used to directly minimize the cost function because they are inherently designed to maximize fitness. In order to minimize a function, the minimization

problem must be converted to a maximization problem so that a transformation yielding a low cost value has a high fitness value and conversely.

The transformations encoded by the chromosomes in the population are evaluated with the cost function defined by Equation (3.34). To convert this cost value into a fitness value, we first determine the chromosome with the maximum cost present in the population. Then we define the fitness of a chromosome as the difference between its cost value and the maximum cost value. This will give a fitness value of 0 to chromosomes having the maximum cost value, and will give the maximum fitness value to chromosomes with the smallest cost value. The fitness of a given chromosome c_k in population P is defined by the following equation:

$$\text{Fitness}(c_k) = \max_{c_i \in P} \{\text{cost}(T_{c_i})\} - \text{cost}(T_{c_k}) \quad (4.5)$$

where T_{c_k} is the 3D rigid transformation specified by chromosome c_k and $\text{cost}()$ is the cost function defined in Equation (3.37).

4.3.5 Fitness-Proportionate Reproduction

As was shown in Figure 4.6, the genetic algorithm conducts and regulates the evolution of the population until a proper solution is reached. Starting with the initial population, new populations are derived consecutively from the previous one. Once the fitness of all the chromosomes in the population has been evaluated, if the termination condition is not met, a new generation of chromosomes is created. This new population will be the same size as the current one and the chromosomes in it will be modified versions of the chromosomes present in the current population. The algorithm generates a new population of transformations by altering the current one through a specific sequence of operations.

The first operation is referred to as the *fitness-proportionate reproduction*. The reproduction step of the genetic algorithm is responsible for deciding which chromosomes currently present in the population are sufficiently viable to be reproduced into the new generation.

The viability of any given chromosome is based on its fitness value. When deciding which chromosomes are fit enough to survive into the next generation, chromosomes corresponding to high fitness transformations will be favored over those with low fitness transformations so

that at each generation, the overall fitness of the population increases. Thus, the more fit a chromosome is, the more likely it will survive to the next generation. When a chromosome is chosen to survive it is simply copied into the new population. In other words, the string of 1's and 0's forming the chromosome is replicated in the new population.

The method used for the reproduction is referred to as the *roulette wheel selection* method. In this method, every chromosome is allocated a portion of a wheel. The size of the portion allocated to a chromosome is proportional to its fitness. A chromosome with a high fitness value will be given a large section of the roulette wheel whereas a low fitness chromosome will only be given a slim section of the roulette wheel. To determine which chromosomes should be replicated in the new generation, the wheel is spun. The chromosome whose allocated section is designated when the wheel stops spinning is the winner. The winning chromosome gets copied into the next generation population. The process is repeated until the number of chromosomes copied is equal to the desired population size. If the size of the population is P , then the wheel will be spun P times, once for every chromosome in the population.

The portion of the total wheel allocated for a given chromosome is simply the fitness value obtained for that chromosome divided by the total fitness value of the population. The total fitness value of the population is the sum of the fitness value for all the chromosomes in the population. Give a population of size P , Equation (4.6) defines the ratio r_k of space allocated on the roulette wheel to the total wheel space for chromosome c_k :

$$r_k = \frac{\text{Fitness}(c_k)}{\sum_{i=1}^P \text{Fitness}(c_i)} \quad (4.6)$$

With the roulette wheel selection mechanism, the probability of reproduction of a chromosome is directly proportional to its fitness value. Chromosomes with high fitness will have a high probability of reproduction and chromosomes with low fitness value will have a low probability of reproduction. Note that if all chromosomes have the same cost value, then they will all have a fitness of 0. This renders the ratio definition in Equation (4.6) to be undefined. When this condition occurs it is handled by simply allocating each chromosome an equal portion of the roulette wheel. Since all chromosome have the same fitness value, they are given an equal probability of reproduction.

A random number generator is used to implement the spinning of the roulette wheel.

A number r is generated randomly between 0 and 1 with a uniform distribution. The ratio values r_k of the chromosome are added in sequence until the sum exceeds r . The chromosome whose ratio was added last is the winning chromosome and gets reproduced in the new population. Given a random number r uniformly distributed in the interval $[0, 1]$, chromosome k will be the chosen chromosome among a population of P chromosomes where k is the minimum integer value satisfying the following:

$$\sum_{i=1}^k r_i > r \quad (4.7)$$

This random process is repeated P times (P being the size of the population), once for every chromosome to be reproduced in the new population.

4.3.6 Crossover Operator

Once a new generation of chromosomes has been created through the reproduction process (using roulette wheel selection), the chromosomes in the new population are transformed so that new regions of the search space can be explored. The chromosomes are transformed by a crossover and a mutation operator.

The crossover operator takes bits and pieces of chromosomes within the population and recombines them to form new chromosomes. This has the effect of mixing the characteristics of different transformations into new ones. The children chromosomes produced in this fashion replace the parent chromosomes in the population. The hope is that the transformation defined by a child chromosome will yield a higher fitness (a better registration) than the transformations of the parent chromosomes from which its parts came from.

Two parent chromosomes are required for crossover and two child chromosomes are created as a result. These new chromosomes share the characteristics of the two parent chromosomes in that they are made up of binary substrings of each parent. The parent chromosomes are chosen at random from the new population, but since the reproduction step has already introduced a random ordering of the chromosomes in the new population, the parent chromosomes can simply be chosen two at a time in sequence.

Not all parent chromosomes will necessarily undergo crossover. A crossover probability parameter μ_c controls the frequency at which crossover occurs. For each set of parent chromosomes chosen, a decision is made with probability μ_c of success as to whether to

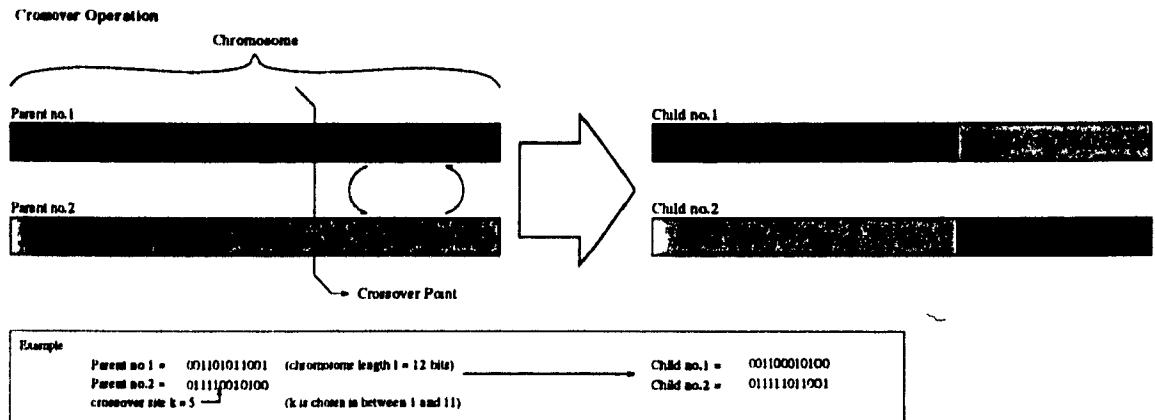


Figure 4.7: This figure shows how crossover is performed on two parent chromosomes. First, a crossover site is chosen randomly along the length of the chromosome. Then the sections of chromosomes past the crossover site are interchanged between the two parents to create two child chromosomes.

perform crossover or not. However, because the mixing of chromosomes is responsible for most of the search, the crossover probability is usually very high, a typical value is $\mu_c = 1$ meaning that crossover always occur.

The crossover operator produces a new chromosome by mixing parts of one parent chromosome with parts of the other parent. The crossover method employed in this research is referred to as *single point crossover*. It is done as follows: First, a crossover site is selected at random. A crossover site is any position along the chromosome string where it can be split into two substrings. We represent the crossover site as an integer position k in the chromosome string. The site k is thus chosen at random between 1 and the length of the chromosome minus 1. For a chromosome of l bits in length, k is chosen at random in the interval $[1, l - 1]$. The two child chromosomes are then created by swapping all bits from position $k + 1$ to l inclusively. Figure 4.7 illustrates this operator.

4.3.7 Mutation Operator

Following crossover, a mutation operation is performed on the population. As Goldberg [6] mentions, the role of the mutation operation is secondary in a genetic algorithm. The bulk of the search is carried out by the reproduction and crossover stages. The mutation operation allows the exploration of regions of the search space which would not be reached by reproduction and crossover alone. Another way of looking at mutation is as a safeguard against the premature loss of potentially useful genetic information.

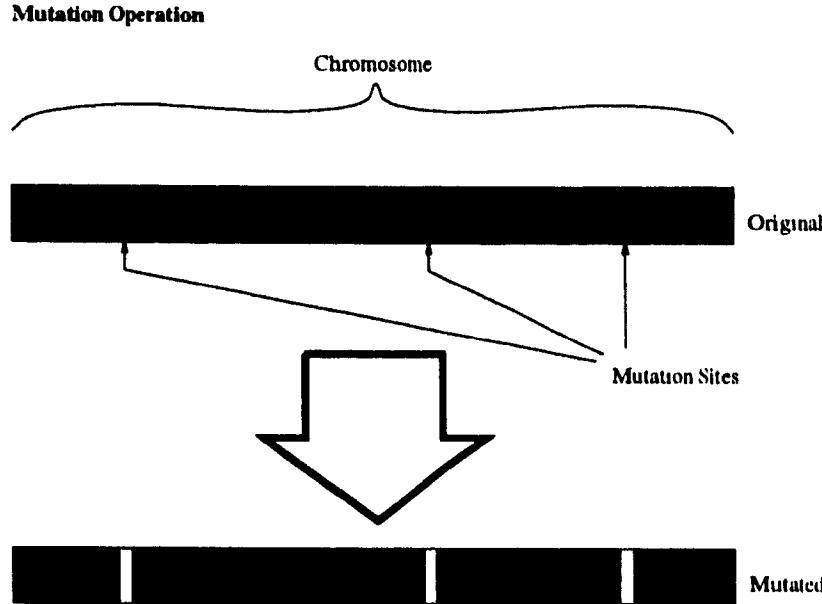


Figure 4.8: This figure shows how mutation is performed on a chromosome. For each bit inside the chromosome a decision is made whether the bit should undergo mutation or not. All bits have the same probability of being mutated. The mutation probability is usually fairly small to ensure that mutation does not completely disrupt the chromosome. When a bit mutates, it is simply toggled. If the bit is 0 it is set to 1, if it is 1 it is set to 0.

With reproduction and crossover alone, some genetic information can be lost as a population evolves. For example, a 1 or 0 at a particular location in the population could be lost (i.e., all the chromosomes in the population have the same bit value at that location). Crossover alone cannot recover this lost information. If this particular genetic characteristic happens to be crucial to the solution, we must ensure that there exists some mean of introducing it in the population. This is the purpose of the mutation operator.

The mutation operator acts by randomly altering bits in the chromosome string. A mutation probability μ_m specifies the frequency at which mutation occurs in the population. This frequency is usually in the order of one mutation per thousand bits in the population so a typical value is $\mu_m = 0.001$. To perform the mutation operation, the bits of all chromosomes in the new generation are taken one at a time and a decision is made as to whether it should undergo mutation or not. A bit will mutate with probability μ_m . The mutation of a bit simply consists in its value being changed from a 0 to a 1, or from a 1 to a 0 (i.e., toggle the bit value). Figure 4.8 illustrates the mutation of a chromosome.

The simplest implementation of mutation is as follows: for each bit of each chromosome

in the population, generate a random number r uniformly distributed between 0 and 1. If $r > \mu_m$ then toggle the value of the bit.

However, it is not very efficient to test all the bits in the population one at a time for a mutation that occurs maybe once every thousand bits. As an alternative, the mutation can be viewed as a geometric process where an event may occur at every step with probability μ_m . Therefore the number of bits n between mutation events has an exponential distribution function. Instead of testing every bit, we can compute the number of bits before a mutation occurs. Considering all the bits of all the chromosomes in the population in sequence, let P_i be the probability that the next mutation occurs at bit position i starting from position 1. The probability that the current bit (at position 1) is mutated is $P_1 = \mu_m$. The probability that the mutation occurs on the second bit is $P_2 = (1 - \mu_m)\mu_m$, the third bit $P_3 = (1 - \mu_m)^2\mu_m$, etc. In general, the probability that the mutation occurs at bit position k is $P_k = (1 - \mu_m)^{k-1}\mu_m$. We know that the sum of all probabilities is equal to 1.

$$\sum_{i=1}^{\infty} P_i = \sum_{i=1}^{\infty} (1 - \mu_m)^{i-1} \mu_m = 1 \quad (4.8)$$

Let M be the number of bits until the next mutation and let r be a random number uniformly distributed between 0 and 1. By summing the P_i terms of Equation (4.8) until a value r is reached, it can be shown that if r is uniformly distributed then the number of terms required has an exponential distribution. We need to satisfy the following:

$$\sum_{i=1}^M P_i = \sum_{i=1}^M (1 - \mu_m)^{i-1} \mu_m = 1 - (1 - \mu_m)^M = r \quad (4.9)$$

Solving for M and replacing r by $1 - r$ (i.e., if r is uniformly distributed between 0 and 1, then so is $1 - r$), we get a solution for M where M is exponentially distributed with parameter μ_m .

$$M = \left\lfloor \frac{\log(r)}{\log(1 - \mu_m)} \right\rfloor \quad (4.10)$$

Using Equation (4.10), the mutation process is performed as follows:

Consider all bits in the population as a single bit string

Bit position = 1 (start on the first bit position)

Repeat

```

compute  $M$  using Equation (4.10)
Bit position = Bit position +  $M$ 
If Bit position  $\leq$  number of bits in population
    toggle the bit value at Bit position
Endif

Until Bit position > number of bits in population

```

This concludes the overview of Genetic Algorithms. We have seen how the minimization problem can be formulated so that a genetic algorithm search can be used to effectively perform the range image registration task. Various techniques exist for optimizing non-convex cost functions. Genetic Algorithms is one. Another optimization method which performs well in the presence of a non-convex cost function is Simulated Annealing. Different variations of Simulated Annealing have been developed and in particular, we will look at the Very Fast Simulated Reannealing version. The next section presents this technique in details.

4.4 Very Fast Simulated Reannealing

In this section we will give a brief description of the concepts involved in Very Fast Simulated Reannealing (VFSR) which is an extension of the Simulated Annealing (SA) optimization technique. In order to have a better understanding of VFSR, it is appropriate to examine its origins in Simulated Annealing. The first part of this section is thus an overview of Simulated Annealing.

VFSR has emerged from successive improvements made on the original Simulated Annealing algorithm. These previous versions of Simulated Annealing, Boltzmann Annealing (regular Simulated Annealing) and Cauchy Annealing (Fast Simulated Annealing), will be examined following the overview section. This will provide a gradual progression to Very Fast Simulated Reannealing which will be presented in the last section.

4.4.1 Overview of Simulated Annealing

Simulated Annealing is an optimization method based on the physical process of cooling studied in statistical mechanics. Statistical mechanics is the central discipline of condensed matter physics and is concerned with the analysis of aggregate properties of large number

of atoms in liquid or solid matter [14]. The idea was first introduced in 1953 by Metropolis *et al.* [16]. They presented a simple algorithm for simulating a collection of atoms in equilibrium at a given temperature.

Optimization methods using the concept of annealing as the basis of their search are relatively recent. The first such optimization method devised was simple Simulated Annealing or Boltzmann Annealing. It was developed to solve a variety of problems involving the minimization of a non-convex cost function (e.g., finding the optimal wiring for a densely wired computer chip) [14]. One of the interesting aspects of this optimization method is the proof that, under certain conditions, the algorithm converges in a statistical sense to the optimum solution. The next variation of this algorithm is Fast Simulated Annealing or Cauchy Annealing where the convergence time is considerably reduced while keeping the statistical convergence property of the original algorithm. Very Fast Simulated Reannealing is yet an improvement on the convergence time of Fast Simulated Annealing.

Given a cost function we wish to minimize, the SA algorithm will probe the search space sequentially based on the current state temperature and the energy specified by the cost function.

We have a cost function $E(s)$ representing the energy of a system when it is in state s . The goal of the optimization search is to find the state s of the system which minimizes its energy E . In the case of range view registration, the state of the system is defined by the rigid 3D transformation $T = (t_x, t_y, t_z, r_x, r_y, r_z)$ and the energy of the system is given by the cost function, $\text{cost}(T)$, defined by Equation (3.37) in Chapter 3. The cost function is at a minimum when the two range views are registered by the transformation T . The following illustrates the procedure performed by the Simulated Annealing algorithm for minimizing the cost function:

The major components involved in Annealing search methods are an Energy or Cost function $E(s)$ to be minimized, a generating function $g()$ used to create new states of the system and allowing for the probing of the search space and finally, an acceptance function $h()$ used to decide whether a new state generated by $g()$ is acceptable or not as the new current state of the system. To simulate the cooling process involved in annealing, a temperature variable is used. Starting from an initial value, the temperature is reduced gradually and this temperature affects the behavior of both the generating function and the acceptance function.

When the search is initiated, the temperature of the system is set to an initial value and the current state of the system is set to an initial state value. A new state s_{k+1} is generated from the current state s_k and from the current temperature of the system $T(k)$ where k represents the annealing discrete time interval. As k increases, the temperature decreases. At high temperatures, the new state s_{k+1} created by the generating function $g()$ can be virtually anywhere in the search space. This is due to the fact that at high temperatures, the probability density function defined by the generating function is very widespread around the current state of the system. Therefore, the probability that the new state be far away (in the search space domain) from the current system state, is high.

To decide whether a new state is accepted or not, an acceptance function $h()$ is used. The acceptance function is a function of the current temperature T and of the energy difference ΔE between the current state s and the generated state. A new state resulting in a lower energy value is automatically accepted. If the energy value of the new state is greater than that of the current state, then it can be rejected based on a probabilistic criterion. The greater the increase in energy between the current state and the next state, the less likely this new state will be accepted. At high temperatures, the acceptance function more or less accepts all new states. This, in conjunction with the widespread generating function, effectively results in a random probing of the search space at high temperatures. However, as the temperature is lowered, the capacity of a new state to be accepted is reduced. At low temperatures, the probability of accepting a new state which results in an increase in energy is extremely small. When the annealing stops, the lowest energy state visited during the search is taken as the solution of the minimization. The following is a brief step by step look at the annealing minimization process:

Set the starting temperature T [$T = T_0$]

Set the initial state of the system s [$s = s_0$]

Initialize the time [$k = 0$]

Compute the Energy/Cost function value of the initial state [$E = E(s)$]

Repeat

Repeat

- Generate a new state for the system

using the generating function [$s' = s + g(s, T)$]

- Compute the Energy/Cost function value

of the new state [$E' = E(s')$]

- Compute the Energy difference between the current and new states [$\Delta E = E' - E$]
- Decide whether or not the new state is accepted using the acceptance function [compute $h(\Delta E, T)$]

Until obtain an accepted new state

Set the new state to be the current state [$s_{k+1} = s'$]

Update the annealing temperature [$T = T(k + 1)$]

Update the annealing time [$k = k + 1$]

Until a given time or a given temperature is reached

This section described Simulated Annealing in a general sense. Various generating functions have been used with this general algorithm resulting in different convergence rates. We will now look at various details relevant to these different SA algorithms. The first algorithm examined in detail is the original Simulated Annealing algorithm or Boltzmann Annealing, followed by Fast Simulated Annealing also referred to as Cauchy Annealing. The last section examines the Very Fast Simulated Reannealing method.

4.4.2 Boltzmann Annealing

The original annealing method developed is often referred to as the Boltzmann algorithm and uses a Gaussian probability density for the generating function.

$$g(\Delta \mathbf{x}) = (2\pi T)^{-\frac{D}{2}} \exp\left(\frac{-\Delta \mathbf{x}^2}{2T}\right) \quad (4.11)$$

where T is the current temperature of the system, $\Delta \mathbf{x}$ is the variation in the state of the system and D is the dimension of the state vector \mathbf{x} . For example, in the case of range image registration, the state vector consists of a transformation $(t_x, t_y, t_z, r_x, r_y, r_z)$ containing six motion parameters; thus $D = 6$. Considering the 1D case, we can plot the generating function for various temperature values (Figure 4.9).

The acceptance function $h()$ is used to determine the probability of acceptance of a new state with energy E_{k+1} compared with the current state energy value E_k . It is used to decide whether a new state generated by the generating function is accepted or not.

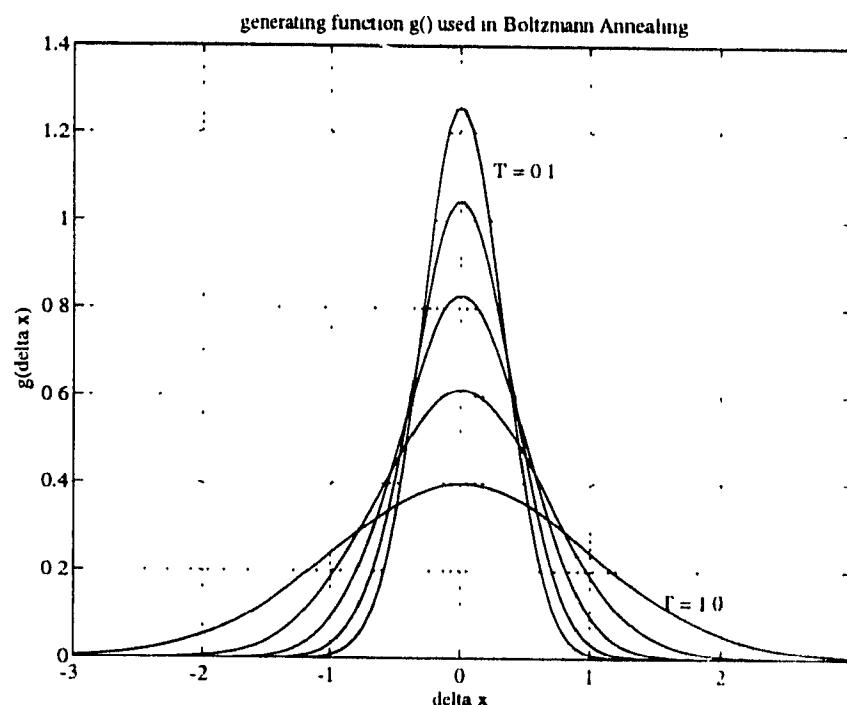


Figure 4.9: The shape of the Boltzmann distribution function used for the generating function in Simulated Annealing is shown for various temperature values. At high temperatures ($T=1.0$), corresponding to the initial stage of the Simulated Annealing search, the shape of the generating function is broader, thereby allowing for a wider search. As the temperature reduces, the search is gradually confined to a smaller and smaller neighborhood Δx around the current state x of the system.

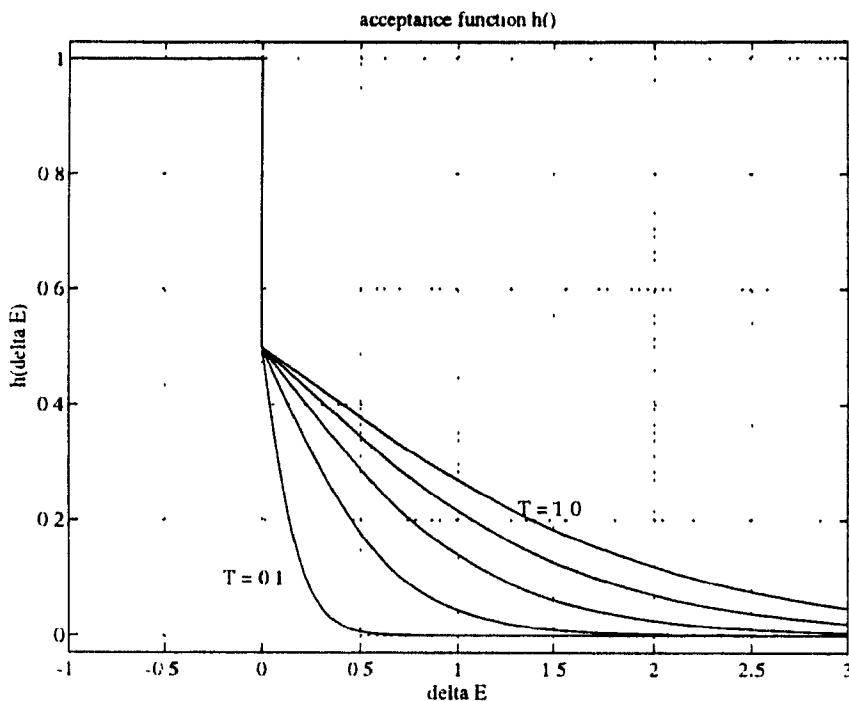


Figure 4.10: This figure shows the shape of the acceptance probability function $h()$ as a function of the energy difference ΔE between the new generated state and the current state of the system. The function is plotted for different temperature values.

The acceptance $h()$ is a function of the energy difference ΔE between the current and new system state and of the current temperature T of the system.

$$h(x) = \frac{\exp(-E_{k+1}/T)}{\exp(-E_{k+1}/T) + \exp(-E_k/T)} = \frac{1}{1 + \exp(\Delta E/T)} \quad (4.12)$$

A plot of the acceptance function is shown in Figure 4.10. The function is plotted for various temperature values.

From the plots of $h()$ shown in Figure 4.10, we see that at high temperatures, corresponding to the initial stage of the Simulated Annealing search, there is a large probability of accepting a new state even though the total energy value of the system may be raised as a result (ΔE positive). For example, at $T_0 = 1.0$, the probability of accepting a new state which increases the energy by 1 is approximately 0.27, whereas at a temperature of $T = 0.1$, this probability is virtually nil. This shows that during the initial search stage, the algorithm can easily move out of local minima. As the search progresses and the temperature is lowered, only states that lower the energy of the system have a reasonable chance of

being accepted. Any state increasing the energy of the system has a very low probability of being accepted.

The interesting property of Simulated Annealing is that, given the generating function $g()$, it has been proved [7, 12] that the algorithm converges in a statistical sense to the global minimum of the energy function $E()$ we wish to minimize provided the temperature is lowered no faster than

$$T(k) = \frac{T_0}{\ln k} \quad (4.13)$$

This so-called cooling schedule has the disadvantage of being very slow. For example, starting at an initial temperature of $T = 1.0$, if one desires to stop the annealing search when $T = 0.01$, then 2.69×10^{43} annealing steps k are required. Knowing that for every step the energy/cost function may have to be evaluated many times (more as the temperature drops and new states are not accepted easily), then it is clear that a large number of cost function evaluations will be required to perform the minimization. The objective then is to accelerate the cooling schedule while maintaining still the guarantee that the algorithm will statistically converge to the global minimum.

4.4.3 Fast Simulated Annealing

It has been noted that the Simulated Annealing algorithms do not need to be restricted to the Boltzmann generating function defined in Equation (4.11), but other generating functions can also be employed. Fast Simulated Annealing uses a different generating function $g()$ that allows the temperature to be reduced at a much faster rate than Boltzmann annealing while preserving the convergence aspect of Boltzmann annealing. Fast Simulated Annealing is also referred to as Cauchy Annealing because of the use of the Cauchy distribution for the generating function:

$$g(\mathbf{x}) = \frac{T}{(\Delta \mathbf{x}^2 + T^2)^{(D+1)/2}} \quad (4.14)$$

As can be seen in Figure 4.11, the properties of this generating function are similar to the Boltzmann generating function shown in Figure 4.9. Both permit a wide sampling of the search space during the initial stage of the annealing search and both gradually restrict the allowed state variation as the temperature is lowered. The main difference is that

the Cauchy distribution has a much wider tail than the Boltzmann distribution. This has the effect of permitting a larger extent of the search at the beginning of the minimization and thus allowing a better coverage of the local minima present in the search space. As indicated above, this generating function also allows the use of a faster cooling process. For Fast Annealing, the temperature function given in Equation (4.13) can be replaced by

$$T(k) = \frac{T_0}{k} \quad (4.15)$$

and still maintain the assurance that the search will statistically converge. This cooling process is clearly faster than the cooling specified for SA by Equation (4.13). For example, starting with an initial temperature of $T_0 = 1.0$, and a final temperature of $T = 0.0001$, only 1000 annealing steps are required.

The acceptance function $h()$ used in Cauchy Annealing is the same as for Boltzmann Annealing.

4.4.4 Very Fast Simulated Reannealing

In 1989, Ingber [12] published a paper in which he described his work on Very Fast Simulated Reannealing (VFSR). Along the same lines as Fast Annealing, VFSR uses a different generating function which allows it to accelerate the cooling process even more while maintaining the statistical convergence property of the original algorithm. Both SA and FA sample an infinite search space. For VFSR, Ingber took a different approach. Say we are minimizing an energy/cost function $E()$ defined over a D dimensional domain. Let $\vec{\alpha}$ be a point in that domain; $\vec{\alpha}$ is thus a D -dimensional vector with parameters α_i ($i = 1 .. D$). When performing a search, one often has a realistic knowledge of the parameter constraints. VFSR uses this fact to limit the search space so that at any annealing time k , we have

$$\alpha_{i,k} \in [A_i, B_i] \quad (4.16)$$

where A_i and B_i define the constraints on parameter α_i . The value of this parameter is calculated with a random variable y_i as follows:

$$\begin{aligned} \alpha_{i,k+1} &= \alpha_{i,k} + y_i(B_i - A_i) \\ y_i &\in [-1, 1] \end{aligned} \quad (4.17)$$

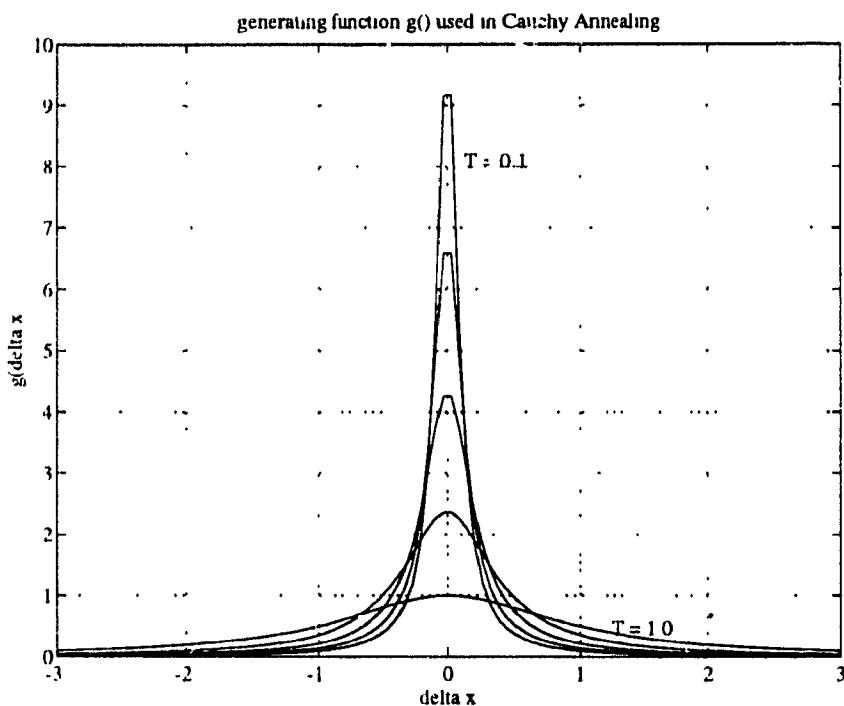


Figure 4.11: The shape of the Cauchy distribution function used for the generating function in Fast Simulated Annealing is shown for various temperature values. At high temperatures ($T=1.0$), corresponding to the initial stage of the Simulated Annealing search, the shape of the generating function is broader thus allowing for a wider search. As the temperature reduces, the search is gradually confined to a smaller and smaller neighborhood Δx around the current state x of the system. Compared with the Boltzmann distribution function in Figure 4.9, the Cauchy distribution function has a much wider tail allowing for a better coverage of the search space.

The generating function is then defined as

$$g(y) = \prod_{i=1}^D \frac{1}{2(|y_i| + T_i) \ln(1 + 1/T_i)} \equiv \prod_{i=1}^D g_i(y_i) \quad (4.18)$$

The y_i 's are generated from a uniformly distributed random variable $u_i \in U[0, 1]$, by

$$y_i = \text{sgn}\left(u_i - \frac{1}{2}\right) T_i \left[(1 + 1/T_i)^{|2u_i - 1|} - 1\right] \quad (4.19)$$

From Equation (4.18), we can see that all D parameters can be generated independently with their own individual temperature values. This has the advantage that the cooling process can be different for each parameter and thereby account for the distinct sensitivities of the parameters over the energy/cost function. A plot of the generating function given by Equation (4.18) is shown in Figure 4.12. In Figure 4.13, all three generating functions are plotted to show the difference in shapes between them.

With the generating function defined in Equation (4.18), the annealing schedule of VFSR which still ensures that the search will statistically converge to the global minimum is given by

$$T_i(k) = T_{i,0} \exp(-c_i k^{1/D}) \quad (4.20)$$

where c_i is a user defined parameter. Given an initial temperature $T_{i,0}$, a desired final temperature $T_{i,f}$ and a desired number of iterations k_f , it is suggested that the value of c_i be chosen to satisfy the following conditions:

$$\begin{aligned} T_{i,f} &= T_{i,0} \exp(-m_i) \text{ when } k_f = \exp(n_i) \\ c_i &= m_i \exp(-n_i/D) \end{aligned} \quad (4.21)$$

The parameters m_i and n_i can be considered as "free" parameters for tuning VFSR for specific problems.

To better view the relative speed of convergence of the different annealing schedules for SA, FA and VFSR, Figure 4.14 shows plots of all three temperature schedules. The temperatures are plotted for k varying from 0 to 50, an initial temperature of $T_0 = 1.0$ is used, and the parameter dimension D is set to 1. A value of $c_i = 0.1842$ is used to plot Equation (4.20), corresponding to a final temperature of $T_{i,f} = 0.0001$ and final number

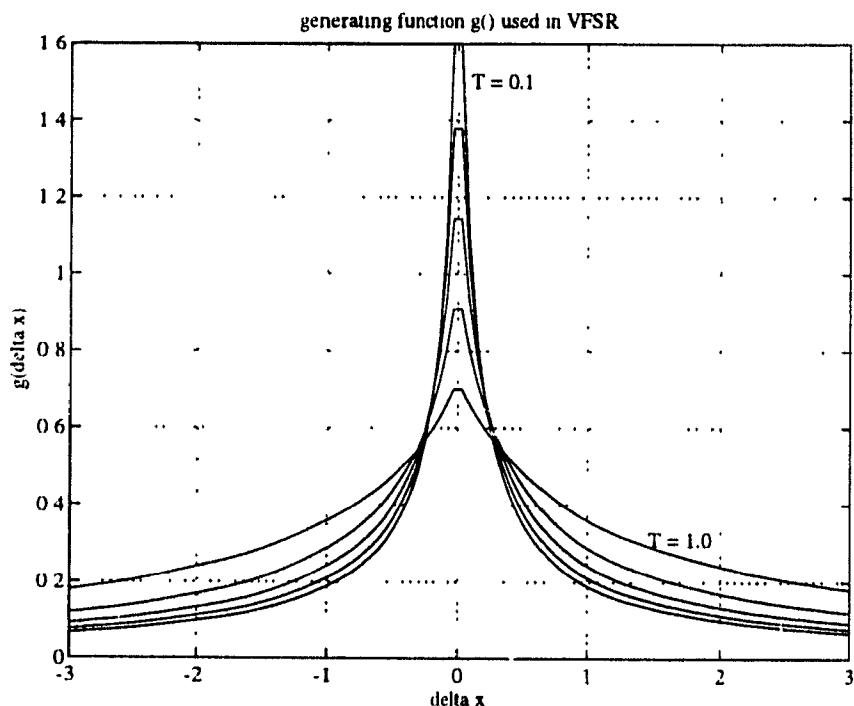


Figure 4.12: The shape of the distribution function used for Very Fast Simulated Reannealing is shown for various temperature values. At high temperatures ($T=1.0$), corresponding to the initial stage of the Simulated Annealing search, the shape of the generating function is broader, thus allowing for a wider search. It is close to being flat which would in fact result in a uniform distribution corresponding to a purely random search. As the temperature reduces, the search is gradually confined to a smaller and smaller neighborhood Δx around the current state x of the system. Compared with the Boltzmann distribution function in Figure 4.9, and to the Cauchy distribution function in Figure 4.11, the VFSR distribution function has a much wider tail allowing for a better coverage of the search space. It also narrows down at a much faster rate than its counterparts.

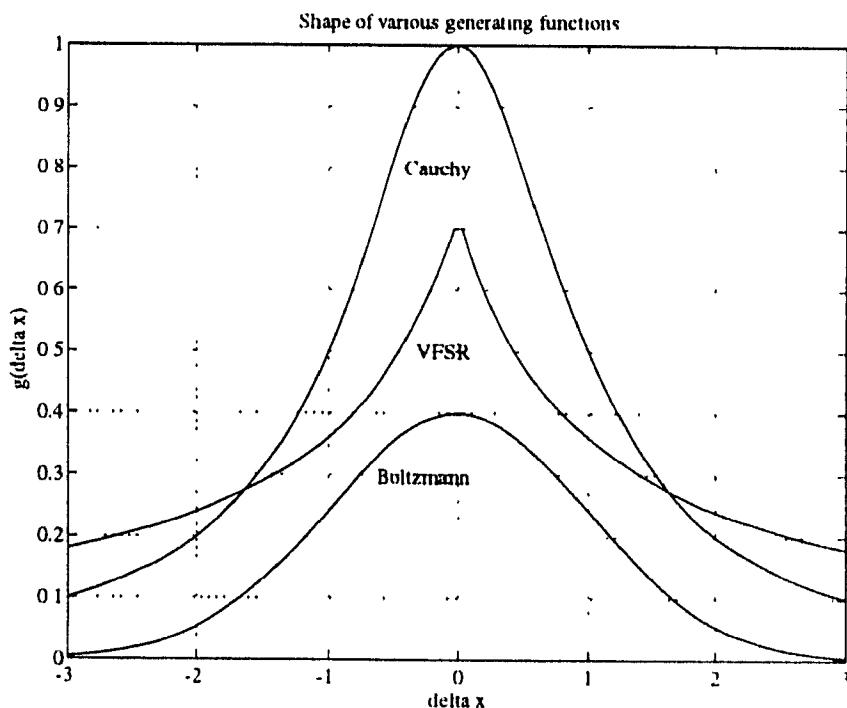


Figure 4.13: This figure shows the three generating functions – the Boltzmann, Cauchy and VFSR distributions. The VFSR generating function has the greatest tail of the three for a better coverage of the search space. All curves were plotted for one dimension $D = 1$, a temperature of $T = 1.0$ and a parameter variation of Δx between -3 and 3.

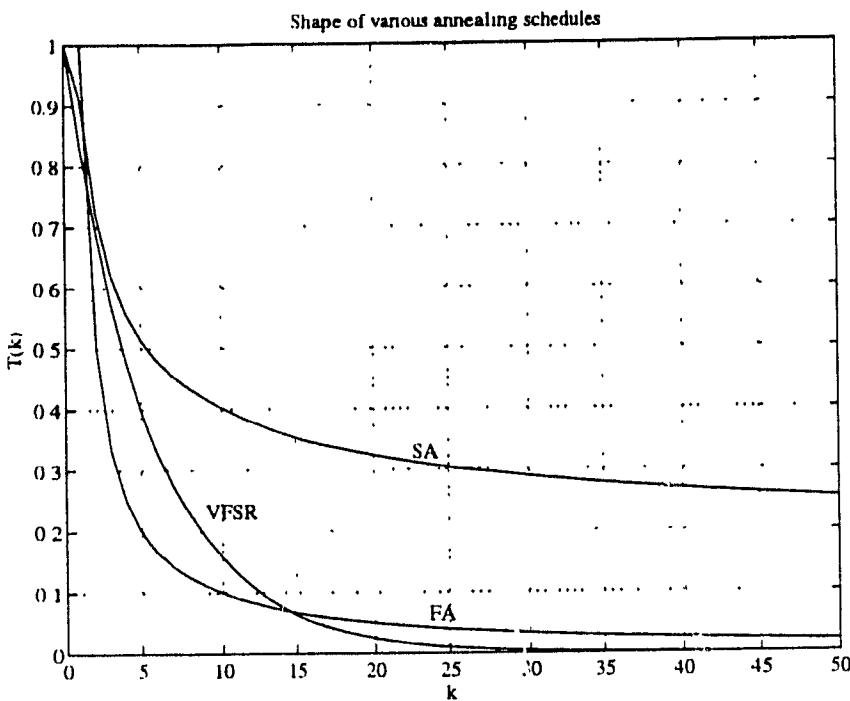


Figure 4.14: This figure shows the shape of the annealing schedule for Simulated Annealing (SA), Fast Annealing (FA) and Very Fast Simulated Reannealing (VFSR). It can be seen that the temperature is reduced at a much faster rate in the VFSR case than in the SA or FA case.

of iterations $k_f = 50$. Thus, only 50 iterations are necessary in VFSR to reduce the temperature to $T = 0.0001$. It is clear that with the parameters m_i and n_i , one has a much greater control over the cooling process in VFSR than in FA or SA.

Different parameters often have different sensitivities within the search space. For example, with range image registration, our cost function might be more sensitive to variations in parameter r_x than parameter r_z . That is, a change in r_x could produce a much larger change in the value of the cost function than a variation in r_z . When searching a relatively flat and insensitive dimension, the generating distribution should be wider than the distribution used when searching a dimension more sensitive to change. The reannealing aspect of VFSR consists of periodically rescaling the annealing times k_i for each individual state parameter α_i (which has the effect of rescaling the annealing temperatures) in order to accommodate the varying sensitivities of the parameters during the search.

By rescaling the temperature or individual parameters, the reannealing effectively expands or contracts the generating distribution of individual parameters. It has proven useful

to periodically rescale the annealing-time, k , every hundred or so acceptance-events [13]. This rescaling is performed by measuring the gradient of the cost function at the point of minimum cost found at that time to determine the sensitivities of each parameter. The sensitivity s_i of parameter α_i is obtained from the energy/cost function E by

$$s_i = \frac{\partial E}{\partial \alpha_i} \quad (4.22)$$

Then, using the largest parameter sensitivity $s_{max} = \max_i \{s_i\}$, each parameter annealing time k_i is rescaled as follows:

$$\begin{aligned} k_i &\rightarrow k'_i \\ T_{i,k'} &= T_{i,k}(s_{max}/s_i) \\ k'_i &= (\ln(T_{i,0}/T_{i,k'})/c_i)^D \end{aligned} \quad (4.23)$$

At the beginning of the minimization search, all temperatures $T_{i,0}$ are initialized to unity. This is more than adequate to span each parameter dimension [13]. Note that each parameter α_i has its own time value k_i and temperature $T_{i,k}$ at that time. Similarly, the acceptance function has its own temperature and time. Remember that the acceptance function used for VFSR is still the same as used for BA and FA (Equation (4.12)). When reannealing is performed, the acceptance temperature is also rescaled. When starting the minimization, the initial value of the acceptance temperature is set by randomly sampling the energy/cost function at a few points before the search and setting the temperature to the average value of these samplings. When reannealing, this temperature is reset to the most current minimum of the cost function E , and the corresponding annealing-time is set so that the new temperature equals the lowest value of the cost-function encountered to date. Let E_k be the current value of the energy/cost function and let E_{best} be the lowest energy value encountered up to that point in the minimization search. Then, when reannealing, the temperature used for the acceptance function $h()$, T_h and the corresponding annealing time k_h are set as follows:

$$\begin{aligned} k_h &\rightarrow k'_h \\ T_{h,0} &= E_k, T_{h,k'} = E_{best} \\ k'_h &= (\ln(T_{h,0}/T_{h,k'})/c_i)^D \end{aligned} \quad (4.24)$$

4.5 Summary

In this chapter, two different optimization methods were presented: Genetic Algorithms and Very Fast Simulated Reannealing.

Both are effective methods for finding the optimum point of non-convex cost functions. For our range image registration problem, we need to find a rigid 3D transformation which minimizes the cost function defined in Section 3.5.1. Many local minima are present in this cost function and these optimization techniques were suitable for such a problem.

The first section of the chapter introduced the various concepts involved with Genetic Algorithms. An interesting aspect of Genetic Algorithms is their ease of implementation. Reproduction, crossover and mutation operations are all fairly simple procedures to code on a computer. A legitimate question about Genetic Algorithms is whether there is any guarantee that the algorithm will converge to the global minimum of the cost function. Unfortunately, no formal proof of convergence has been derived for genetic algorithms. The support for this optimization technique comes from its experimental success. Many researchers have used it on various types of problems and have obtained good results with it [6], [9], [20], [21]. In the next chapter, the effectiveness of Genetic Algorithms will be tested on performing the registration of real range data.

Like Genetic Algorithms, Simulated Annealing is also a stochastic optimization technique. Contrary to Genetic Algorithms, a proof that Simulated Annealing statistically converges to the global optimum under specific conditions exists. This is an interesting property of Simulated Annealing. Although a simple Simulated Annealing algorithm has an extremely slow rate of convergence, improvements made on the original algorithm have allowed the convergence rate to increase considerably without affecting the proof of convergence. Very Fast Simulated Reannealing has been developed by Lester Ingber in 1989 [12] and is the latest improvement made on the original Simulated Annealing algorithm. Ingber has also performed comparison between Genetic Algorithms and Very Fast Simulated Reannealing [13] and found that in many cases the convergence of Very Fast Simulated Reannealing was orders of magnitude faster than Genetic Algorithms. This makes this optimization technique rather attractive for the purpose of our minimization problem.

These optimization techniques will be tested and compared in the next chapter.

This chapter presents the experiments conducted in conjunction with this research. A description of the experimental setup and the data acquisition process is given in Section 5.1.

In Section 5.2, the measurement error introduced by the rangefinder camera during data acquisition is examined. Section 5.3 gives a description of the preprocessing performed on range images prior to their registration and Section 5.4 explains the method used to determine an estimate of the transformation between range views. To obtain an idea of the computing time necessary to perform a registration search, the time required to evaluate the cost function is determined in Section 5.5.

As was presented in Chapter 4, two optimization methods have been used for the registration: Genetic Algorithms and Very Fast Simulated Reannealing. These optimization techniques are subject to various control parameters which affect their performance. Before any registration experiments are conducted, it is important to establish the optimum setting of these parameters. Section 5.6 discusses the search parameters of each optimization method and determines values which yield optimum performance of the registration search.

Section 5.7 presents various dual-view registration experiments. Each experiment consists of the registration of two range views obtained by sampling an object from two different viewing positions. Finally, in Section 5.7.6, a multi-view registration experiment is presented. The notion of local/global optimization arising when registering multiple views of an object is examined and a solution is presented. A summary of the chapter is given in Section 5.8.

5.1 Experimental Setup and Data Acquisition

An eye-in-hand system is used for the acquisition of range images. The system consists of a rangefinder camera attached to the end effector of a PUMA 560 robot arm. The robot is attached to the ceiling to permit easier positioning of the camera for viewing objects at various angles. Figure 5.1 shows the camera/robot setup used in the experiments.

An alternate method of data acquisition is a turntable. While maintaining the rangefinder camera in a fixed position, a precision turntable is used to accurately rotate

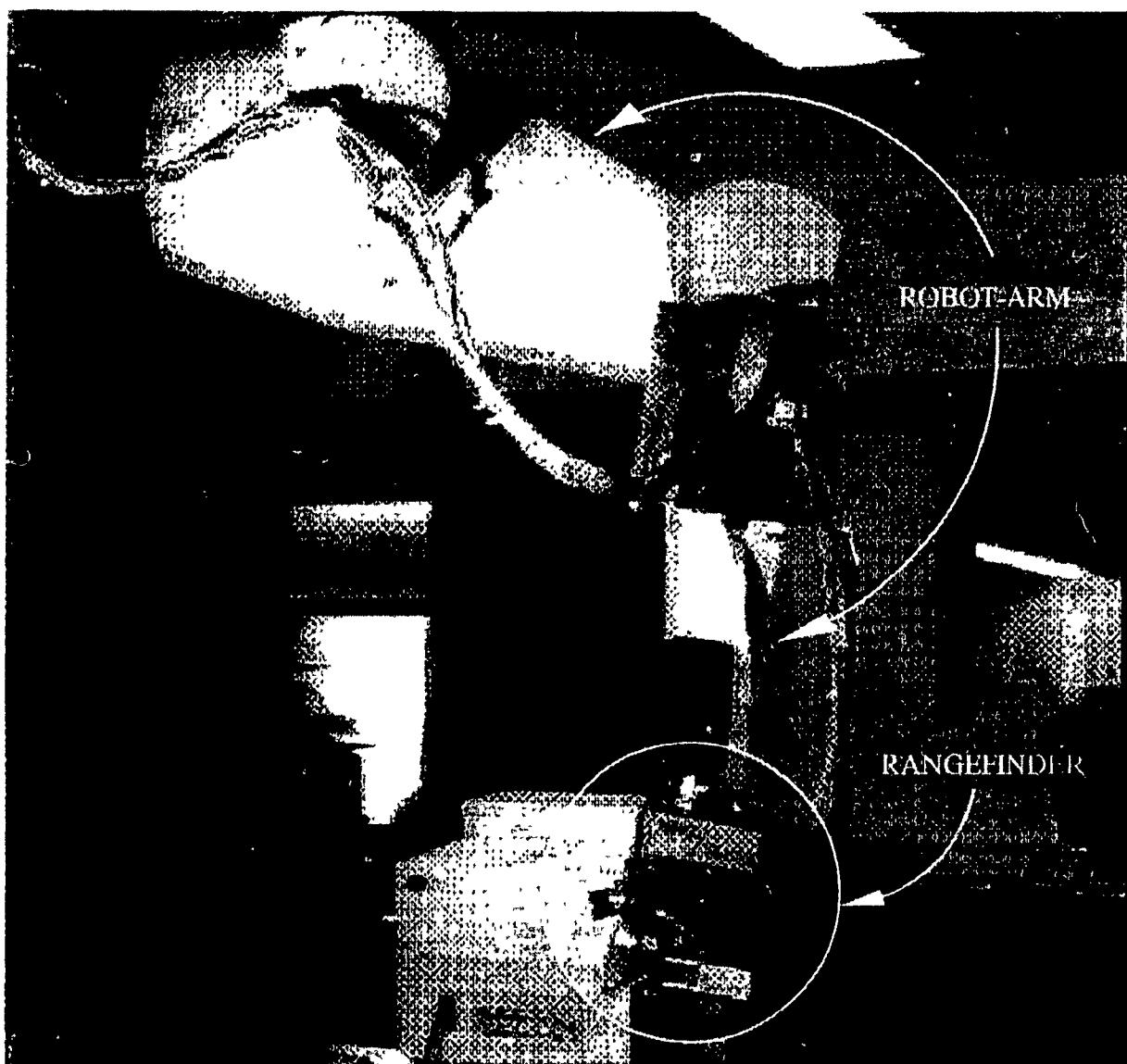


Figure 5.1: The eye-in-hand system used in the range image registration experiments. A PUMA 560 robot arm attached to the laboratory ceiling is used to move the rangefinder camera around objects for data acquisition. The NRC/McGill laser rangefinder described in Section 3.4 is attached to the end-effector of the robot arm.

objects so that sampling of objects from different viewpoints can be achieved. The position of the turntable can be specified as an absolute angular value in degrees. The turntable setup is shown in Figure 5.2.

The 3D data describing the surfaces of objects we wish to model are acquired using a laser rangefinder camera. This camera was briefly described in Section 3.4. The size of the range images taken was set to 256 by 256 samples. This is the maximum dimension possible with the rangefinder camera used in this research. The size of the images can be made smaller, but it was decided to make use of the full precision allowed by the camera. The registration program developed can nevertheless handle range images of any size. This can be useful when exploring small regions of object which could have been occluded during a first sampling. In such cases, it might be advantageous to use smaller images to speed up the registration process.

When using the robot arm system for data acquisition, the robot is controlled manually in order to obtain an arbitrary viewing position for the camera. With the turntable, the acquisition process is automated. A control program was written to alternatively sample the object and then rotate it to a new position. The desired angular displacement between consecutive views is specified and the acquisition process is carried out automatically. Figure 5.3 shows a typical example of multiple view range data acquired using the turntable. Nine range views of an owl figurine obtained at 40 degree interval are shown.

The registration algorithm is executed on a Silicon Graphics Iris 4D machine (it has also been tested on a Sun Sparc architecture). The results of the registrations are displayed on the Silicon Graphics workstation with a range image display tool called RIM (Range Image Monitor, developed at McRCIM by Peter Whaite and Gilbert Soucy). This tool allows one to display range images obtained with the NRC/McGill rangefinder as three-dimensional grids of points or three-dimensional shaded surfaces. It also allows one to display multiple range images with their respective registration parameters; this is necessary to view the result of a registration.

5.2 Sampling Error of the Laser Rangefinder

When using the laser rangefinder to sample a point on a surface, a small error is introduced in the distance measured compared with the actual distance of the sampled point. To quantify the quality of a registration fit, a sum of Euclidean distances between corresponding points

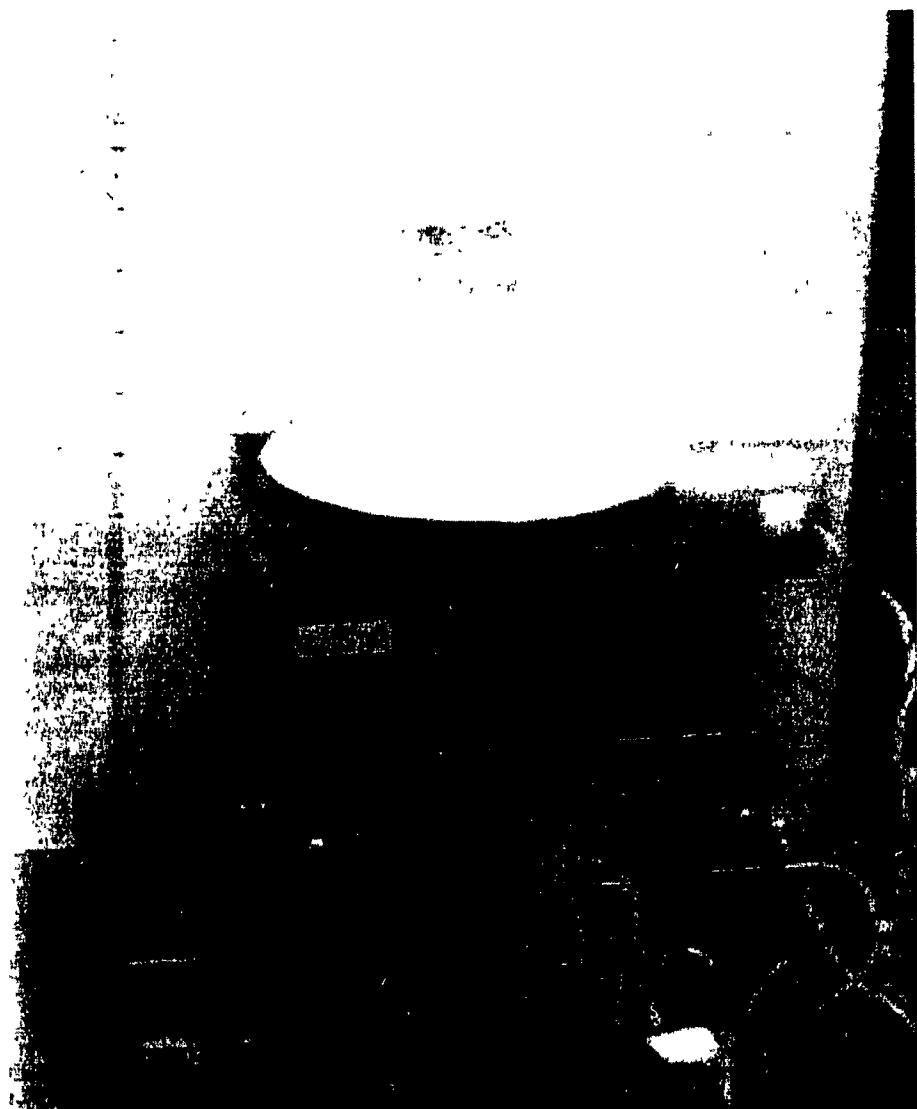


Figure 5.2: Precision turntable used for the acquisition of multiple range views of objects. The figure shows a typical object (owl figurine) positioned on the table. A rangefinder acquires surface data of the object from a fixed position. The object is then reoriented by the turntable so that a different view can be sampled.

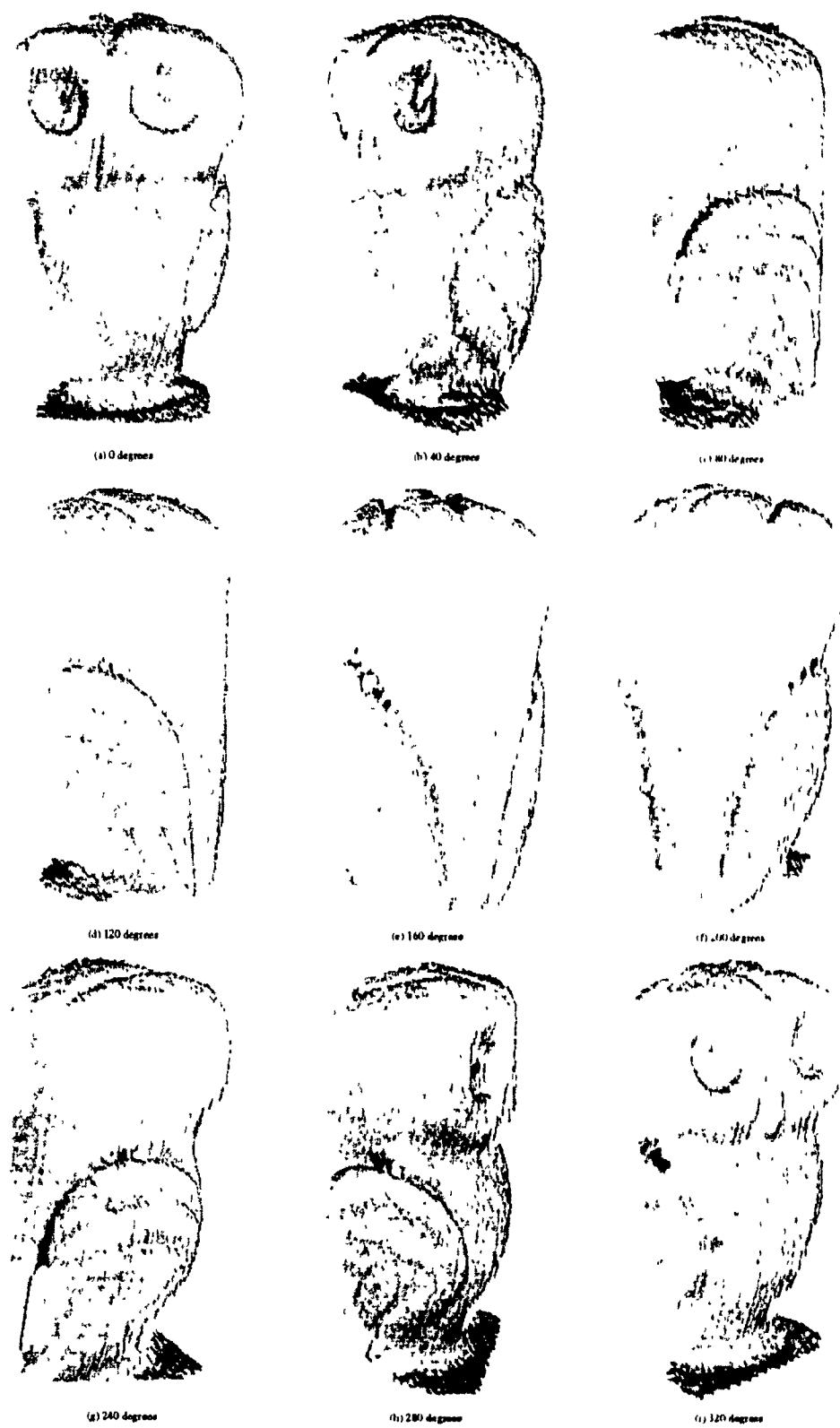


Figure 5.3: This figure shows nine range images of a small owl figurine seen from nine different viewpoints. The views were obtained by rotating the object on a precision turntable by 40 degrees intervals. The views shown above were preprocessed in order to remove spurious data points and background surfaces such as the platform of the turntable on which the object is resting, so that only the owl's surface remains.

across views is computed. This sum of distances defines the cost function used in the registration search (Equation (3.37)). In the ideal case of two perfectly registered range views, the distance between overlapping regions of the views is zero. In practice however, this distance is non-zero. This is due to the discretization of the surfaces (continuous surfaces are represented as grids of sampled points), the digitization of the distances measured (digitized by the measuring instrument to a fixed number of bits) and to the measurement error of the instrument.

Because the cost function is based on the Euclidean distance between corresponding points in a pair of range views, the measurement error of the rangefinder will affect its value. The effect of digitization on the cost function value is negligible. Ten bits are used to encode the distance measured by the rangefinder and this is quite sufficient for the measuring accuracy of the rangefinder (if more bits were used, their value would be meaningless). We assume that the density of sampled points on a surface is high enough so that when a point correspondence is made, the *discrete* correspondence point found through the inverse calibration equations is very close (in Euclidean sense) to the *true* correspondence point. We thus assume that the effect of discretization on the cost function value is negligible. Therefore, the value of the cost function is affected mainly by the sampling error of the rangefinder. If two range views were to be perfectly registered, the cost function, which is simply an average distance between corresponding points, should reflect the average sampling error. Thus, by knowing the sampling error of the rangefinder camera, we are able to better ascertain, based on the cost function value, the quality of a registration.

To determine the value of the average sampling error, a planar surface was sampled at different distances. By doing this, we can determine a relation between the sampling error and the sampling distance. Range images of the plane are obtained, each image containing 256 by 256 (65536) sampled points. A planar model is fit to the data using a least squares fitting method. The distance of each data point to the plane is then computed and analysed statistically.

Figure 5.4 shows probability density curves computed from data obtained by sampling a plane at different distances. Each curve represents a different distance. We can see that the error is Gaussian distributed and that as the sampling distance increases, the measurement error of the rangefinder increases. This can be seen in the widening of the Gaussian distribution curves (more points have a large sampling error) as the sampling distance is

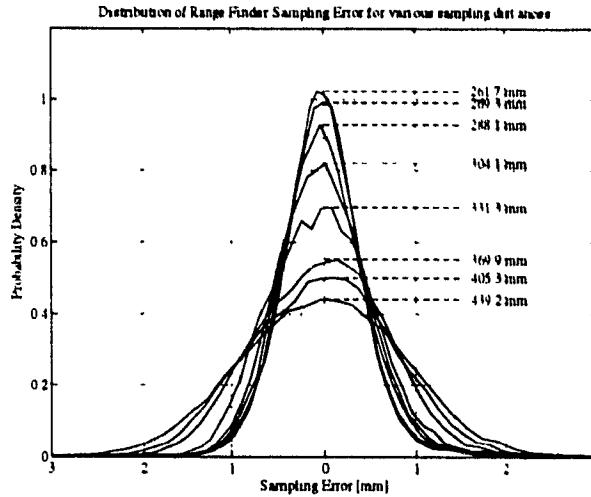


Figure 5.4: This graph shows how the sampling error is distributed for different sampling distances. The probability density of the error is plotted for various sampling distances. We see that the error is Gaussian distributed and that the error increases (the distribution becomes wider) as the distance to the plane being sampled is increased.

increased. Figure 5.5 shows this more clearly. It is a plot of the mean absolute error (the average unsigned distance from the fit plane) as a function of the scanning distance. We can see that a linear relationship exists between the error and the scanning distance. In most experiments conducted for this research, the data acquisition was performed with the rangefinder at a distance of around 40 centimeters away from the object scanned, sometimes more depending on the size and shape of the object. At this distance we see that the average error in the measured distance of a sampled point is approximately 0.625 millimeters.

With this in mind, a range surface can be seen as a perfect 3D representation of the surface of the object being scanned plus some added noise. This noise is Gaussian distributed with a standard deviation proportional to the scanning distance. When registering two range views, we therefore expect the average Euclidean distance between corresponding points in each view to be twice the mean absolute sampling error (the errors in each view get added). Since the cost function computes an approximation of this average distance, when two views are properly registered we expect the cost function value to be around 1.25 millimeters, which is twice the mean absolute error for an object scanned at a distance of 40 centimeters.

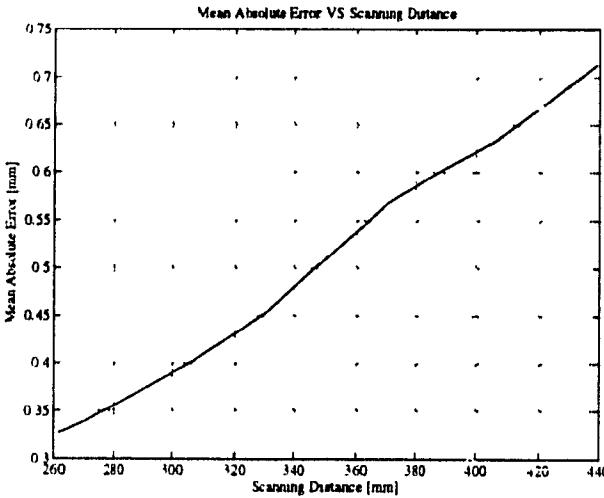


Figure 5.5: This graph shows a plot of the mean absolute error as a function of the sampling distance. We see clearly that as the distance between the rangefinder camera and the plane being scanned increases, the measurement error increases

5.3 Preprocessing of Range Images

Before registering the range images, some preprocessing is necessary. When the views are acquired with the turntable, the object is rotated with respect to the camera, but the background is fixed. As shown in Figure 5.6(a), a vertical plane positioned behind the turntable as well as the platform of the turntable are visible when acquiring these range views. This background information is removed from the images so that it does not interfere with the registration. A simple segmentation program has been designed to specifically remove the background information. Points are manually extracted from both the rear plane and the turntable platform and planes are fit. Using these two plane equations, the program determines which points in the range image are behind the rear plane or below the platform. All points that are within a certain threshold distance in front of the back plane are eliminated as well as all the points behind it. Similarly, all points above a specified threshold distance from the platform plane and all points below it are eliminated. Figure 5.6(b) shows the resulting range image once the background has been eliminated from the raw image. The “cutting” thresholds were also set manually.

The removal of spurious data points is another preprocessing step performed on the range images. These points sometimes result from shadows, highly reflective surface regions on the object being scanned or other sources of noise present during the data acquisition. The

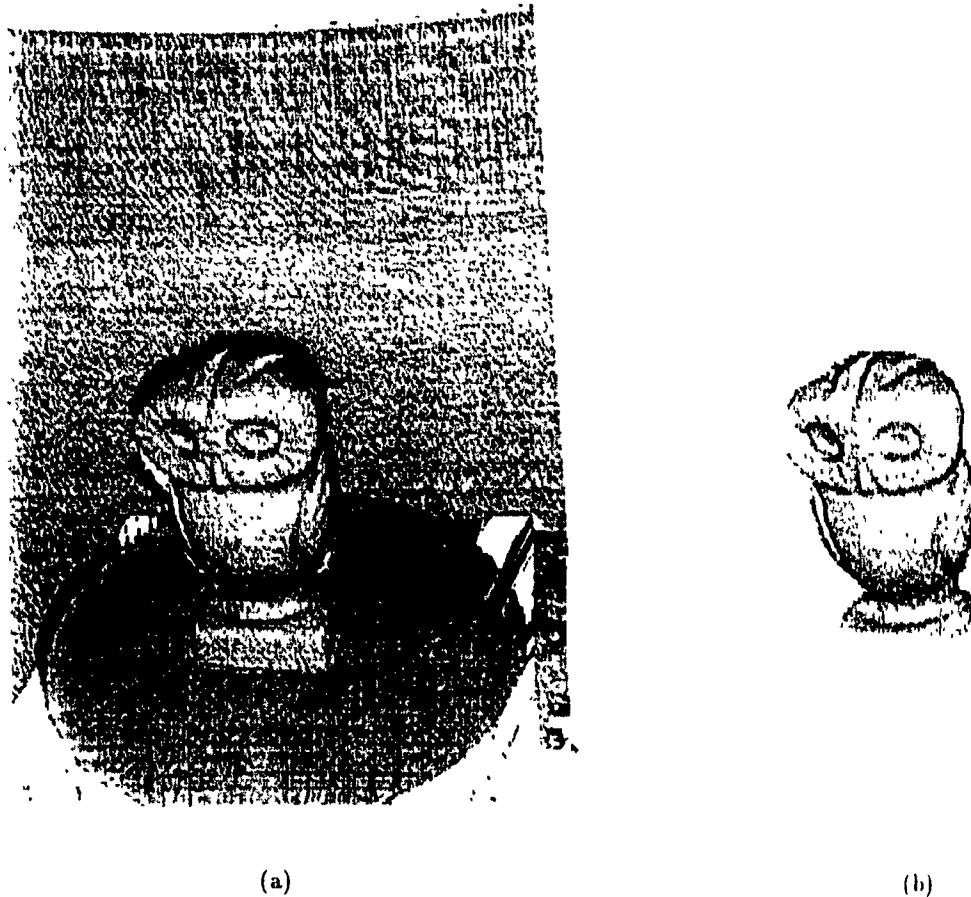


Figure 5.6: Figure (a) shows a typical raw range image obtained with the precision turntable. A vertical background plane is present and the platform of the turntable is also part of the raw data. Figure (b) shows the range view obtained once these background points have been eliminated from the raw image

range images consist of two-dimensional arrays of 3D points. Therefore each point (except for points on the image boundary) has 8 immediate neighboring points. Similar to median filtering, a point is considered spurious if the Euclidean distance between it and one of its neighbors is greater than a specified threshold. The points on the image boundary are simply eliminated.

5.4 Determination of the Initial Transformation Estimate

One of the assumptions made when registering two range views is that an initial estimate of the transformation between the views is available as a starting point for the optimization search.

When using the precision turntable to obtain different viewpoints of an object, the exact rotation of the platform on which the object is resting is known. To determine the transformation estimate between two views, the position of the axis of rotation of the turntable with respect to the rangefinder must also be known. To compute the position of the axis, a special calibration tool has been designed. The tool consists of a metal cylinder machined to have an edge (two faces meeting at 90 degrees) in the location of the central axis. The tool is inserted in a hole at the center of the turntable platform and scanned by the rangefinder from the position it will occupy during all the following range view acquisitions. In this position, the edge located on the cylinder central axis corresponds exactly with the axis of rotation of the turntable. Figure 5.7 shows the calibration tool inserted in the center of the turntable.

By scanning the calibration tool with the rangefinder, the axis of rotation of the turntable can be obtained by extracting points on the edge of the tool from its range image. Two points $\vec{p_1}$ and $\vec{p_2}$ are extracted from the edge of the calibration tool. These two points define the axis of rotation of the turntable. A more accurate extraction of the axis would be obtain with a least-square fit of many points on the axis, but the use of two point was considered sufficiently accurate for our needs. Given these two points and a rotation angle θ , the initial transformation estimate T_e is computed as follows.

First, the vector defining the axis of rotation is computed from the two extracted points and it is normalized:



Figure 5.7: This figure shows the calibration tool used to determine the axis of rotation of the precision turntable. When the tool is inserted in the center hole of the turntable, its edge falls exactly in line with the axis of rotation of the turntable. By scanning the tool with the rangefinder camera, one is able to extract points on this edge. From these points, the relative position of the camera with respect to the axis of rotation can be determined. Given the angle of the rotation between two range views, the initial transformation estimate between the views is computed.

$$\vec{a} = \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} = \frac{\vec{p}_2 - \vec{p}_1}{\|\vec{p}_2 - \vec{p}_1\|} \quad (5.1)$$

From this we then find the rotation matrix Q corresponding to a rotation of θ around the axis defined by vector \vec{a} . Note that the angle θ is specified in radians:

$$Q = \begin{pmatrix} a_x^2\beta + \cos\theta & a_x a_y \beta - a_z \sin\theta & a_x a_z \beta + a_y \sin\theta \\ a_x a_y \beta + a_z \sin\theta & a_y^2\beta + \cos\theta & a_y a_z \beta - a_x \sin\theta \\ a_x a_z \beta - a_y \sin\theta & a_y a_z \beta + a_x \sin\theta & a_z^2\beta + \cos\theta \end{pmatrix} \quad (5.2)$$

where

$$\beta = 1 - \cos\theta$$

From Equation (5.2), obtaining the Euler angles is simply a matter of equating Q with the matrix R defined in Equation (3.3) and solve for the three Euler angles r_x , r_y and r_z .

The translation parameters t_x , t_y and t_z are obtained as follows:

$$\begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} = \vec{p}_1 - Q\vec{p}_1 \quad (5.3)$$

When acquiring range views using the robot arm, the absolute position of the camera during data acquisition is automatically returned by the robot controller. From this, an estimate of the transformation between two views can be derived.

The robot arm is calibrated so that the absolute position and orientation in space of its end-effector can be determined directly from the spatial configuration of all its links. Another calibration is done to establish the relative transformation between the robot's end-effector on which the rangefinder camera is attached and the optical center of the rangefinder camera. This information is sufficient for the robot controller to determine the absolute position of the camera (see Soucy [27]).

For each view acquired, the controller returns a translation vector and a rotation matrix specifying the position and orientation of the camera in space with respect to a unique

world coordinate system. Computing the relative rotation and translation between the two camera position is then straightforward. Given two vectors t_{1w} and t_{2w} representing the two camera positions expressed in the world coordinate system, and given two rotation matrices Q_{1w} and Q_{2w} representing the orientation of the camera with respect to the world coordinate system, the relative displacement and rotation between the two camera positions are obtained as follows:

$$\begin{aligned} Q_{12} &= Q_{1w}Q_{w2} = Q_{1w}Q_{2w}^{-1} \\ t_{12} &= Q_{1w}(t_{2w} - t_{1w}) \end{aligned} \quad (5.4)$$

where Q_{12} represents the rotation necessary to go from the reference frame of the camera in view 1 to the reference frame of the camera in view 2 and where t_{12} represents the translation necessary to displace the origin of frame 1 to the origin of frame 2. Again, to obtain the Euler angles, one equates Q_{12} to the rotation matrix R of Equation (3.3) and solves for the three angles r_x , r_y and r_z .

5.5 Time required for a Cost Function Evaluation

In order to get an appreciation for the time required to perform a registration search, the computing time necessary for performing a single evaluation of the cost function is determined. A typical set of range views is used for this purpose and the cost function is evaluated for the transformation estimate between these two views. Two views of the owl figurine (see Section 5.7.1) are used. The control points were obtained by sub-sampling one of the range views at an interval of 5 pixels to obtain 252 control points (i.e. in the 256×256 range image sampled with a grid 5 pixels in width, 252 points were found on the surface of the object). It is clear that if a smaller sub-sampling interval is chosen, more control points will be selected on the range surface and the time required for an evaluation of the cost function will increase accordingly. A sub-sampling of 5 is chosen because it yields good registration results in most cases.

The cost function evaluation time was determined for two different computer architectures. A Sun Sparc 1+ computer and a Silicon Graphics Iris 4D computer were used for the experiments. The graph in Figure 5.8 shows the computing time required to perform a varying number of cost function evaluations. Two curves are shown, one for each of the computer systems used. The time required for a single evaluation of the cost function is

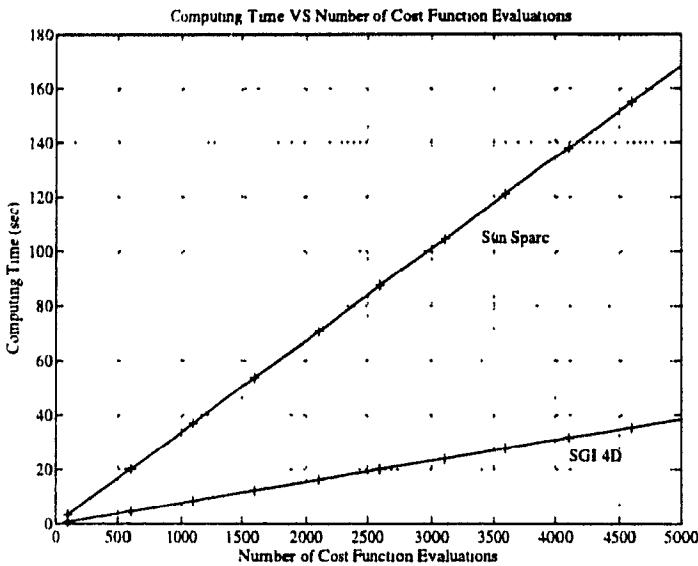


Figure 5.8: This graph shows the computing time required to perform various numbers of cost function evaluations. The time is plotted for two different computer architectures: a Sun Sparc and a Silicon Graphics Iris 4D. The SGI architecture is clearly faster in performing the cost evaluation than the Sparc

obtained from the slope of the lines. From this we get

Sun Sparc	33.65 ms / evaluation
Silicon Graphics Iris 4D	7.72 ms / evaluation

We see that the SGI 4D architecture is more than four times faster than the Sun Sparc architecture when performing this task. We will therefore use the SGI for all experiments.

5.6 Search Control Parameters

The parameters controlling the registration process can be divided into three categories. The first set of parameters are those controlling the size of the search space and the general shape of the cost function. The other two sets of parameters are those controlling the progress of the Genetic Algorithm and of the Very Fast Simulated Annealing minimization. These are briefly discussed in the following sections.

5.6.1 Search Space and Cost Function Parameters

When performing the registration of two range views, an estimate of the 3D transformation between the views is available as a starting point of the search. To limit the search, an

upper and lower bound is set for each of the motion parameters t_x , t_y , t_z , r_x , r_y and r_z . These boundaries are set by specifying a search range around the estimate of each motion parameter.

Let Δ_{tx} , Δ_{ty} , Δ_{tz} , Δ_{rx} , Δ_{ry} , Δ_{rz} be the search range for the motion parameters t_x , t_y , t_z , r_x , r_y and r_z , respectively. Using the transformation estimate, the lower bound for each parameter is obtained by subtracting the Δ range from the estimated parameter and the upper bound is obtained by adding the range to the estimated parameter (See Equation (4.4) in Chapter 4).

The size of the search range specified should depend on how accurate the transformation estimate is. This in turn is defined by the accuracy of the range data acquisition process. For example, if the precision turntable is used, then a small search range can be specified since the transformation estimate is likely to be very close to the optimum transformation. On the other hand, if the range data acquisition is performed with the eye-in-hand system, the transformation estimate between acquired range views is likely to have a higher degree of inaccuracy. Therefore, to guarantee that the optimum solution is within the search space, a larger search range must be specified. Appropriate settings of the search range were determined by experimentation with various range views obtained with the turntable and the robot arm described in Section 5.1. Search ranges of ± 2 millimeters and ± 1 degree seem to be adequate when the turntable is used, and search ranges of ± 10 millimeters and ± 5 degrees when the robot arm is used.

The other parameters affecting the search performance are related to the evaluation of the cost function. These parameters are:

- (i) Control point sample interval (see Section 3.5.2)
- (ii) Overlap factor (see Section 3.5.1)
- (iii) Distance Threshold (see Section 3.5.1)

A range image consists of a two-dimensional array of sampled 3D points. A full-size image will contain 256×256 points. The control point sample interval is an integer value specifying at which interval the control points are selected in the range image. The smaller the interval, the greater the number of control points used for the cost function evaluation. The smallest value possible for the sampling interval is 1. With an interval of 1, all the points contained in the range image will be used for the cost evaluation. With a sampling

interval of 5, for example, one control point would be selected for every 5 points in each of the 2 dimensions of the array of points. Figure 3.13 shows a typical selection of control points in a range image.

The value of this parameter should be based on the complexity of the range images that must be registered. For smooth surfaces the control points can be selected at a greater interval without affecting the proper registration of the two range images. If the surfaces contained in the range images are fairly complex and contain significant high spatial frequency information (edges), then a smaller sample interval must be chosen in order to perform the registration search with the best chance of success. Enough control points must be selected so that the surface they describe is sufficiently well defined by these points alone. One can view the sampling interval as the Nyquist sampling rate for the surface contained in the range image. A sampling rate that is too small (large sampling interval) would not represent the surface properly. This in turn could lead to an improper registration of the range images.

Using all the points contained in the range image is the safest choice for the registration. However, the more control points used, the more computing time is required to evaluate the cost function. The choice for this parameter is thus a compromise between speed of execution and registration performance. For most range images tried, a sampling interval of 5 provided good results.

The second parameter controlling the cost function evaluation is the overlap factor (See Section 3.5.1). It is used to guarantee that the range views overlap by at least a certain specified amount when they are being registered. During the search, all transformations yielding less than the specified overlap are discarded. This parameter was experimentally estimated for each pair of range views. An overlap factor of approximately 30% was used with success for the registration of many range views.

The distance threshold parameter limits the value of the Euclidean distance between a control point and its corresponding point in the other image. When evaluating the cost function, a sum of distances between control points in one view and corresponding points in the other view is computed. All distances are thresholded before being added. Justifications for this parameter are given in Section 3.5.1. The value of this parameter was also determined through experimentation and a threshold value of 5 millimeters was used with success.

5.6.2 Genetic Algorithm Parameters

The parameters controlling the genetic algorithm (see Section 4.3) are:

B_{t_x}	number of bits to encode the translation parameter t_x
B_{t_y}	number of bits to encode the translation parameter t_y
B_{t_z}	number of bits to encode the translation parameter t_z
B_{r_x}	number of bits to encode the rotation parameter r_x
B_{r_y}	number of bits to encode the rotation parameter r_y
B_{r_z}	number of bits to encode the rotation parameter r_z
P	population size
G	number of generations before stopping genetic search
μ_c	crossover probability
μ_m	mutation probability

Appropriate values for these search parameters were obtained through experimentation with the registration of various range views. Typical values producing acceptable results are:

$$B_{t_x} = B_{t_y} = B_{t_z} = B_{r_x} = B_{r_y} = B_{r_z} = 5$$

$$P = 300$$

$$G = 45$$

$$\mu_c = 1.0$$

$$\mu_m = 0.001$$

It was found experimentally that the proper convergence of the genetic search is particularly sensitive to the appropriate setting of the population size and of the mutation probability. To determine the best setting for P and μ_m , the relative performance of the genetic search was compared for different values of these parameters. The next two sections explains how the values of $P = 300$ and $\mu_m = 0.001$ shown above were obtained.

Optimum Population Size

Two sets of range views were used to conduct this experiment: two views of an owl figurine (Figure 5.18) and two views of a metal pipe (Figure 5.24). The optimum 3D transformation was known for each set of two views. The transformation estimates were obtained by varying

the motion parameters of the optimum transformations. To obtain the transformation estimate, 8 millimeters were added to the optimum translation parameters and 4° were added to the optimum rotation parameters. To guarantee that the optimum solution was contained within the search space, the search range was set to ± 10 millimeters for all translations ($\Delta_{tx}, \Delta_{ty}, \Delta_{tz}$) and $\pm 5^\circ$ for all rotations ($\Delta_{rx}, \Delta_{ry}, \Delta_{rz}$). The control point sampling interval, the overlap factor and the distance threshold were set to 5, 0.3 and 5.0, respectively.

To determine the best population size, the genetic search was performed with various population sizes while the other search parameters were kept constant. This approach does not yield the overall optimum for the search parameters, but nevertheless provides us with near optimum values. The population size was set to 25, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275, 300, 400, 500 and 600. For each set of range views, the registration was conducted 10 times per population size and the average convergence was computed (69.3% of the convergence curves were within one standard deviation and 98.8% within two standard deviations from the mean convergence curve). In all runs of the genetic algorithm, the number of generations was set so that the total number of cost function evaluations was approximately the same in all the tests. It was found experimentally that approximately 12500 cost function evaluations was necessary for the genetic search to converge to a stable point. This can be seen in Figure 5.9 (c) and (d) where the convergence curves level off. The number of cost function evaluations during a genetic search is equal to the population size times the number of generations. Therefore, while the population size was varied, the number of generations was adjusted to make the total number of cost function evaluations close to 12500. This is done so that the comparison of the performance of the various population sizes is fair (i.e. no case is allowed to run longer than the others). The number of generations used for each population size is shown in Table 5.1.

The other parameters were set as follows:

$$B_{t_x} = B_{t_y} = B_{t_z} = B_{r_x} = B_{r_y} = B_{r_z} = 5$$

$$\mu_c = 1.0$$

$$\mu_m = 0.001$$

Figure 5.9 shows the results obtained for the experiment with the owl and pipe. The first two graphs show the average convergence curves (each curve is the average of 10 GA

Population Size	Number of Generations	Cost func. Evaluations	Population Size	Number of Generations	Cost func. Evaluations
25	500	12500	225	56	12600
50	250	12500	250	50	12500
75	167	12525	275	45	12375
100	125	12500	300	42	12600
125	100	12500	400	31	12400
150	83	12450	500	25	12500
175	71	12425	600	21	12600
200	62	12400			

Table 5.1: This table shows the number of generations and the number of cost function evaluations used for running the genetic search with various population sizes. It was found that the genetic search converges in approximately 12500 cost function evaluations. Therefore, for each population size, the number of generations is set so that the number of cost function evaluations is near 12500.

searches) for each population size and the other two graphs show plots of the minimum cost function value obtained as a function of the population size. The optimum population size (for minimum time convergence) is approximately 300, requiring about 45 generations.

Optimum Mutation Probability

This experiment was conducted with the same sets of range views used for determining the optimum population size: two views of an owl figurine and two views of a metal pipe. All parameters were the same as before except this time the mutation probability parameter μ_m was varied and the population size was kept constant.

The genetic search was performed for the following mutation probability: 1e-6, 5e-6, 1e-5, 5e-5, 1e-4, 5e-4, 1e-3, 5e-3, 1e-2, 5e-2, 1e-1 and 5e-1 (Note: 5e-6 = 5×10^{-6}). The registration was conducted 10 times in each case, for each set of range views, and the average of the convergence curves was computed (75.1% of the convergence curves were within one standard deviation and 95.7% within two standard deviations from the mean convergence curve). The population size was set to the optimum value found previously, namely 300 and the number of generations was set to 45.

Figure 5.10 (c) shows the results obtained for the experiment with the owl views and Figure 5.10 (d) shows the results obtained with the pipe views. The first graph shows the average convergence curves (of 10 runs each) for each mutation probability and the second graph shows a plot of the minimum cost function value obtained as a function of the mutation probability.

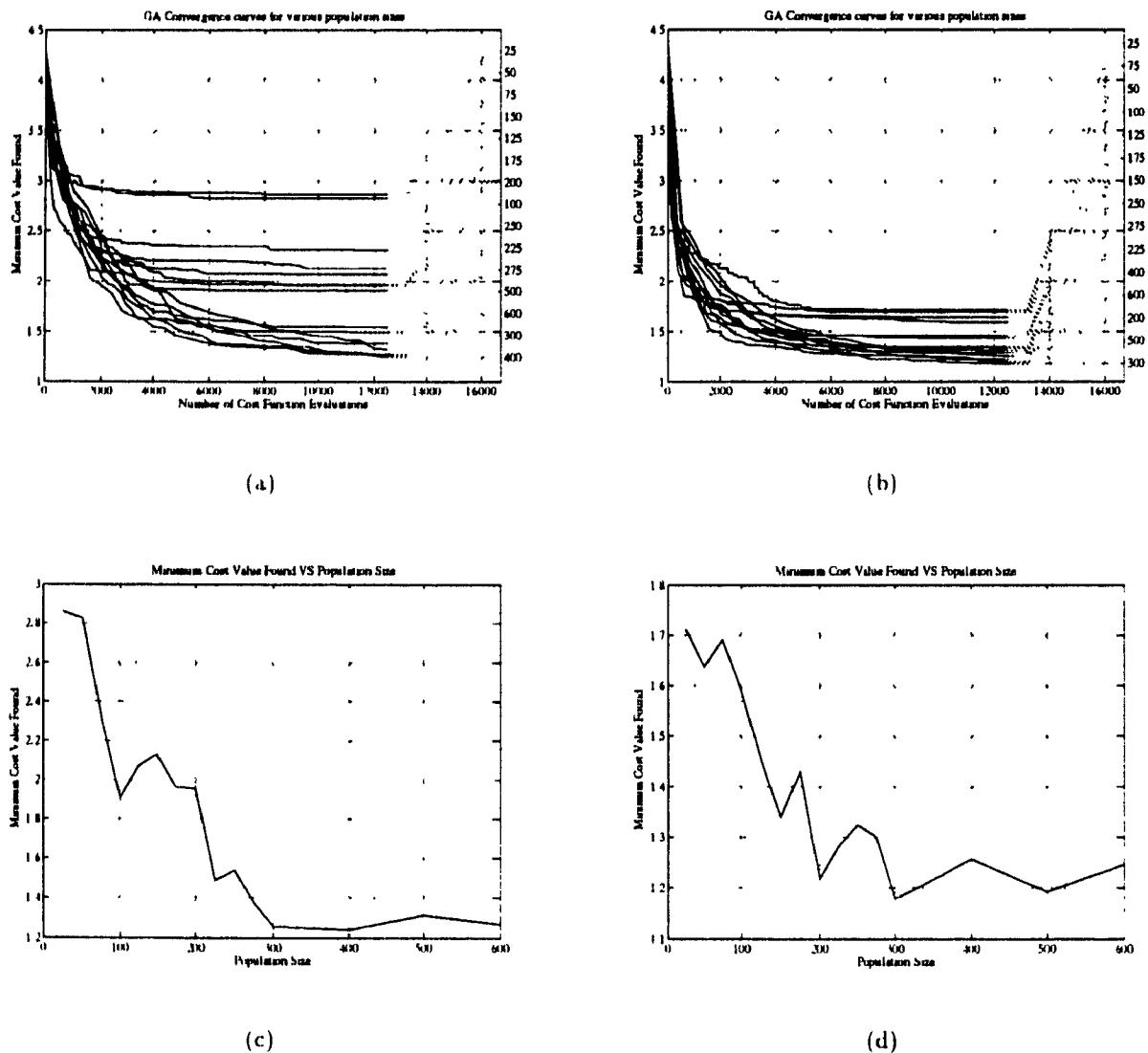


Figure 5.9: These graphs show the convergence of the genetic algorithm search obtained for various population sizes. The results were obtained by performing the registration of two range views of an owl figurine ((a) and (c)) and two range views of a metal pipe ((b) and (d)). Figures (a) and (b) show the convergence curves obtained for each population size. Each curve is the average of 10 GA searches and is linked by a dotted line to its corresponding population size label on the right hand side of the graph. Figures (c) and (d) show the optimum cost value found by the GA search as a function of the population size.

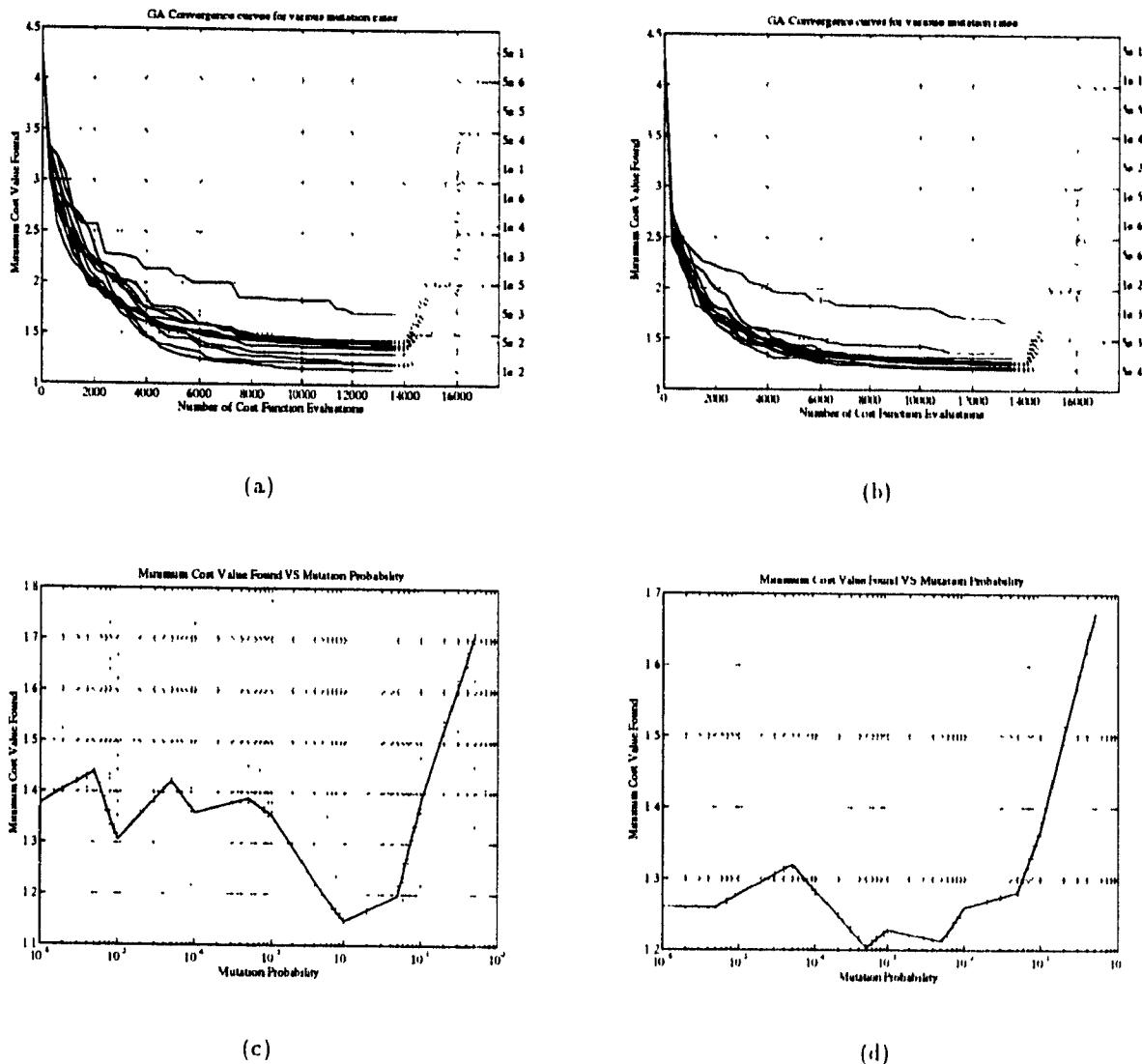


Figure 5.10: These graphs show the convergence of the genetic algorithm search obtained for various mutation probabilities. The results were obtained by performing the registration of two range views of an owl figurine ((a) and (c)) and of two range views of a metal pipe ((b) and (d)). Figures (a) and (b) show the convergence curves obtained for each mutation probability. Each curve is the average of ten GA searches and is linked by a dotted line to its corresponding mutation probability label on the right hand side of the graph. Figures (c) and (d) show the optimum cost value found by the GA search as a function of the mutation probability. We can see that in the case of the owl views (c), the GA search has a slightly better performance with a mutation probability of 0.01. For the pipe views, a mutation probability between 0.0005 and 0.005 yields an optimum solution.

From Figure 5.10 (c) and (d) we observe that a mutation probability between 0.001 and 0.01 is appropriate for a registration search.

It is important to note that the best search parameters are image dependent. Different convergence rates were obtained for the registration of the owl and the pipe range views. Range images of other objects would result in different convergence curves. The optimum search parameters for the owl and the pipe might not be optimum in all cases, but the results obtained with these two objects is a good indication that they will be adequate to register range images of a large number of objects.

5.6.3 Very Fast Simulated Annealing Parameters

The most important parameters controlling the performance of the VFSR algorithm are:

T_0	initial parameter temperature
TRS	temperature ratio scale ($m_i = -\ln(TRS)$)
TAS	temperature annealing scale ($n_i = \ln(TAS)$)

The VFSR minimization program was provided by Professor Lester Ingber¹ and Bruce Rosen and was adapted to perform our registration task. In this program, more parameters are provided to control the search performance of the algorithm than the ones actually mentioned above. However these are the primary parameters for controlling the annealing search process. Note that m_i and n_i are the two free parameters discussed in Section 4.4.4.

Again, appropriate values for these control parameters were determined through experimentation. The same two sets of range images used to determine the population size and the mutation probability (Section (5.6.2)) were used here. Each parameter was varied while the others remained constant to provide us with a near optimum set of values for these parameters. To get a better representation of the search response, the VFSR algorithm was executed 10 times for every parameter setting and the average response was computed (the percentage of convergence curves within one standard deviation was 73.1%, 50.7% and 76.2% for T_0 , TRS and TAS , respectively; and 96.5%, 71.3% and 97.1%, respectively, were within two standard deviations). When T_0 , TRS and TAS were kept constant, they were set to 1.0, 0.001 and 10000, respectively, since it was determined experimentally that these values yield relatively good registration results.

¹Professor Lester Ingber, Lester Ingber Research, P.O. Box 857, McLean, VA 22101.

Optimum Initial Temperature

Figures 5.11 and 5.12 show the results obtained when the initial parameter temperature was varied. The results for the owl and the metal pipe are both shown. From the convergence curves (Figure 5.11) we can see that T_0 has an insignificant effect on the shape of the convergence curves. It does however affect the total search time, because a lower initial temperature results in a shorter annealing time, which causes the optimization to stop early. From figures 5.12 (a) and (b) we can see that an appropriate value for T_0 lies between 0.1 and 1. In this range, the minimum value of the cost function found by the search algorithm is near the optimum point. However, figure 5.12 shows that the higher the initial temperature is, the larger the number of cost function evaluations are required to perform the optimization. The setting of the initial temperature search parameter is thus a compromise between the minimum cost value reached at the end of the search and the number of cost function evaluations necessary to get there. From figure 5.12 we observe that a value of 0.1 produces an acceptable solution in a reasonable number of cost function evaluations, but slightly better solutions are obtained with a value of 1.0. For the registration experiments presented in this thesis, in order to guarantee that the search yields a good registration, the initial temperature was set to a value of 1.0. The time required to perform the registration search will be greater but it will give a better guarantee of obtaining suitable results.

Optimum Temperature Ratio Scale

Figures 5.13 and 5.14 show the results obtained when the temperature ratio scale is varied. The results for the owl and the metal pipe are both shown. From the convergence curves (figure 5.13) we can see that TRS affects both the final optimum solution found and the number of cost function evaluations necessary to reach that optimum. Figures 5.14 (a) and (b) show that for TRS between 0.0001 and 0.001, the minimum cost value reached is near the optimum. As figures 5.14 (c) and (d) indicate, however, a VFSR search with $TRS = 0.001$ reaches the optimum faster (requires a lesser number of cost function evaluations) than with $TRS = 0.0001$. Therefore, $TRS = 0.001$ yields to a better performance of the VFSR search.

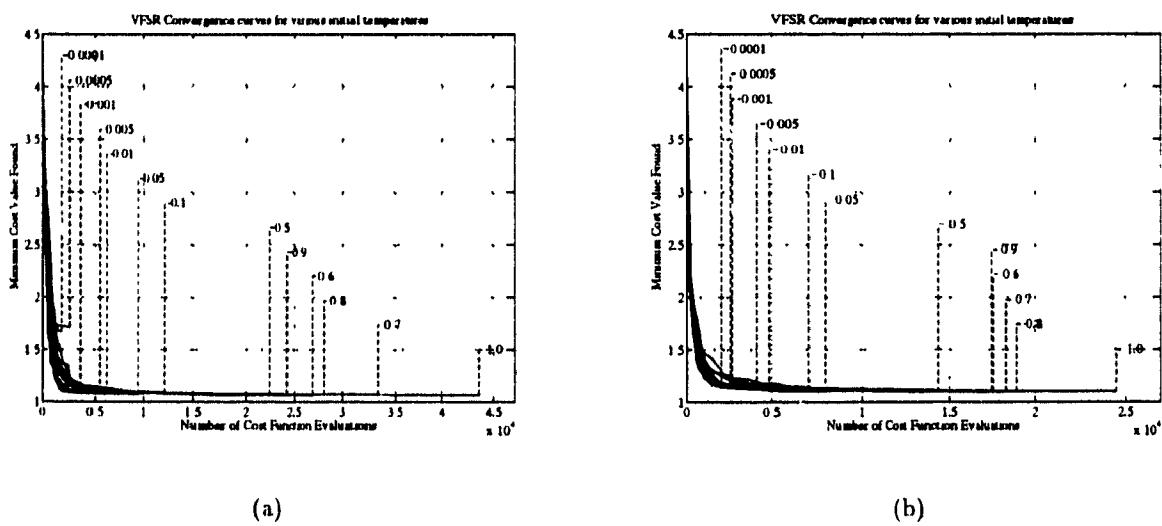


Figure 5.11: These two graphs show convergence curves of the VFSR algorithm obtained by varying the value of the initial temperature search parameter. The registration was performed on two range views of the owl figurine and two range views of a metal pipe. Each curve is the average of 10 VFSR searches. As the graphs indicate, the initial temperature does not affect the shape of the convergence curves significantly but has an impact on the search time. The lower the initial temperature, the shorter the cooling process will be and therefore the earlier the search will be terminated.

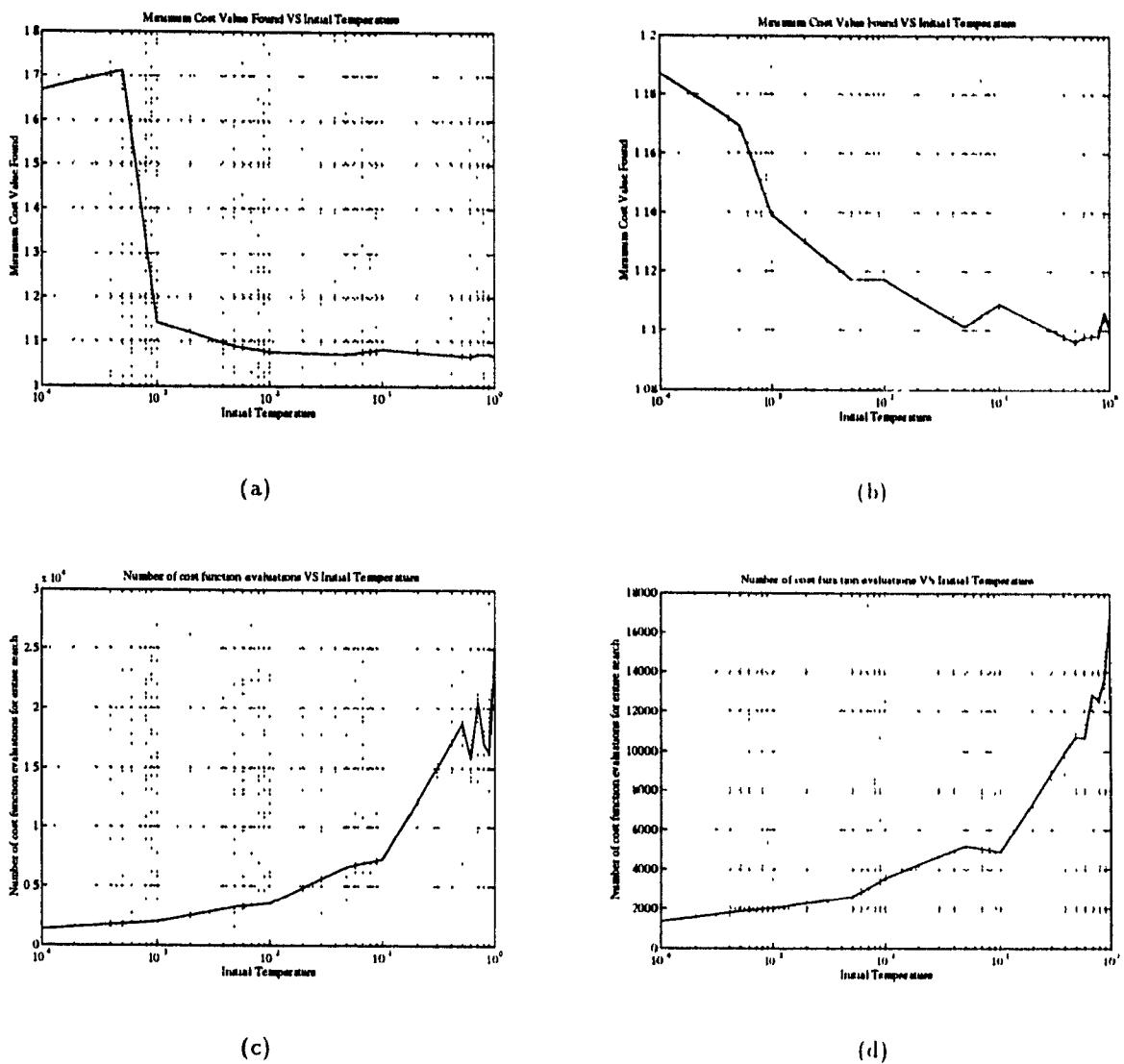


Figure 5.12: These graphs show the results obtained for VFSR in the registration of two sets of range views (owl figurine: (a) and (c), metal pipe: (b) and (d)). These curves are derived directly from the convergence curves of Figure 5.11. Figures (a) and (b) show the final cost value reached at the end of the VFSR optimization search as a function of the initial temperature. Figures (c) and (d) show the number of cost function evaluations performed before the termination of VFSR as a function of the initial temperature. Figures (a) and (b) indicate that for high initial temperatures ((a): $T_0 > 0.01$, (b): $T_0 > 0.1$), the algorithm converges to an acceptable solution (near optimum cost value). From figure (c) and (d) we see that high initial temperatures cause the algorithm to run for a longer time and thus require a larger number of cost function evaluations. An initial temperature of 0.1 is a good compromise between the optimum solution reached and the number of cost function evaluations necessary to reach it.

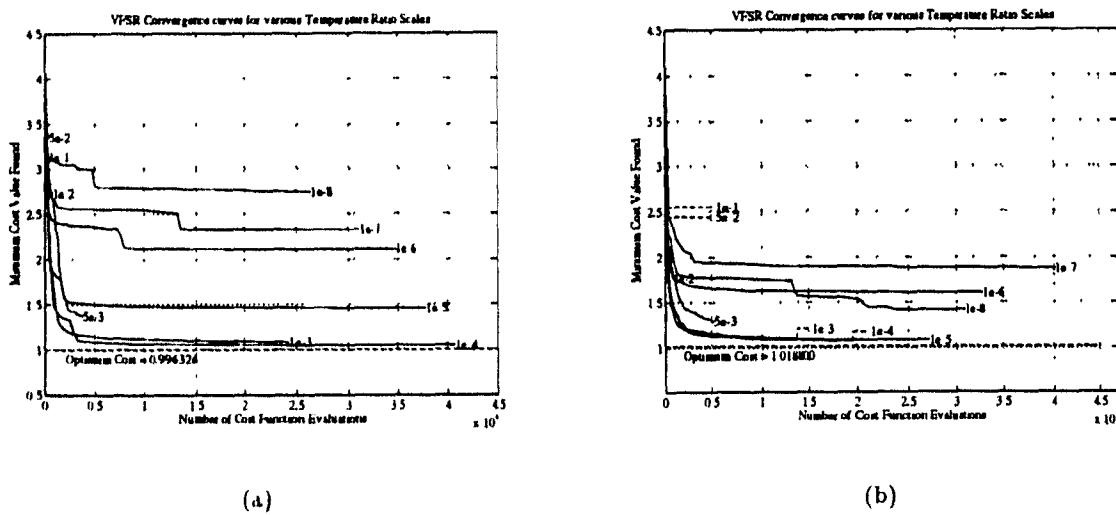


Figure 5.13: These two graphs show convergence curves of the VFSR algorithm obtained by varying the value of the temperature ratio scale parameter (TRS). The registration was performed on two range views of owl figurine and two range views of a metal pipe. Each curve is the average of 10 VFSR searches. As the graphs indicate, the value of TRS affects both the convergence result (final cost value) and the convergence time (number of cost function evaluations).

Optimum Temperature Annealing Scale

Figures 5.15 and 5.16 show the results obtained when the temperature annealing scale is varied. The results for the owl and the metal pipe are both shown. From the convergence curves (figure 5.15) we can see that TAS affects both the final optimum solution found and the number of cost function evaluations necessary to reach that optimum. Figure 5.16 (a) and (b) show that for TAS between 1000 and 10000, the minimum cost value reached is near the optimum. As figure 5.16 (c) and (d) indicate however, a VFSR search with $TAS = 10000$ terminates earlier than with $TAS = 1000$. Therefore, a value of $TAS = 10000$ will require a lesser number of cost function evaluations to carry out the VFSR search.

As was indicated for the GA case, the best search parameters found for VFSR are image dependent. Different convergence rates were obtained for the registration of the owl and the pipe range views and range images of other objects would result in different convergence curves. Even though the values chosen for T_0 , TRS and TAS might not be optimum in all cases, the fact that the results obtained with these the owl and pipe views were good, is a good indication that they will be adequate to register range images of a large number of objects.

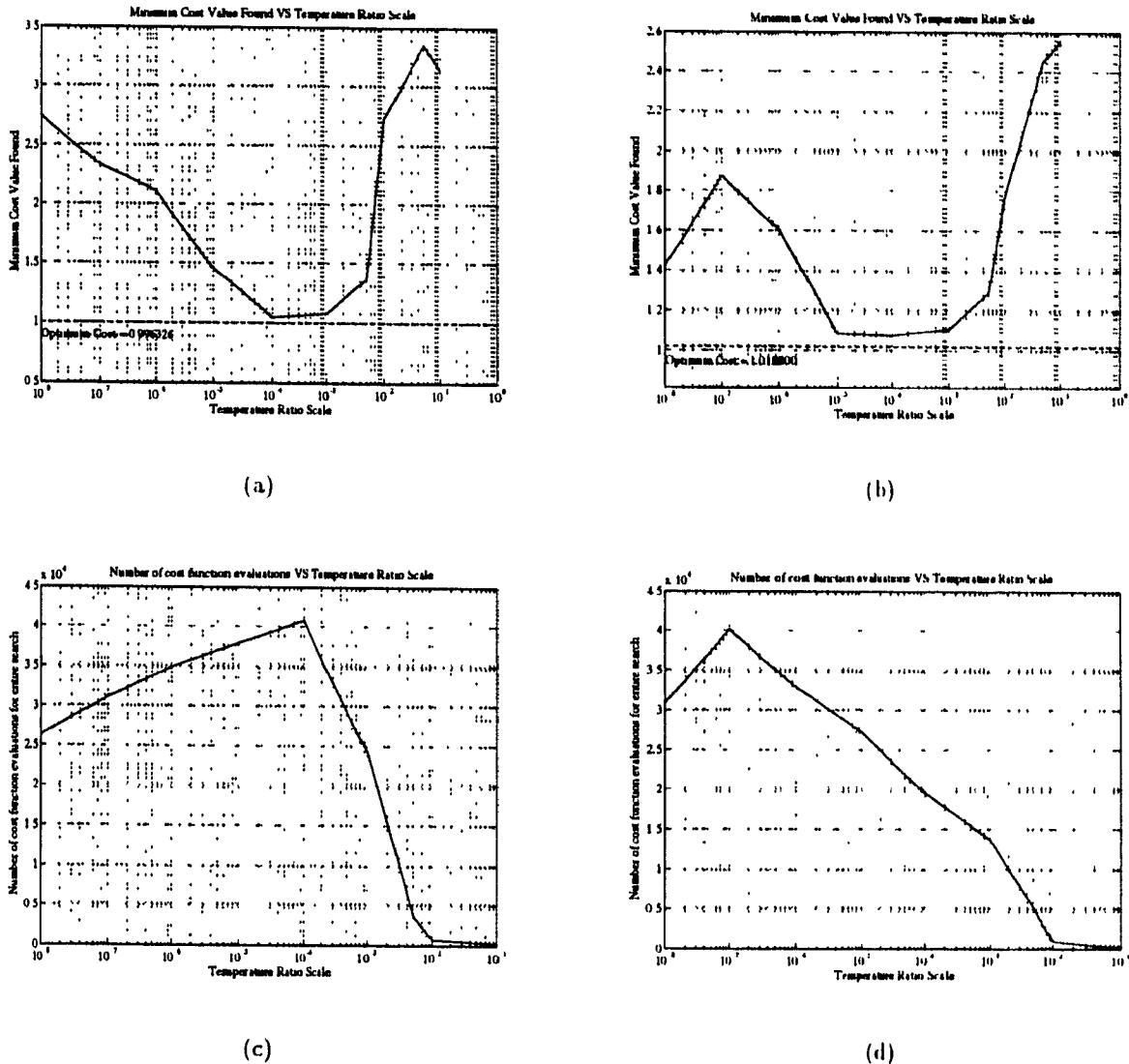


Figure 5.14: These graphs show the results obtained with VFSR in the registration of two sets of range views (owl figurine: (a) and (c), metal pipe (b) and (d)). The curves are derived directly from the convergence curves of figure 5.13. Figures (a) and (b) show the final cost value reached at the end of the VFSR optimization search as a function of the temperature ratio scale (TRS). Figures (c) and (d) show the number of cost function evaluation performed before the termination of VFSR as a function of TRS. Figure (a) and (b) indicate that for TRS between 0.0001 and 0.001, the algorithm converges to an acceptable solution (near optimum cost value). From figure (c) and (d) we see that VFSR terminates sooner (after a smaller number of cost function evaluations) with $TRS = 0.001$ than with $TRS = 0.0001$. Therefore, a value of $TRS = 0.001$ is adequate for the optimization.

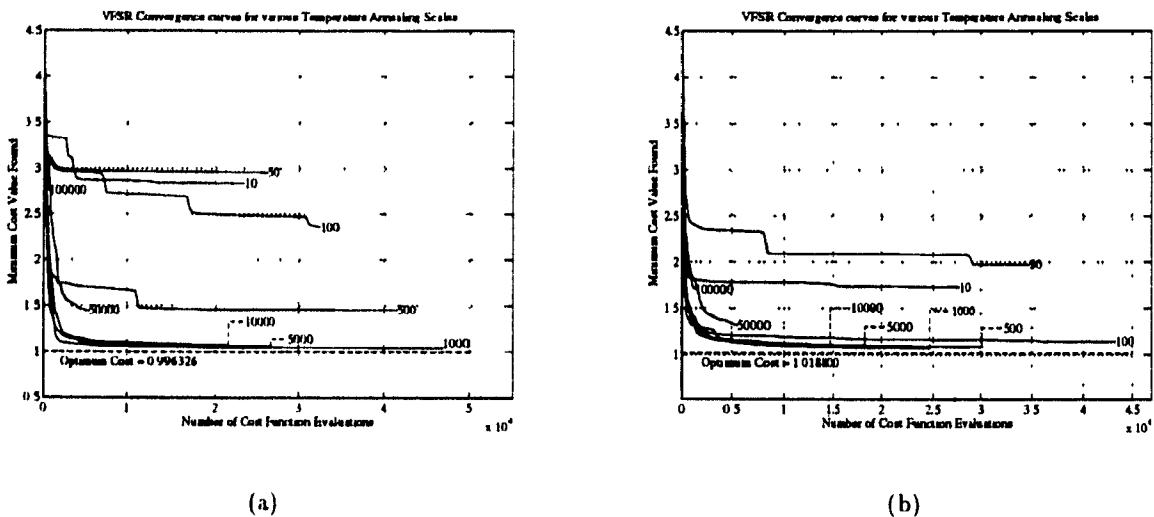


Figure 5.15: These two graphs show convergence curves of the VFSR algorithm obtained by varying the value of the temperature annealing scale parameter (TAS). The registration was performed on two range views of an owl figurine and two range views of a metal pipe. Each curve is the average of 10 VFSR searches. As the graphs indicate, the value of TAS affects both the convergence result (final cost value) and the convergence time (number of cost function evaluations).

5.6.4 Genetic Algorithm Search Versus Very Fast Simulated Reannealing

In Chapter 4, two different optimization techniques were presented: Genetic Algorithms and Very Fast Simulated Reannealing. Both methods are suitable for performing our registration task. This registration is accomplished through the minimization of the cost function defined in Section 3.5.1.

In order to decide which of the two optimization techniques we should use, we compare their relative performance at performing a typical registration. Again, the two range views of the owl figurine and the metal pipe were used for this purpose. Using the best search parameters for the GA and VFSR optimization programs, the views were registered. This was done a total of ten times and the average convergence was computed. Figure 5.17 shows the results obtained.

From the convergence curves obtained, we observe that the VFSR optimization yields better results than the GA optimization. A lower value of the cost function is reached implying that the registration fit is better. The difference in the quality of the fit might be difficult to observe visually, but the value of the cost function reached clearly indicates that VFSR finds a better transformation than GA. The convergence curves also show that

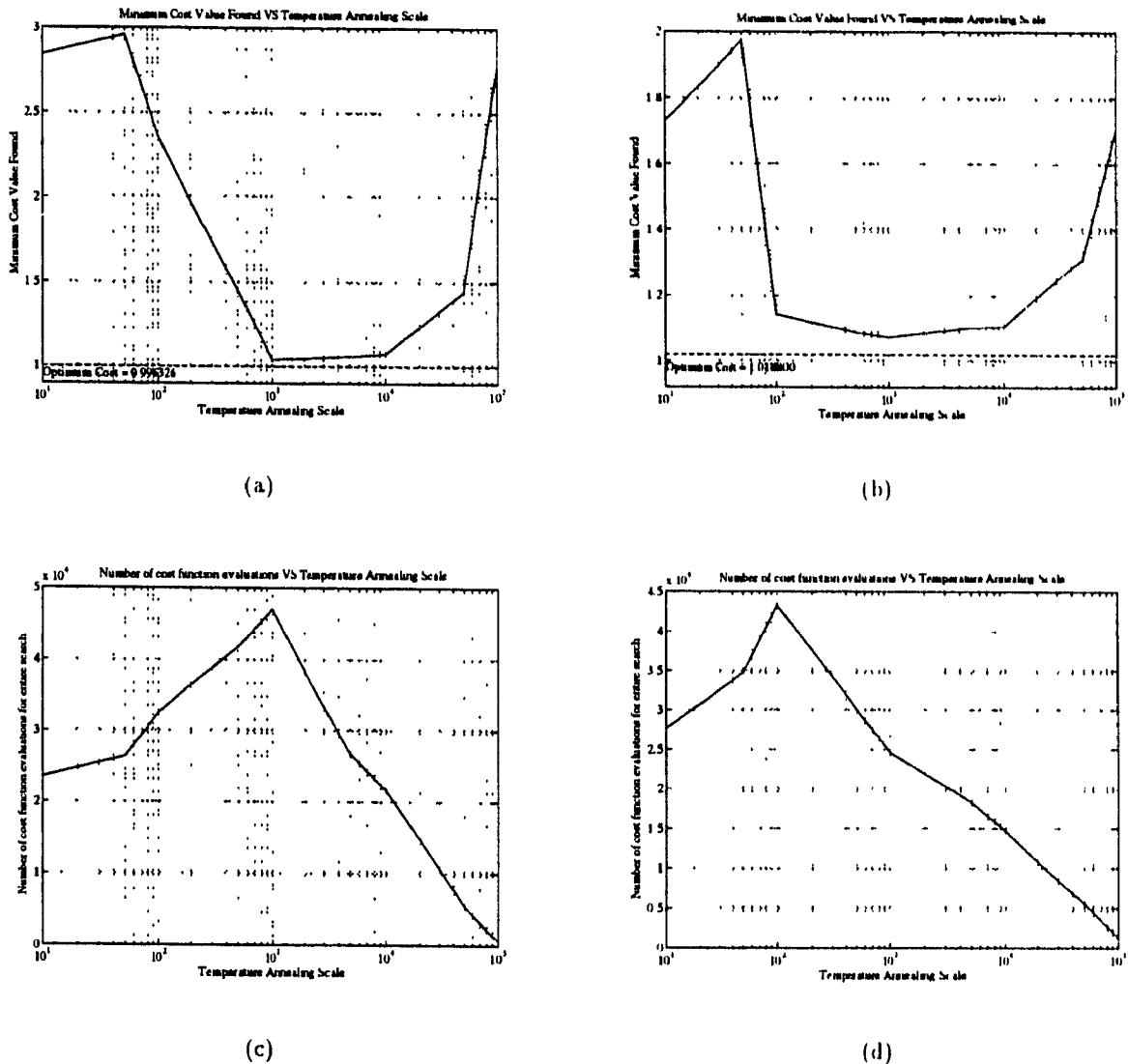


Figure 5.16: These graphs show the results obtained for VFSR in the registration of two sets of range views (owl figurine (a) and (c), metal pipe (b) and (d)). The curves are derived directly from the convergence curves of figure 5.15. Figures (a) and (b) show the final cost value reached at the end of the VFSR optimization search as a function of the temperature annealing scale (TAS). Figures (c) and (d) show the number of cost function evaluation performed before the termination of VFSR as a function of TAS . Figures (a) and (b) indicate that for TAS between 1000 and 10000, the algorithm converges to an acceptable solution (near optimum cost value). From figures (c) and (d) we see that VFSR terminates sooner (after a smaller number of cost function evaluations) with $TAS = 10000$ than with $TAS = 1000$. Therefore, a value of $TAS = 10000$ is adequate for the optimization.

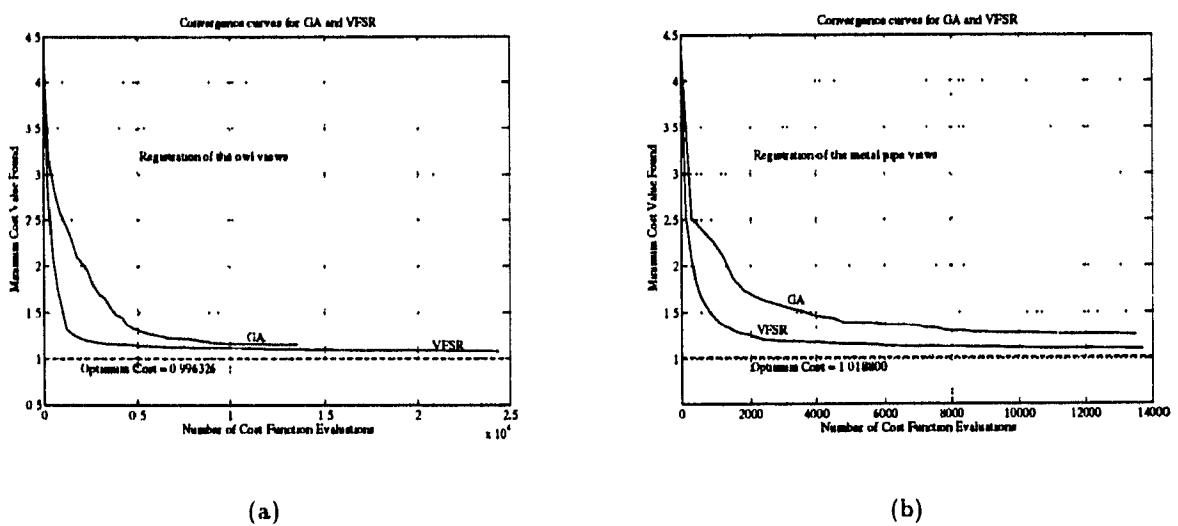


Figure 5.17: The two graphs shown above compare the convergence of the GA and VFSR optimization programs when performing a registration search. Figure (a) shows the convergence curves obtained for the registration of two views of the owl figurine. Figure (b) shows the convergence curves obtained for the registration of two views of the metal pipe. Each convergence curve is obtained by registering the views ten times and by computing the average convergence. The registration was conducted using the best search parameters for GA and VFSR as determined through experimentation. We can clearly see that in both cases VFSR performs better than GA. In both cases, VFSR converges faster than GA and reaches a lower cost solution.

VFSR converges at a faster rate than GA. This can be seen in the convergence curves where the VFSR curve drops down to a low cost value very fast, early in the search. Genetic Algorithms have the reputation of performing very well in the presence of many local optima. From the cost function plots (Figures 4.2, 4.3, 4.4 and 4.5) we see that the cost function has one large minimum with many little bumps due to noise and surface irregularities. Therefore, even though the cost function has many local minima, it can be essentially seen as a single minimum with added noise. In such a case, a GA search might not be the most efficient and here VFSR performs better. Because the VFSR optimization performs better than the GA optimization, all subsequent experiments were conducted using VFSR.

5.7 Dual-View Registration Experiments

This section presents various experiments realized with the registration algorithm described in this thesis. All experiments consist of the registration of two range views. The range views were obtained from different 3D objects.

Sections 5.7.1, 5.7.2, 5.7.3, 5.7.4 and 5.7.5 present registration results obtained for range views of different 3D objects. Each section basically presents the same information about the experiment conducted. Therefore, in order to avoid repeating the same explanations in each experiment, we first present a description of the experiments and of the results that are shown in all cases.

We will refer to Section 5.7.1 when presenting the various elements that constitute the experiments. For each dual-view registration experiment, the following information is presented:

- 1. The object used for the experiment.** A brief description of the object is given.
- 2. The method used for acquiring the range views of the object.** This can be either the turntable or the robot arm (see Section 5.1).
- 3. A figure showing the two views after preprocessing.** The two range views are thresholded to remove all background information and filtered to remove spurious data points. See Figure 5.18.
- 4. The parameters used in the registration algorithm.** The parameters of the

registration search were examined in Section 5.6. There are three sets of parameters, the *Cost Function Parameters*, the *VFSR Parameters* and the *Search Space Parameters*. The parameters defining the search space are the initial transformation estimate and the search range for each of the motion parameters. The transformation estimate between the two views is obtained directly from the acquisition system calibration. See Table 5.2.

Because the turntable is very precise, the transformation estimate it provides is always very near the optimal registration transformation. In order to perform a more rigorous test of the registration algorithm, the motion parameters of the estimate were varied when the turntable was used for the acquisition. A variation of 8 millimeters was added to each of the translation parameters and a variation of 4 degrees was added to each of the rotation parameters. The resulting transformation is used as the initial estimate in the registration search. This was done for the experiments of Section 5.7.1 and 5.7.2.

5. **A figure showing the two views as they are related by the initial transformation estimate.** Figure 5.19 (a) illustrates this. In order to differentiate between the two views, one of the views is rendered as a grid surface and the other is rendered as a shaded surface.
6. **A figure showing the two views registered with the optimum transformation found by the registration algorithm.** Figure 5.19 (b) illustrates this. Here again, one of the views is rendered as a grid surface and the other as a shaded surface. This makes it easier to distinguish the two views in the image and to observe the quality of the registration. An interweaving of the grid and shaded surfaces indicates a high quality registration. This can be seen in figure 5.19 (b) where the mesh of the first view seems to be partially immersed in the shading of the second view.
7. **The value of the cost function for both the initial estimate and the optimum transformation is given.** The value of the cost function indicates the average distance between a control point in the first range view and its corresponding point in the second range view. It is therefore a good indication of the quality of the registration fit. When the views are acquired, the sampled surfaces will contain noise due to the sampling error of the rangefinder camera. The sampling error of the

rangefinder is of the order of 0.625 millimeters. When the views are well registered, the value of the cost function should be close to twice this sampling error, namely 1.25 millimeters (see Section 5.2).

8. **A figure showing the distance between the two views before and after the registration.** As another indication of the registration quality, a display of the Euclidean distance measure between the two range views is provided. As shown in Figure 5.20, the distance is represented as various shades of gray. A dark colored point indicates that the distance between this point and its correspondence in the other view is small. A light colored point indicates a large distance. The minimum distance, 0 millimeters, is indicated with black and the maximum distance, the distance threshold, is indicated by a light gray color. Points without correspondence and points whose distance to their corresponding point in the other view is greater than the distance threshold are shown with the lightest gray color. A gray level scale is provided (Figure 5.20 (e)) to determine the distance between each point in a view and the other view. The initial distance and the final distance between the views is shown. As an example, figures 5.20 (a) and (b) show the distance between the two views when the initial estimate is used and figures 5.20 (c) and (d) show the distance between the two views when the views are registered with the optimum transformation found by the algorithm. The dark regions clearly show where the two views overlap.

5.7.1 View Registration for an Owl Figurine

The object used in this experiment consists of a white owl figurine having a slightly textured surface. Two views are used for the registration experiment. Both views were obtained with the precision turntable. The rangefinder camera is held in a fixed position while the object is scanned, rotated clockwise by 40° and scanned again. The two views thus obtained are shown in Figure 5.18. The setting of the various search parameters used for the registration with VFSR and the results obtained for the registration are shown in Table 5.2. Figure 5.19 (a) shows the two views using the initial transformation estimate and figure 5.19 (b) shows the two views registered with the optimum transformation found by VFSR. Figure 5.19 (a) shows the two views using the initial transformation estimate and figure 5.19 (b) shows the two views registered with the optimum transformation found by VFSR. Figure 5.20 displays

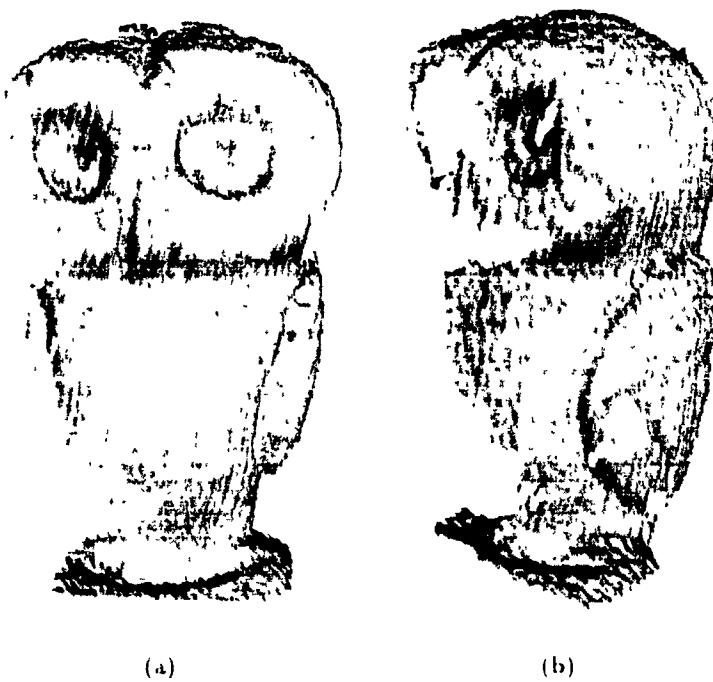


Figure 5.18: Two range views of the owl figurine used in the experiment are shown after preprocessing. The views were obtained using a fixed rangefinder and the owl resting on the platform of a precision turntable. Figure (a) is the frontal view of the owl and Figure (b) is the view obtained after rotating the owl by 40° in a clockwise direction.

the Euclidean distance measure between the two range views as a gray level. We can see that the overlap region of each view is dark, indicating a good registration fit. The cost function value for the initial transformation estimate is 4.30 and the optimum cost function value found is 1.05. This is lower than the expected cost value due to the rangefinder sampling error and is a good indication that the fit is near optimal.

5.7.2 View Registration for a Teapot

The object used for this experiment is a small clay object in the shape of a teapot. Two views are used for the registration experiment. Both views were obtained with the precision turntable. The rangefinder camera is held in a fixed position while the object is scanned, rotated clockwise by 40° and scanned again. The two range views were thresholded to remove all background information and filtered to remove spurious data points. The two views thus obtained are shown in Figure 5.21.

The setting of the various search parameters used for the registration with VFSR and

Parameters used for the registration of the owl range views						
Cost Function Parameters	Control point sample interval			5		
	Overlap factor			0.3		
	Distance Threshold			5.0		
VFSR Parameters	T_0			1.0		
	TRS			0.001		
	TAS			10000		
Search Space Params	t_x	t_y	t_z	r_x	r_y	r_z
T_e from Calibration	-48.374565	-148.612106	83.162674	-33.809300	5.700111	-19.279339
T_e (after variation)	-40.374565	-140.612106	91.162674	-29.809300	9.700111	-15.279339
Search Range	Δ_{tx}	Δ_{ty}	Δ_{tz}	Δ_{rx}	Δ_{ry}	Δ_{rz}
	10.0	10.0	10.0	5.0	5.0	5.0
Initial Cost Value						4.30 mm
Results obtained						
Optimum Transformation found by VFSR	t_x	t_y	t_z	r_x	r_y	r_z
	-48.20997	-149.3088	82.92254	-33.70389	5.699673	-19.13927
Optimum Cost Value						1.05 mm
Number of Cost Function Evaluations						27044
Registration Time (Note: varies with system load during experiment)						5 min. 57 sec.

Table 5.2: Parameters used in the registration of the owl range views and results obtained.

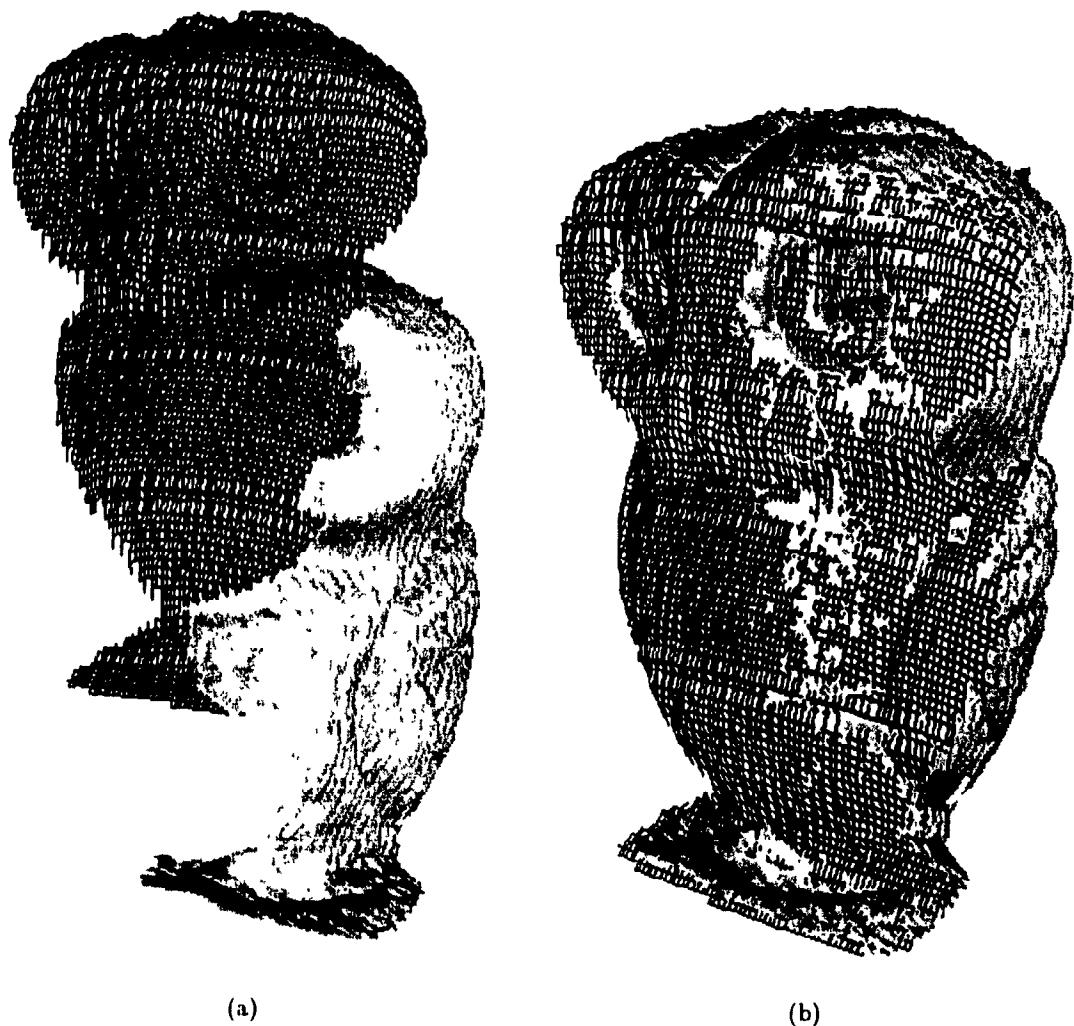


Figure 5.19: Figure (a) shows the relation between the two views of the owl figurine as defined by the initial transformation estimate. Figure (b) shows the registration of the views obtained from the optimum transformation found by the registration algorithm.

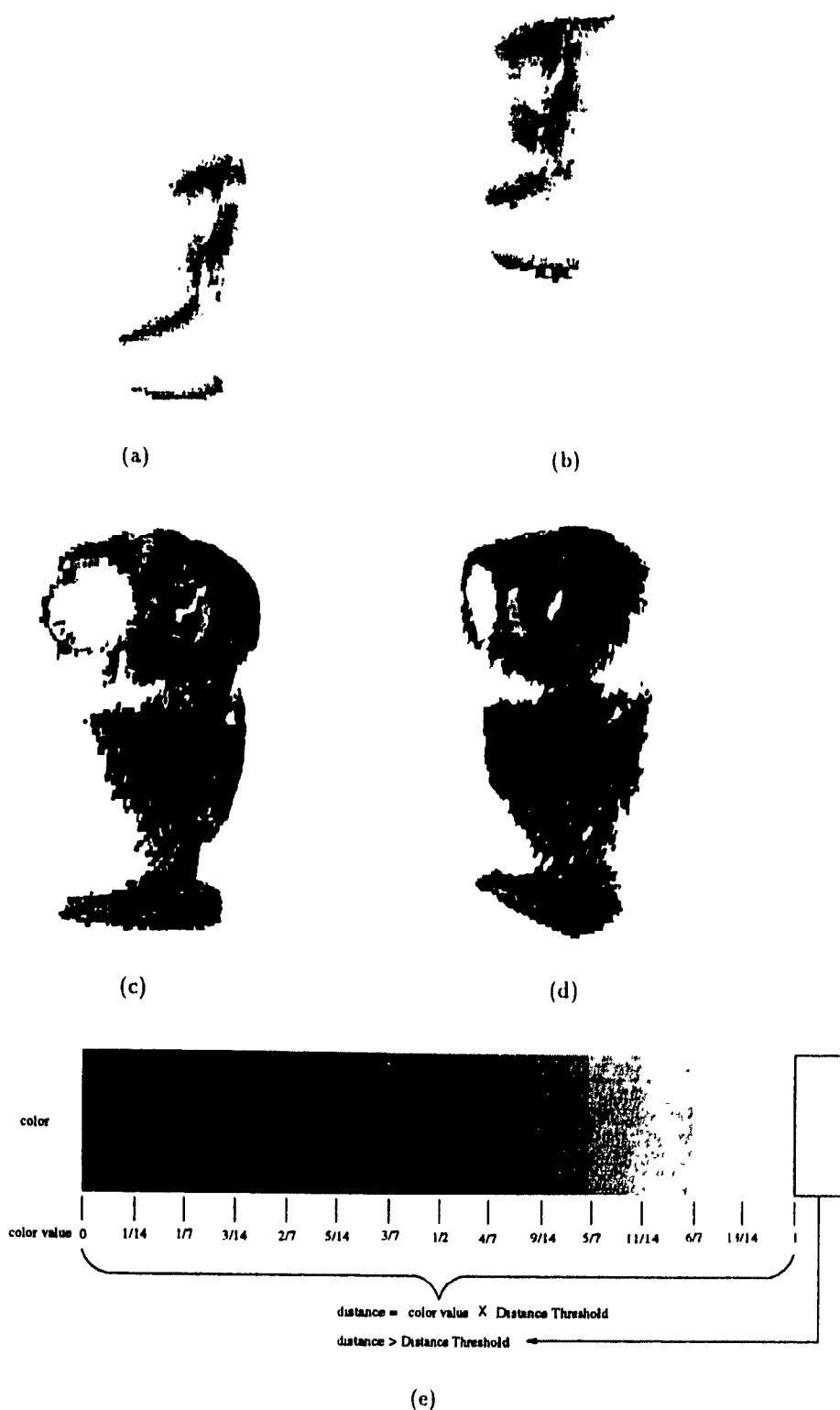


Figure 5.20: Figures (a) and (b) show the distance between the views when the initial transformation estimate is used. Figures (c) and (d) show the distance between the views when registered by the optimum transformation found by the algorithm. Figure (e) shows a scale for interpreting the gray levels of the images. Figures (c) and (d) are very dark indicating a good registration fit of the two range views.

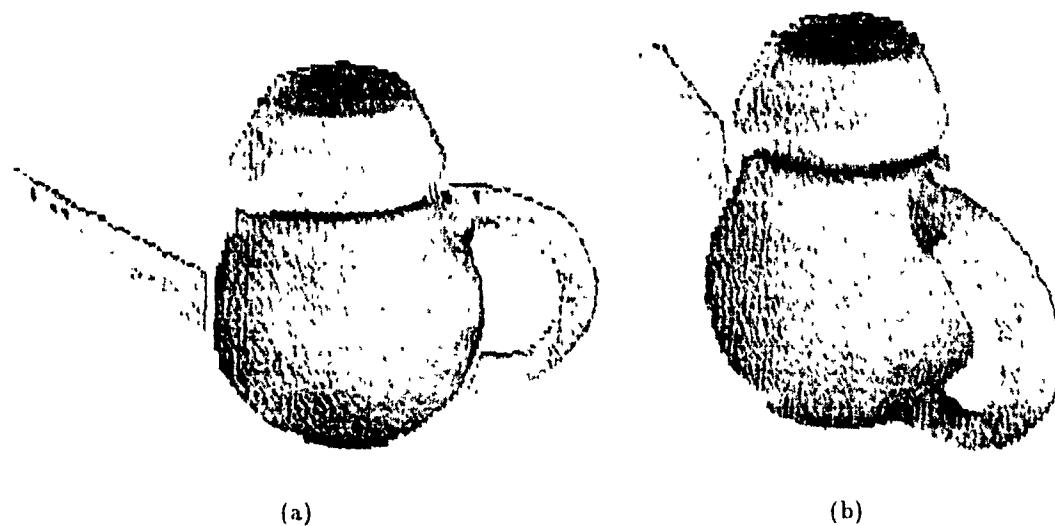


Figure 5.21: Two range views of the clay teapot used in the experiment are shown after preprocessing. The views were obtained using a fixed rangefinder and the teapot resting on the platform of a precision turntable. Figure (a) is the frontal view of the teapot and Figure (b) is the view obtained after rotating the teapot by 40° in a clockwise direction.

the results obtained are shown in Table 5.3. Figure 5.22 (a) shows the two views using the initial transformation estimate and figure 5.22 (b) shows the two views registered with the optimum transformation found by VFSR. Figure 5.22 (a) shows the two views using the initial transformation estimate and figure 5.22 (b) shows the two views registered with the optimum transformation found by VFSR. An interweaving of the grid and shaded surfaces indicates a high quality registration. Figure 5.23 displays the Euclidean distance measure between the two range views as a gray level. We can see that the overlap region of each view is dark, indicating a good registration fit. The cost function value for the initial transformation estimate is 4.39 and the optimum cost function value found is 1.01. This is lower than the expected cost value due to the rangefinder sampling error and is a good indication that the fit is near optimal.

The teapot is a typical object where self-occlusion occurs. As the object is rotated on the precision turntable, the handle hides a portion of the surface which would normally be visible (Figure 5.21 (b)). The results obtained demonstrate clearly that the registration algorithm is robust to self-occlusion.

Parameters used for the registration of the teapot range views						
Cost Function Parameters	Control point sample interval			5		
	Overlap factor			0.3		
	Distance Threshold			5.0		
VFSR Parameters	T_0	TRS	TAS	r_x	r_y	r_z
				1.0		
				0.001		
				10000		
Search Space Params	t_x	t_y	t_z	r_x	r_y	r_z
T_e from Calibration	-50.773682	-146.584274	80.790619	-33.645642	6.491144	-19.108817
T_e (after variation)	-42.773682	-138.584274	88.790619	-29.645642	10.491144	-15.108817
Search Range	Δ_{tx}	Δ_{ty}	Δ_{tz}	Δ_{rx}	Δ_{ry}	Δ_{rz}
	10.0	10.0	10.0	5.0	5.0	5.0
Initial Cost Value						4.39 mm
Results obtained						
Optimum Transformation found by VFSR	t_x	t_y	t_z	r_x	r_y	r_z
	-46.60912	-146.4977	78.82416	-33.281	5.618795	-18.63888
Optimum Cost Value						1.01 mm
Number of Cost Function Evaluations						10606
Registration Time (Note: varies with system load during experiment)						6 min. 34 sec.

Table 5.3: Parameters used in the registration of the teapot range views and results obtained.

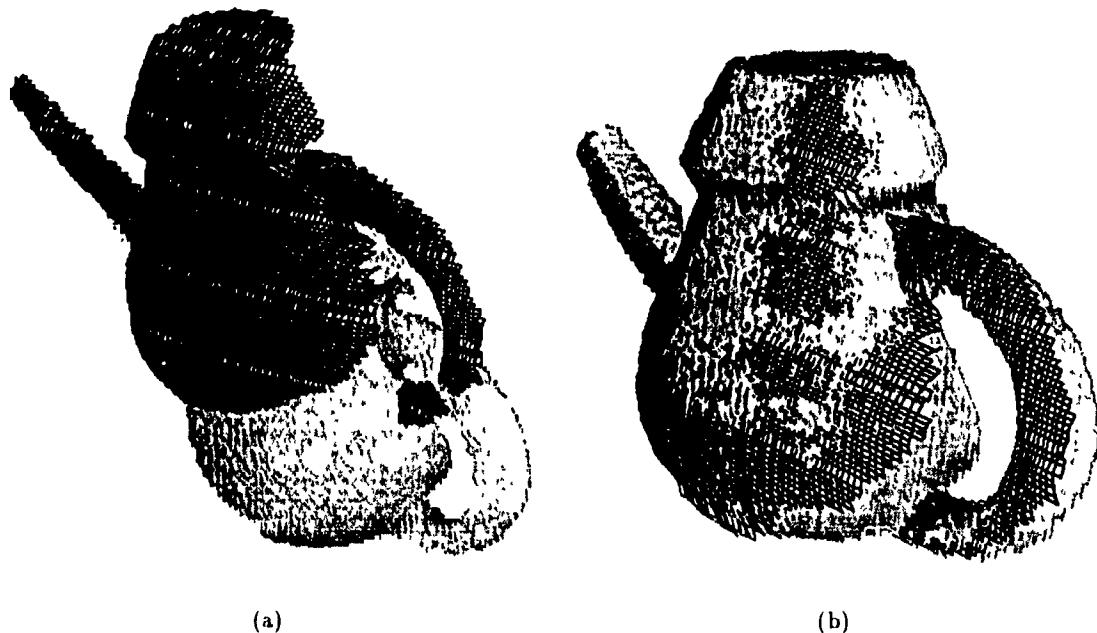


Figure 5.22: Figure (a) shows the relation between the two views of the teapot as defined by the initial transformation estimate. Figure (b) shows the registration of the views obtained from the optimum transformation found by the registration algorithm.

5.7.3 View Registration for a Metal Pipe

The object used for this experiment is a smooth metal pipe with a bend in it. Two views are used for the registration experiment. Both views were obtained with the robot arm. The metal pipe was kept in a fixed position and the rangefinder camera was moved around the object by the robot arm to a new position. The two range views were thresholded to remove all background information and filtered to remove spurious data points. The two views thus obtained are shown in Figure 5.24.

The setting of the various search parameters used for the registration with VFSR and the results obtained are shown in Table 5.4. Figure 5.25 (a) shows the two views using the initial transformation estimate and figure 5.25 (b) shows the two views registered with the optimum transformation found by VFSR. Figure 5.25 (a) shows the two views using the initial transformation estimate and figure 5.25 (b) shows the two views registered with the optimum transformation found by VFSR. An interweaving of the grid and shaded surfaces indicates a high quality registration. Figure 5.26 displays the Euclidean distance measure between the two range views as a gray level. We can see that the overlap region of each

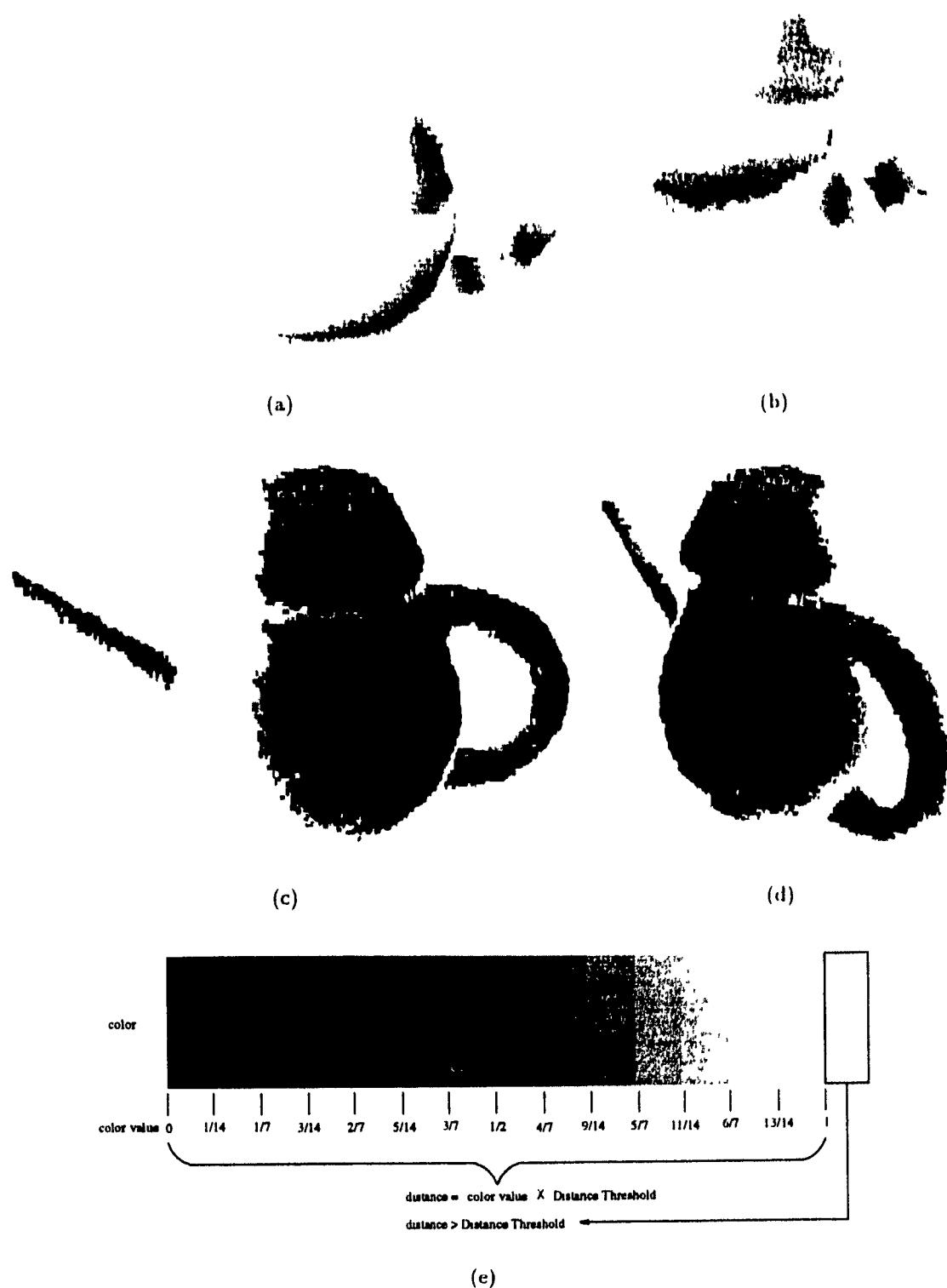


Figure 5.23: Figure (a) and (b) show the distance between the views when the initial transformation estimate is used. Figure (c) and (d) show the distance between the views when registered by the optimum transformation found by the algorithm. Figure (e) shows a scale for interpreting the gray levels of the images. Figure (c) and (d) are very dark indicating a good registration fit of the two range views.

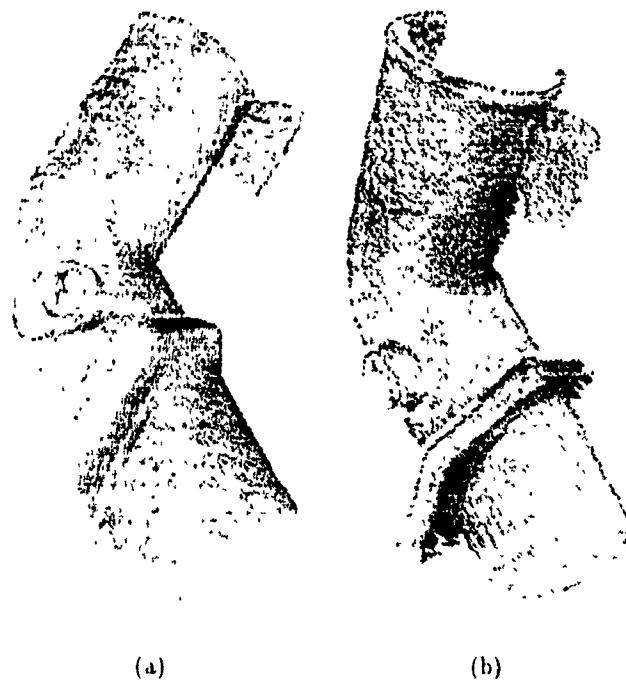


Figure 5.24: Two range views of the metal pipe used in the experiment are shown after preprocessing. The views were obtained using the rangefinder positioned by the robot arm. Figures (a) and (b) show views obtained from two different locations of the rangefinder

view is dark, indicating a good registration fit. The cost function value for the initial transformation estimate is 4.24 and the optimum cost function value found is 1.09. This is lower than the expected cost value due to the rangefinder sampling error and is a good indication that the fit is near optimal.

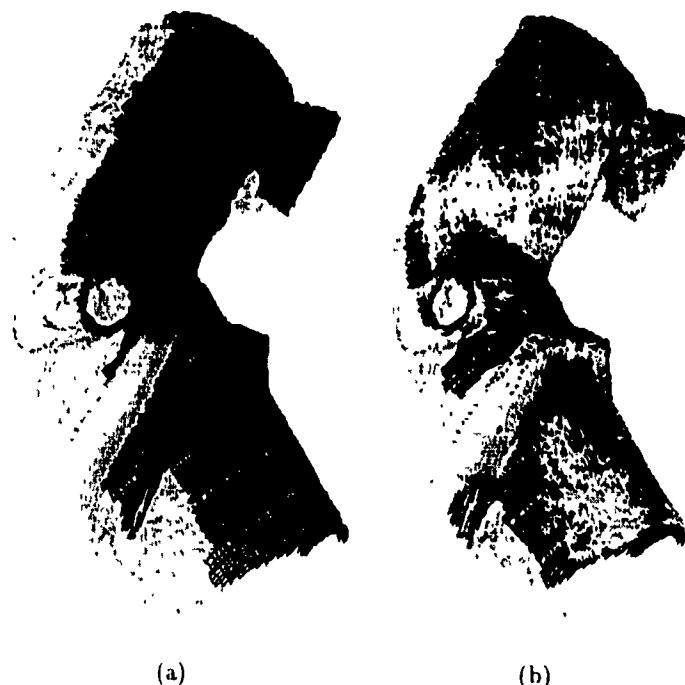
5.7.4 View Registration for a Model Car

The object used for this experiment is a small model car. Two views are used for the registration experiment. Both views were obtained with the robot arm. The car was kept in a fixed position and the rangefinder camera was arbitrarily moved around the object by the robot arm. The two range views were thresholded to remove all background information and filtered to remove spurious data points. The two views thus obtained are shown in Figure 5.27.

The setting of the various search parameters used for the registration with VFSL and the results obtained are shown in Table 5.5. Figure 5.28 (a) shows the two views using the

Parameters used for the registration of the pipe range views						
Cost Function Parameters	Control point sample interval Overlap factor Distance Threshold				5 0.3 5.0	
VFSR Parameters	T_0 TRS TAS				1.0 0.001 10000	
Search Space Params	t_x	t_y	t_z	r_x	r_y	r_z
T_c from Calibration	88.013275	128.029785	35.479187	24.185652	-10.577524	30.736097
Search Range	Δ_{tx} 10.0	Δ_{ty} 10.0	Δ_{tz} 10.0	Δ_{rx} 5.0	Δ_{ry} 5.0	Δ_{rz} 5.0
Initial Cost Value						4.24 min
Results obtained						
Optimum Transformation found by VFSR	t_x 94.44749	t_y 120.9073	t_z 29.15821	r_x 24.47806	r_y -10.58511	r_z 29.48826
Optimum Cost Value						1.09 min
Number of Cost Function Evaluations						15033
Registration Time (Note: varies with system load during experiment)						9 min. 59 sec.

Table 5.4: Parameters used in the registration of the pipe range views and results obtained.



(a)

(b)

Figure 5.25: Figure (a) shows the relation between the two views of the pipe as defined by the initial transformation estimate. Figure (b) shows the registration of the views obtained from the optimum transformation found by the registration algorithm.

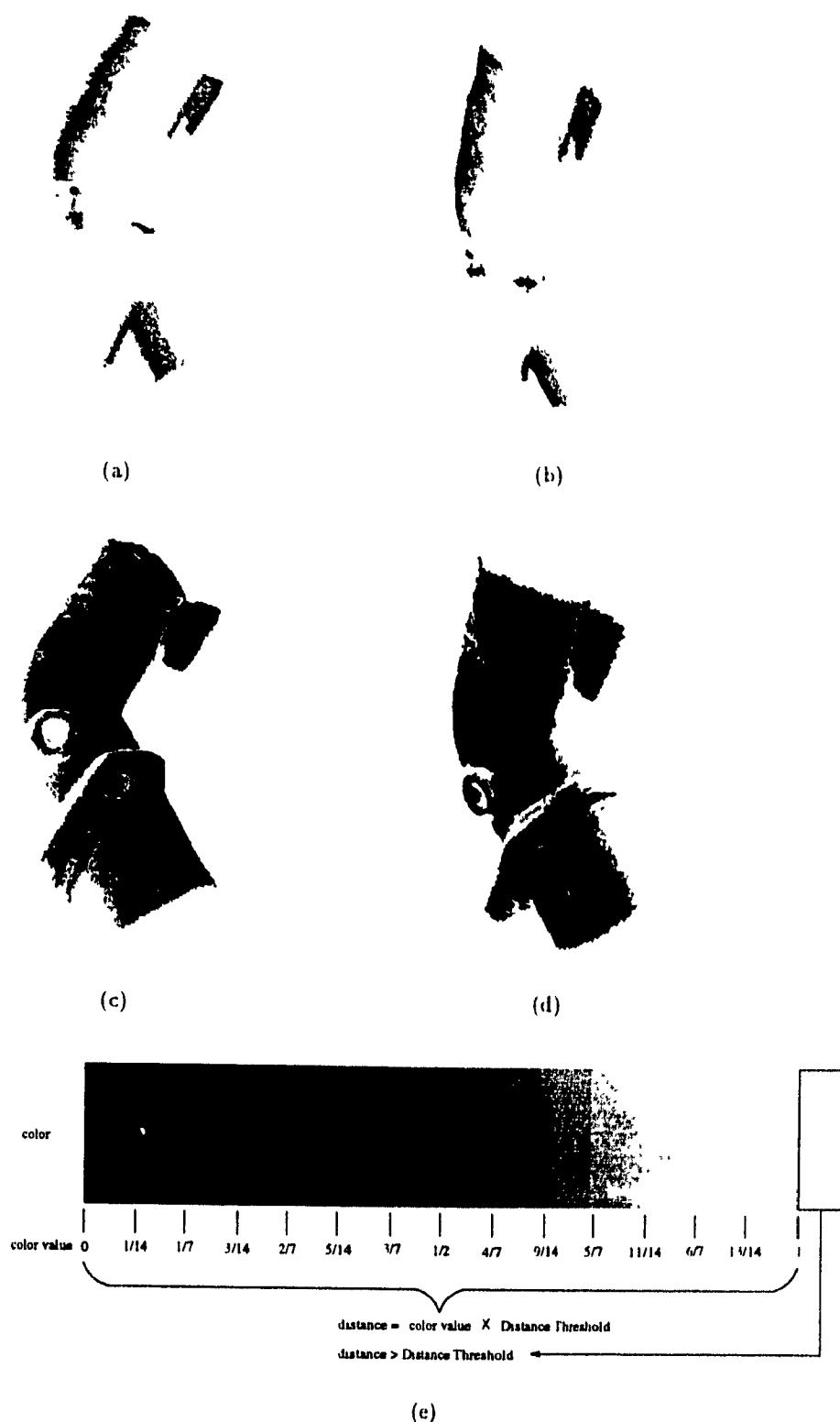


Figure 5.26: Figures (a) and (b) show the distance between the views when the initial transformation estimate is used. Figures (c) and (d) show the distance between the views when registered by the optimum transformation found by the algorithm. Figure (e) shows a scale for interpreting the gray levels of the images. Figures (c) and (d) are very dark indicating a good registration fit of the two range views.

initial transformation estimate and Figure 5.28 (b) shows the two views registered with the optimum transformation found by VFSR. An interweaving of the grid and shaded surfaces indicates a high quality registration. Figure 5.29 displays the Euclidean distance measure between the two range views as a gray level. We can see that the overlap region of each view is dark, indicating a good registration fit. The cost function value for the initial transformation estimate is 4.50 and the optimum cost function value is 2.29.

Although this cost value is larger than the expected cost due to the rangefinder sampling error, the quality of the fit can be inspected visually from figures 5.28 and 5.29. This higher cost function value is the result of correspondence between points that are far apart (but still within the distance threshold) due to the large viewing angle difference between the views. From figure 5.29 (a), we can see that a portion of the roof of the car is pale. This indicates that points in this region are a bit far from their corresponding points in the other view. This is understandable since the distance is not measured along the perpendicular to the surface but along the line of sight of the second view. The second view is more or less a side view and therefore at this angle a small perpendicular distance would result in a large distance between corresponding points. Figure 5.30 illustrates this idea. A visual inspection however, shows that the two views are properly registered.

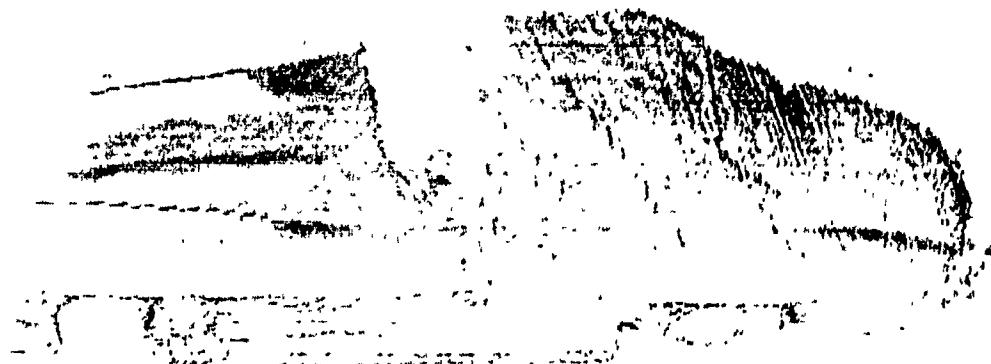
5.7.5 View Registration for Fruits

The objects used for this experiment are three plastic fruits. Two views are used for the registration experiment. Both views were obtained with the robot arm. The fruits were kept in a fixed position and the rangefinder camera was moved around the object by moving the robot arm. The two range views were thresholded to remove all background information and filtered to remove spurious data points. The two views thus obtained are shown in Figure 5.31.

The setting of the various search parameters used for the registration with VFSR and the results obtained are shown in Table 5.6. Figure 5.32 (a) shows the two views using the initial transformation estimate and figure 5.32 (b) shows the two views registered with the optimum transformation found by VFSR. Figure 5.32 (a) shows the two views using the initial transformation estimate and figure 5.32 (b) shows the two views registered with the optimum transformation found by VFSR. An interweaving of the grid and shaded surfaces indicates a high quality registration. Figure 5.33 displays the Euclidean distance measure



(a)

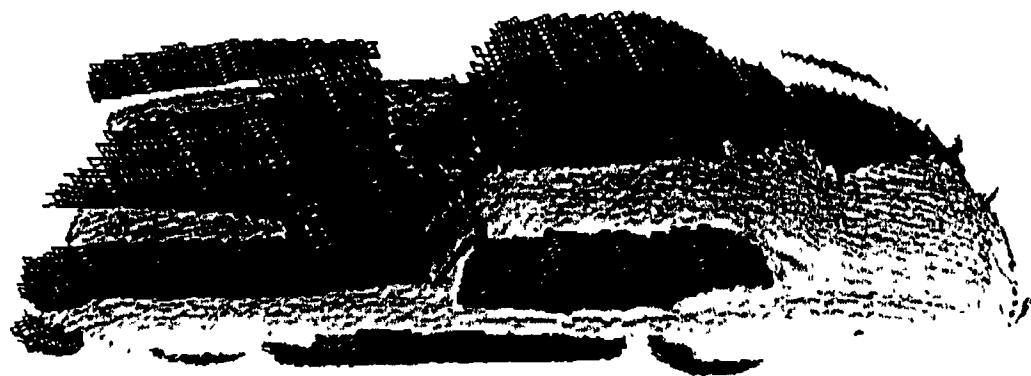


(b)

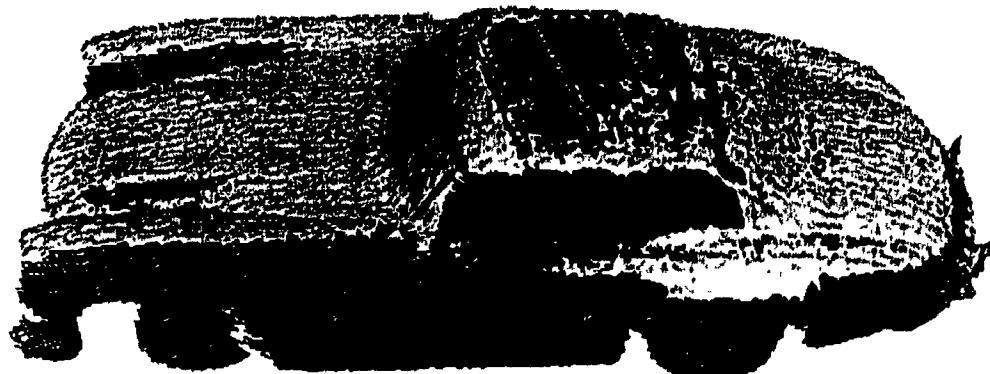
Figure 5.27: Two range views of the model car used in the experiment are shown after preprocessing. The views were obtained using the rangefinder moved around by the robot arm. Figures (a) and (b) show views obtained from two different positions of the rangefinder.

Parameters used for the registration of the car range views						
Cost Function Parameters	Control point sample interval Overlap factor Distance Threshold				5 0.3 5.0	
VFSR Parameters	T_0 TRS TAS				1.0 0.001 10000	
Search Space Params	t_x	t_y	t_z	r_x	r_y	r_z
T_e from Calibration	95.833389	274.947144	168.081055	54.280762	-23.301256	167.649002
Search Range	Δ_{tx} 10.0	Δ_{ty} 10.0	Δ_{tz} 10.0	Δ_{rx} 5.0	Δ_{ry} 5.0	Δ_{rz} 5.0
Initial Cost Value						4.50 mm
Results obtained						
Optimum Transformation found by VFSR	t_x 94.51793	t_y 284.0045	t_z 160.7948	r_x 51.37647	r_y -21.51072	r_z 165.4888
Optimum Cost Value						2.29 mm
Number of Cost Function Evaluations						10359
Registration Time (Note: varies with system load during experiment)						5 min. 59 sec.

Table 5.5: Parameters used in the registration of the car range views and results obtained.



(a)



(b)

Figure 5.28: Figure (a) shows the relation between the two views of the car as defined by the initial transformation estimate. Figure (b) shows the registration of the views obtained from the optimum transformation found by the registration algorithm.

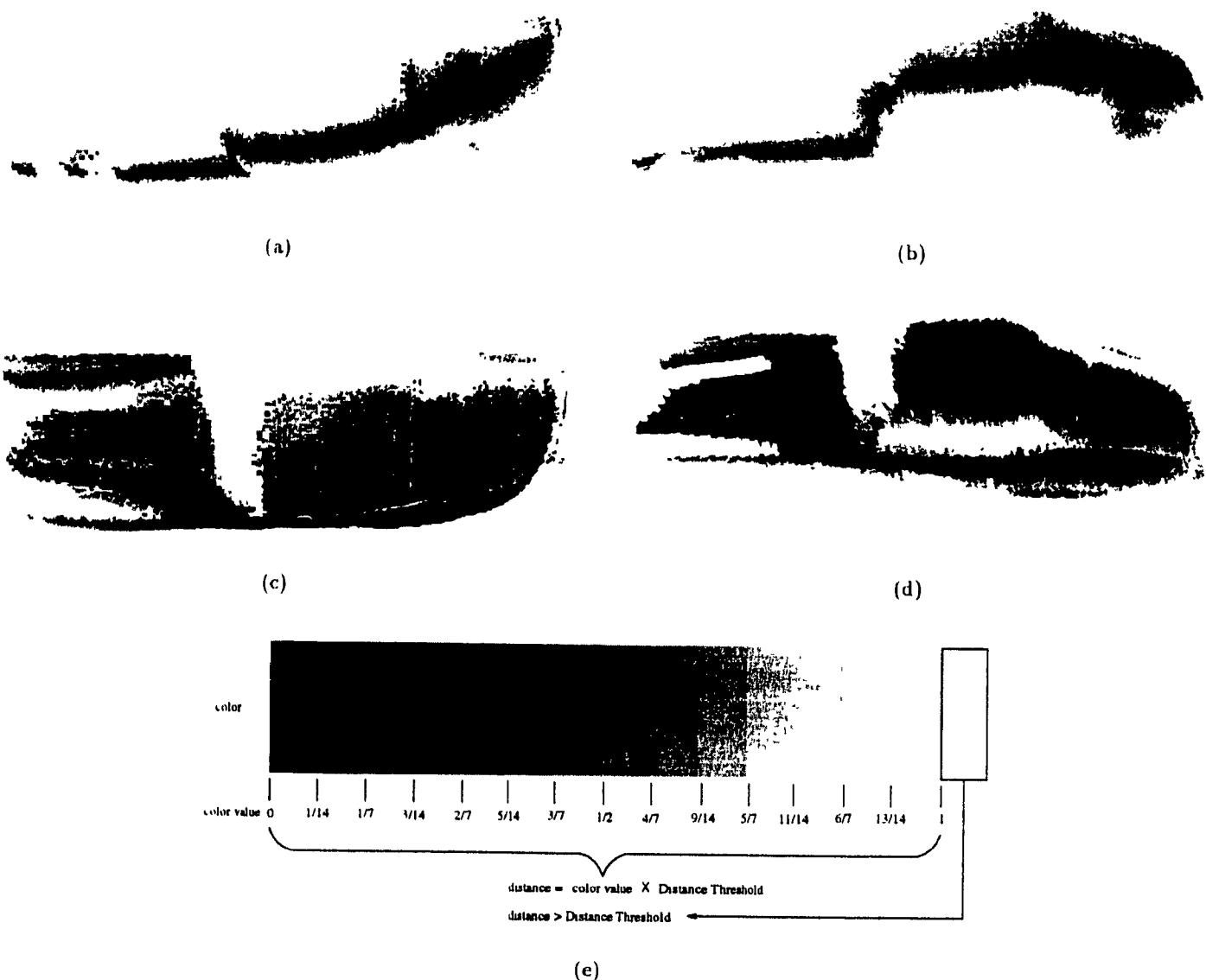


Figure 5.29: Figures (a) and (b) show the distance between the views when the initial transformation estimate is used. Figures (c) and (d) show the distance between the views when registered by the optimum transformation found by the algorithm. Figure (e) shows a scale for interpreting the gray levels of the images. Figures (c) and (d) are very dark indicating a good registration fit of the two range views

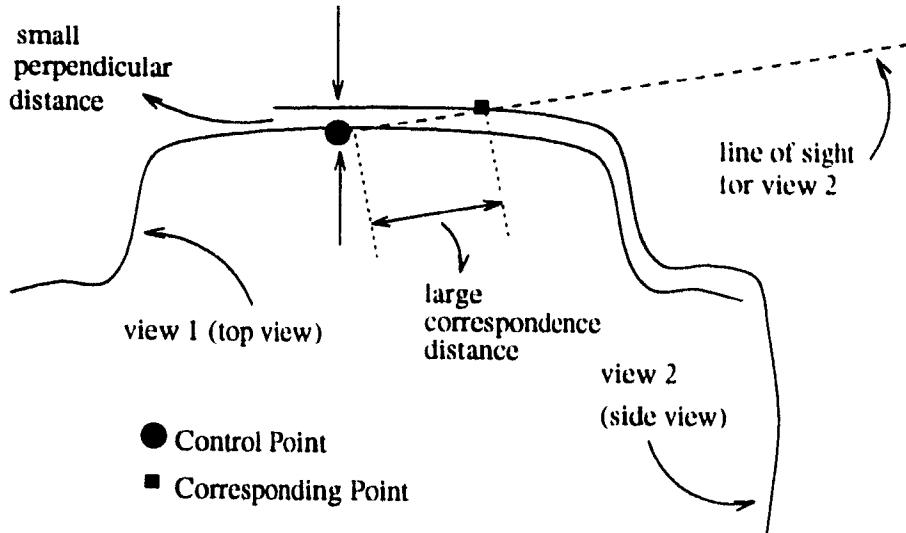


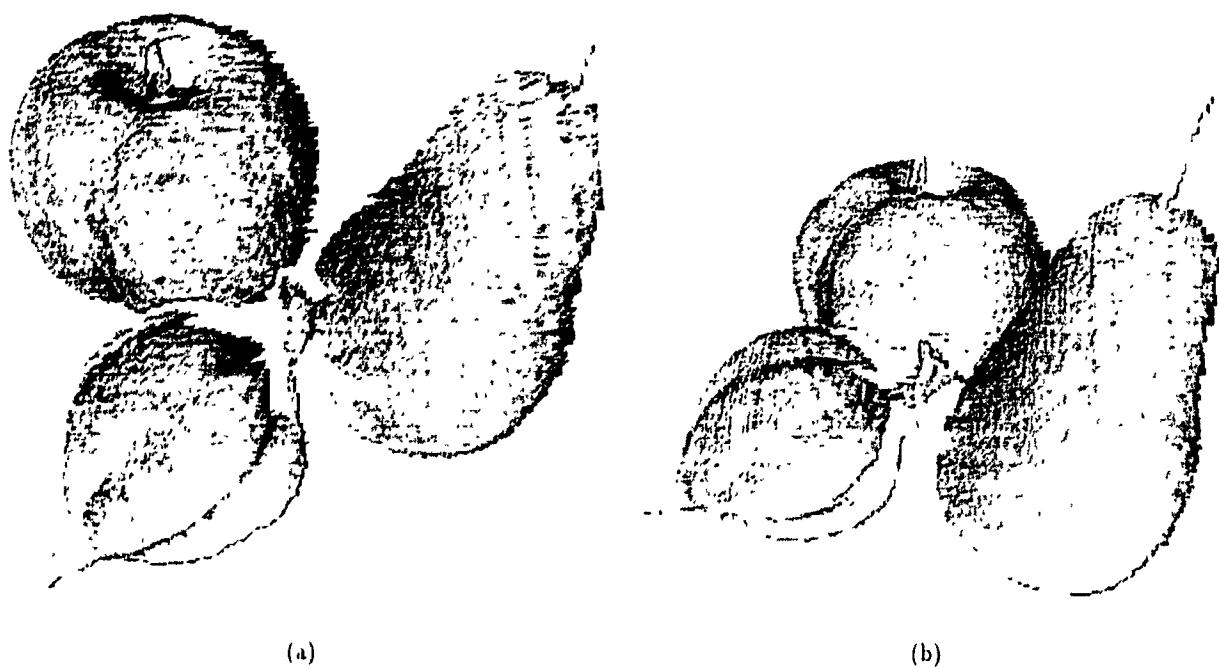
Figure 5.30: This figure illustrates the typical case where the measured distance between corresponding points is much larger than the perpendicular distance between the surfaces being registered. The figure shows a profile view of the car, where the first view is a top view of the car and the second view is a top/side view where the top of the car is still seen, but at a large angle. Because the correspondence between views is established along the scan lines (line of sight), corresponding points on the roof of the car will contribute a large distance to the overall cost function measure.

between the two range views as a gray level. We can see that the overlap region of each view is dark, indicating a good registration fit. The cost function value for the initial transformation estimate is 3.66 and the optimum cost function value found is 1.03. This is lower than the expected cost value due to the rangefinder sampling error and is a good indication that the fit is near optimal.

5.7.6 Multi-View Registration Experiment

The registration of two range views is the first step in the registration of a set of range views. The goal of multi-view registration is to construct a representation of the *complete* surface of an object by registering multiple range views. The registration process consists of integrating the various range views of an object into a unique coordinate frame. Once a complete multi-view registration is accomplished, one can use the result for the generation of models and other higher level tasks such as object recognition, grasping etc.

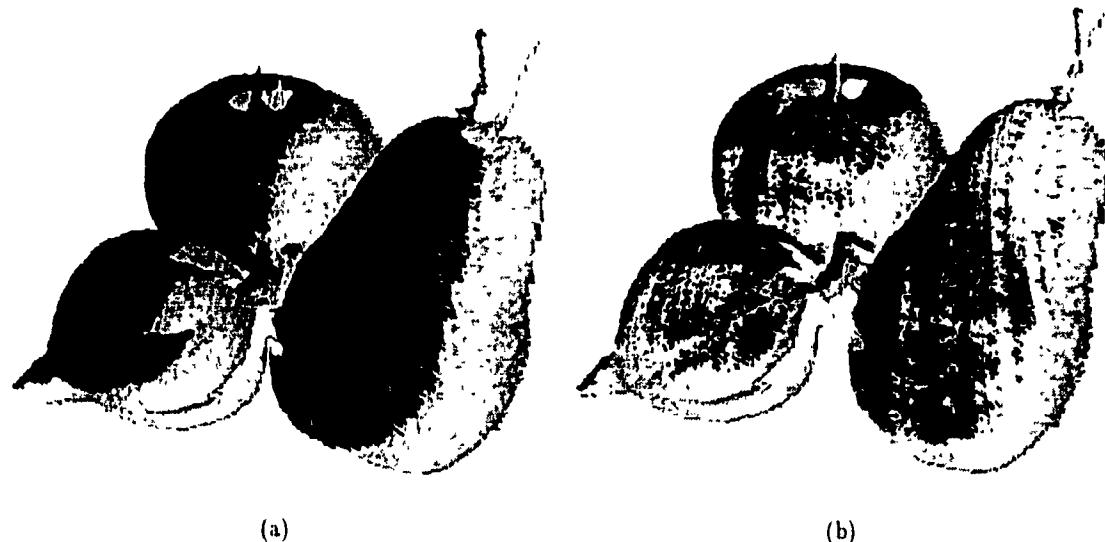
The most straightforward way of performing the registration of multiple views of a 3D object is to register views in pairs. For example, say we acquire six range views of an object numbered from 0 to 5 such that view 0 overlaps view 1, view 1 overlaps view 2, and so



(a)

(b)

Figure 5.31: Two range views of the fruits used in the experiment are shown after preprocessing. The views were obtained using the rangefinder positioned by the robot arm. Figures (a) and (b) show views obtained from two different positions of the rangefinder.



(a)

(b)

Figure 5.32: Figure (a) shows the relation between the two views of the fruits as defined by the initial transformation estimate. Figure (b) shows the registration of the views obtained from the optimum transformation found by the registration algorithm.

Parameters used for the registration of the fruit range views						
Cost Function Parameters	Control point sample interval Overlap factor Distance Threshold				5 0.3 5.0	
VFSR Parameters	T_0 TRS TAS				1.0 0.001 10000	
Search Space Params	t_x	t_y	t_z	r_t	r_y	r_z
T_e from Calibration	-109.662033	-61.974060	64.673950	-14.290806	14.490273	-19.109276
Search Range	Δ_{tx} 10.0	Δ_{ty} 10.0	Δ_{tz} 10.0	Δ_{rx} 5.0	Δ_{ry} 5.0	Δ_{rz} 5.0
Initial Cost Value						3.66 mm
Results obtained						
Optimum Transformation found by VFSR	t_x -99.73408	t_y -62.49808	t_z 60.83395	r_t -13.8953	r_y 14.06274	r_z -17.85085
Optimum Cost Value						1.03 mm
Number of Cost Function Evaluations						11096
Registration Time (Note: varies with system load during experiment)						8 min. 18 sec.

Table 5.6: Parameters used in the registration of the fruit range views and results obtained.

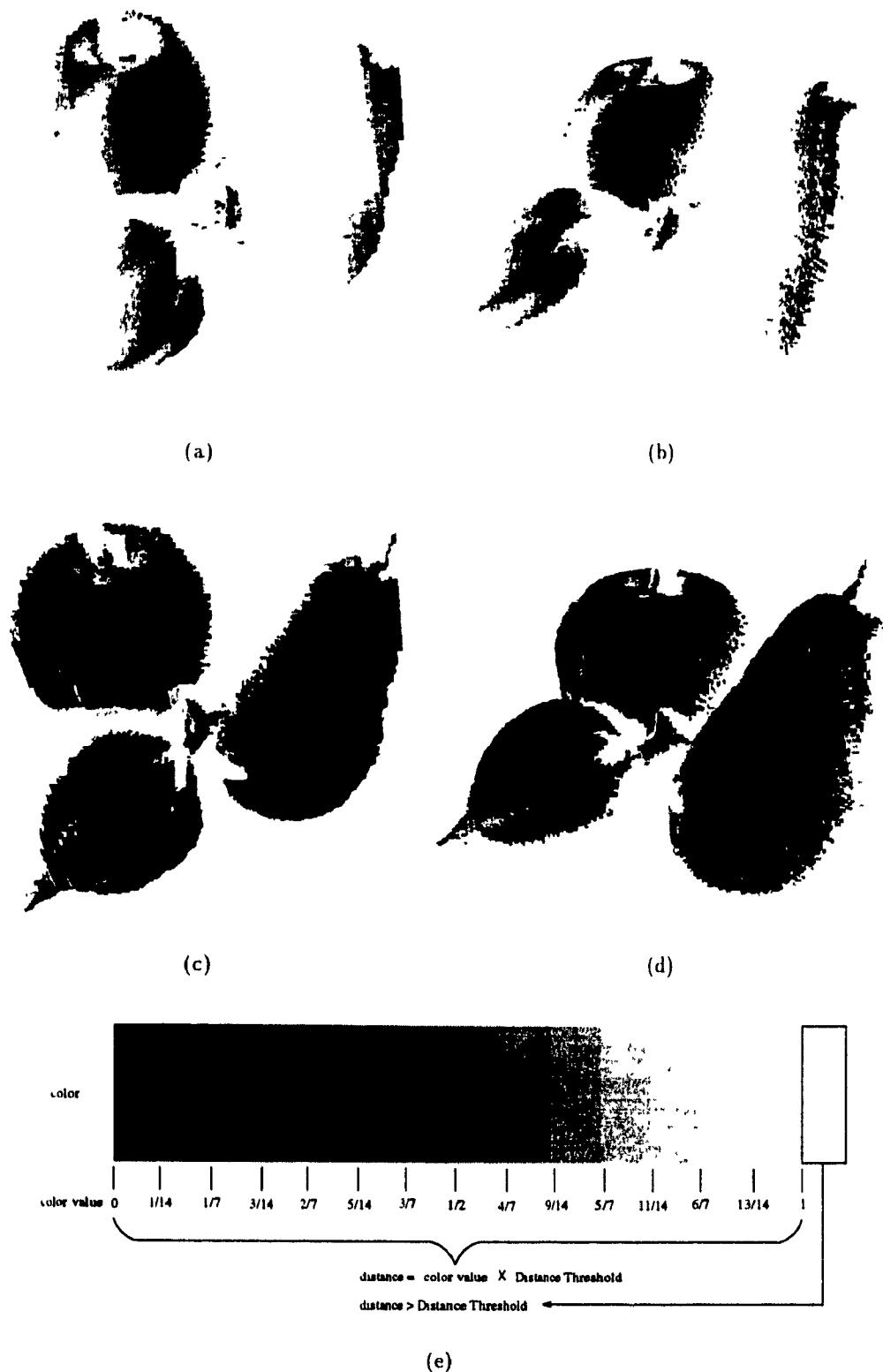


Figure 5.33: Figures (a) and (b) show the distance between the views when the initial transformation estimate is used. Figures (c) and (d) show the distance between the views when registered by the optimum transformation found by the algorithm. Figure (e) shows a scale for interpreting the gray levels of the images. Figures (c) and (d) are very dark indicating a good registration fit of the two range views

on, until view 5 overlaps view 0, thus completing a circle around the object (assuming that the views were obtained by moving the rangefinder camera in a complete circle around the object in order to obtain its full description). With our six range views, we would thus register view 0 with view 1, view 1 with view 2, and so on. We do not register view 5 with view 0 since the transformation between view 5 and view 0 is indirectly specified by the previous transformations. Let the transformation between view i and view j be T_{ij} , then to register all six views, we need to determine five transformations: T_{01} , T_{12} , T_{23} , T_{34} and T_{45} . With these five transformations, the motion relationship between any two range views is completely defined. For example, the transformation between view 5 and view 0 is given by Equation (5.5).

$$\begin{aligned} T_{50} = T_{05}^{-1} &= (T_{45} \circ T_{34} \circ T_{23} \circ T_{12} \circ T_{01})^{-1} \\ &= T_{01}^{-1} \circ T_{12}^{-1} \circ T_{23}^{-1} \circ T_{34}^{-1} \circ T_{45}^{-1} \end{aligned} \quad (5.5)$$

where $T_a \circ T_b$ is the composition of transformation T_a with transformation T_b .

The main problem with this approach is that even though the five transformations found between pairs of views might be optimal, the global registration might not be. A typical problem arising with this is that all the views from view 0 to view 5 seem to be properly registered but view 5 and view 0 are relatively poorly registered. This is a consequence of accumulated errors in the registrations from view 0 to view 5. View 0 and 1 are well registered by T_{01} , but a very small error is present. The same is true of all transformations. This small error in each intermediate transformations is compounded so that by the time the last view wraps around, back to the first view, a large error is present.

To avoid this problem, the views can be registered simultaneously and the error between the first and last view can be taken into consideration. The experiment conducted here uses this approach. A new cost function is defined for the global registration of multiple range views. The new cost is simply the sum of the individual cost of all *direct* transformations plus the cost of any *indirect* transformations. A *direct* transformation is one that directly relates two range views. For example, transformation T_{01} relates all the points in view 0 to the local coordinate system of view 1. An *indirect* transformations is one that indirectly relates one view to another through a series of direct transformations. An indirect transformation is completely defined by direct transformations. For example, transformation T_{50} is an indirect transformation and is completely defined by five direct transformations as shown

by Equation (5.5). The new cost function to be minimized is thus

$$\text{globalcost} = \text{cost}(T_{01}) + \text{cost}(T_{12}) + \text{cost}(T_{23}) + \text{cost}(T_{34}) + \text{cost}(T_{45}) + \text{cost}(T_{50}) \quad (5.6)$$

where $\text{cost}(T)$ is the cost function defined in Equation (3.37). With this new cost function, a small error in all of the direct transformations is reflected as a large error in the indirect transformation. This global optimization ensures that all views fit properly together, not just in pairs.

A multi-view registration experiment was performed for the same owl figurine described in Section 5.7.1. A total of five range views were used. The data acquisition was performed with the precision turntable. The object was rotated by 60 degrees between views. Range views at 0, 60, 120, 180, 240 and 300 degrees were obtained. The search parameters used in the registration and the results obtained are shown in Table 5.7.

Figure 5.34 illustrates the problem arising when the range views are registered two at a time. In this case, view 0° was registered with view 60° , the 60° view with the 120° view, 120° with 180° , 180° with 240° and finally 240° with 300° . Figure 5.34 (a) shows all the views together as seen from above. View 0° is rendered as a grid surface and all other views are rendered as shaded surfaces. This helps in distinguishing views 0° and 300° from one another. We see that all the views appear to be well registered all around but that the last view (300°) and the first view (0°) are poorly registered. A large gap is present between these two views. The gap results from an accumulated error in all the transformations between the first and last view. Figure 5.34 (b) is the same group of registered range views seen from the side. We can see the gap between the first and last view. Figure 5.34 (c) shows an enlargement of a section of figure (b) to show more clearly the large distance between the first and last view.

Figure 5.35 shows the result obtained when global optimization is performed to register the six range views. As can be seen from Table 5.7, the time required to perform the global registration is extremely large. This is due to the fact that when the number of dimensions increases linearly, the annealing time increases exponentially (see Equation 4.20). When registering range views pairwise, the search space has 6 dimensions (one for each motion parameter). When six range views are registered simultaneously, the search space has 30

Parameters used for the multi-view registration of the owl						
Cost Function Parameters	Control point sample interval Overlap factor Distance Threshold				5 0.6 5.0	
VFSR Parameters	T_0 TRS TAS				1.0 0.001 10000	
Search Space Params	t_x	t_y	t_z	r_x	r_y	r_z
T_e from Calibration	-88.974091	-189.533615	152.127502	-49.844002	12.105737	-27.003699
Search Range	Δ_{tx} 2.0	Δ_{ty} 2.0	Δ_{tz} 2.0	Δ_{rx} 1.0	Δ_{ry} 1.0	Δ_{rz} 1.0
Initial Cost Value						4.30 mm
Results obtained (pairwise local registrations)						
Optimum Transformations	t_x	t_y	t_z	r_x	r_y	r_z
$T_{0^\circ-60^\circ}$	-80.19219	-190.2726	144.5746	-48.30938	10.68813	-26.0654
$T_{60^\circ-120^\circ}$	-87.16365	-188.0008	144.1832	-47.94806	11.73128	-27.74214
$T_{120^\circ-180^\circ}$	-87.70704	-189.8285	149.0761	-49.23435	11.84017	-28.28247
$T_{180^\circ-240^\circ}$	-84.26648	-187.5514	143.0579	-47.77332	11.47543	-26.6186
$T_{240^\circ-300^\circ}$	-88.60028	-187.3025	145.4442	-48.2851	12.15576	-27.7237
Which Local Registration	$T_{0^\circ-60^\circ}$					
Optimum Cost Value	1.58 mm					
Number of Cost Function Evaluations	10720					
Registration Time	4m 56s					
Results obtained (global registration)						
Optimum Transformations	t_x	t_y	t_z	r_x	r_y	r_z
$T_{0^\circ-60^\circ}$	-90.71757	-189.848	152.031	-49.70864	12.76964	-27.34013
$T_{60^\circ-120^\circ}$	-90.84272	-191.3197	154.0251	-50.10959	12.79042	-26.89138
$T_{120^\circ-180^\circ}$	-90.41573	-189.4754	150.6458	-49.32046	12.42318	-27.7174
$T_{180^\circ-240^\circ}$	-89.9021	-190.2839	152.058	-49.82738	12.46874	-27.62393
$T_{240^\circ-300^\circ}$	-87.61896	-191.2735	152.6673	-49.89927	12.21386	-26.53895
Optimum Cost Value	1.55 mm					
Number of Cost Function Evaluations	10654822					
Registration Time	83 hr. 17 min. 53 sec.					

Table 5.7: Parameters used in the global registration of the owl views and results obtained.

dimensions (5 transformations of 6 parameters each). Because of this, the annealing time is very large. This experiment would have taken a very long time to complete, but the VFSR minimization was intentionally terminated prematurely in order to avoid waiting for total duration of the annealing process. To reduce the time required to perform the global registration, the algorithm could be partially parallelized. The global cost function defined in Equation (5.6) is a sum of partial cost values, one for each of the transformations involved in the global registration. These individual costs could be computed on separate processors thus dividing the work significantly.

Nevertheless, the results obtained with a global registration are better than with the pairwise local registration. We can now see that the first view and the last view are properly registered. As in Figure 5.34, the registered views are shown from above (a), from the side (b) and an enlargement of a section of figure (b) is shown. This time, there are no gap present between the first view (shown as a grid) and the last view.

Figure 5.36 shows a comparison of the registration quality obtained with the pairwise local and the global registration approach. The cost function value was evaluated between each pair of range views. With the pairwise registration, even though each local fit is good, we can clearly see the large registration error present between the 300° and 0° views of the owl as indicated by the high cost function value (3.18 mm).

Figure 5.37 shows the distance between the first and last view when the pairs of range views are registered locally (Figures 5.37 (c) and (d)) and when the views are globally registered (Figures 5.37 (e) and (f)). The two views are colored with various gray levels indicating the distance between them. The darker the surface, the better the fit. A black region means that the two views are in contact at that point (distance of zero). Figures (a) and (b) show the shaded views. In figures (c) and (d), we can see that the fit between the views is very poor compared with the fit obtained when global registration is performed, as shown in figures (e) and (f). Figure 5.37 (g) shows the scale needed to interpret the images.

Finally, the views were processed in order to remove redundant data. The overlapping surfaces were eliminated by replacing them with their average. The result obtained is a set of points representing the complete surface of the owl figurine². Different views of the

²This data integration program was designed by Kenong Wu and was used here to show the feasibility of generating a simple 3D model once the views have been registered. Kenong Wu is currently working towards his PhD in the field of computer vision at the McGill Research Center for Intelligent Machines.

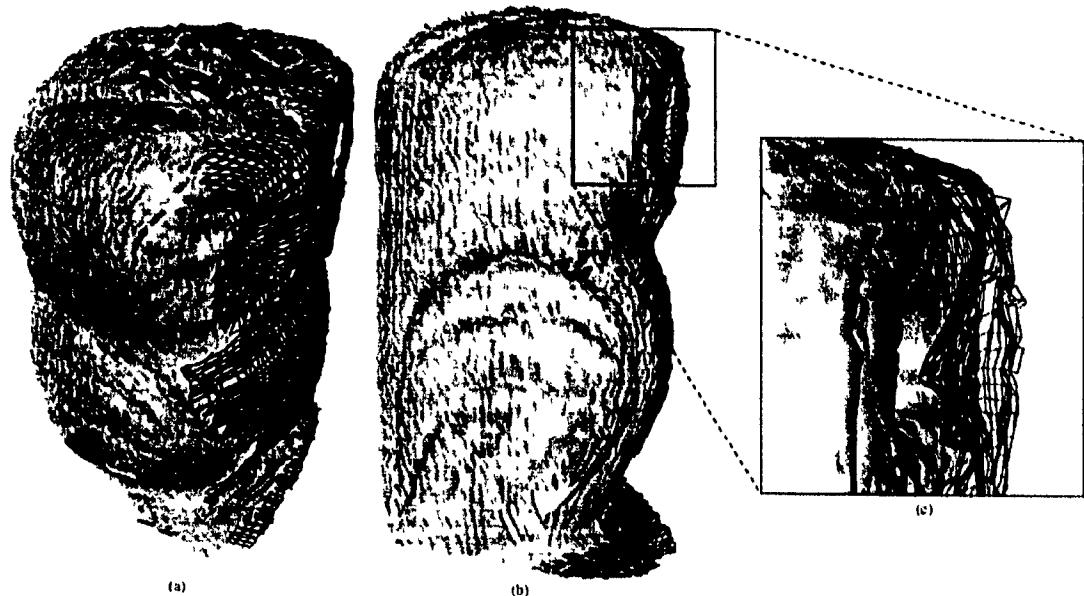


Figure 5.34: This figure illustrates the problem arising when registering multiple views without performing a global optimization. Local optimization is performed to register range views two at a time. In figure (a) we see that although the fit between consecutive views is optimized, the first view (shown as a grid) and last view are poorly registered. Figure (b) shows the same group of registered range views seen from the side, illustrating the gap between the views more clearly. Figure (c) is an enlarged portion of figure (b) showing the large distance between the first and the last view.

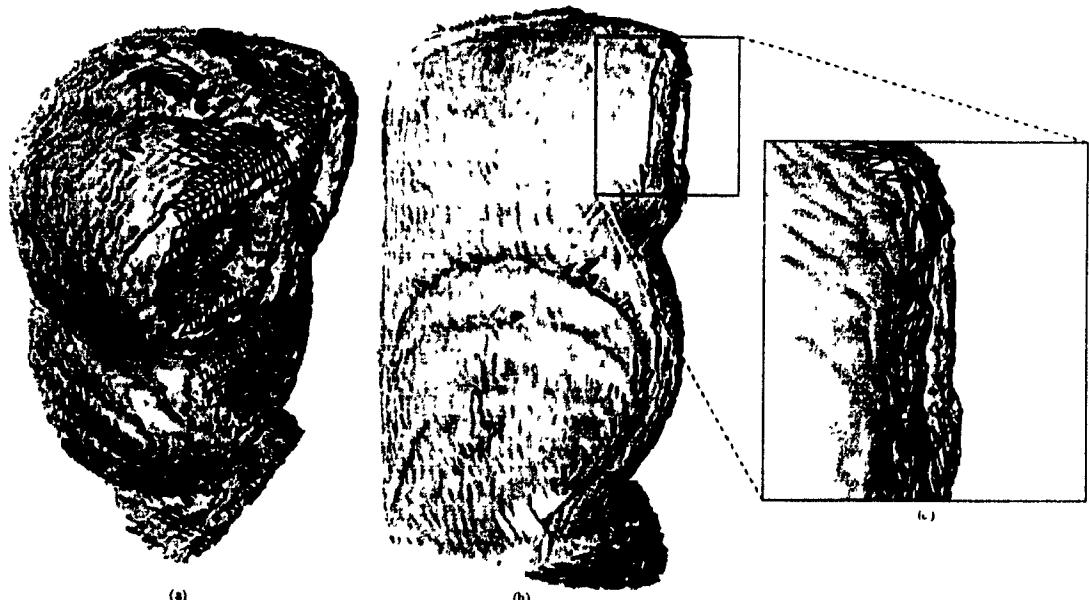


Figure 5.35: This figure shows the results obtained when registering multiple views using a global optimization approach. Six range views were registered simultaneously. The result is shown as seen from above (figure (a)) and from the side (figure (b)). Figure (c) is an enlargement of a section of figure (b) showing how closely the first view (shown as a grid) and the last view are registered.

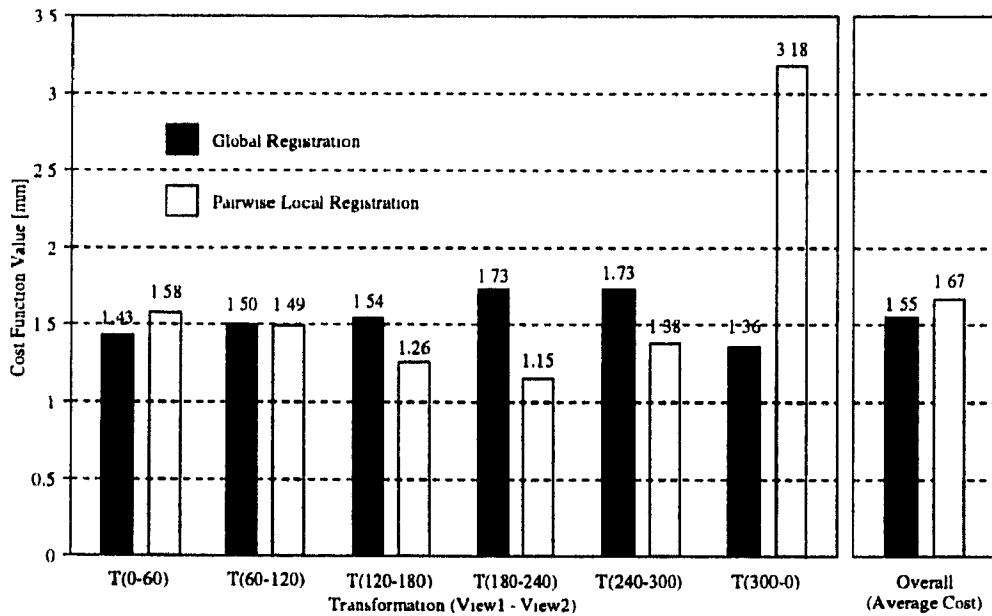


Figure 5.36: This figure compares the respective registration quality obtained with global (black bars) and pairwise local (gray bars) registration. The graph demonstrates that even though the registration quality (evaluated with the cost function) is locally better when range views are registered in pairs, the overall fit is better when global registration is performed. We can see that a large registration error is present between the first (0°) and last (300°) view of the owl, where the cost function evaluated between these two views yields a value of 3.18 mm. When global registration is performed, this error is eliminated and the overall cost function is lower (1.55 mm).

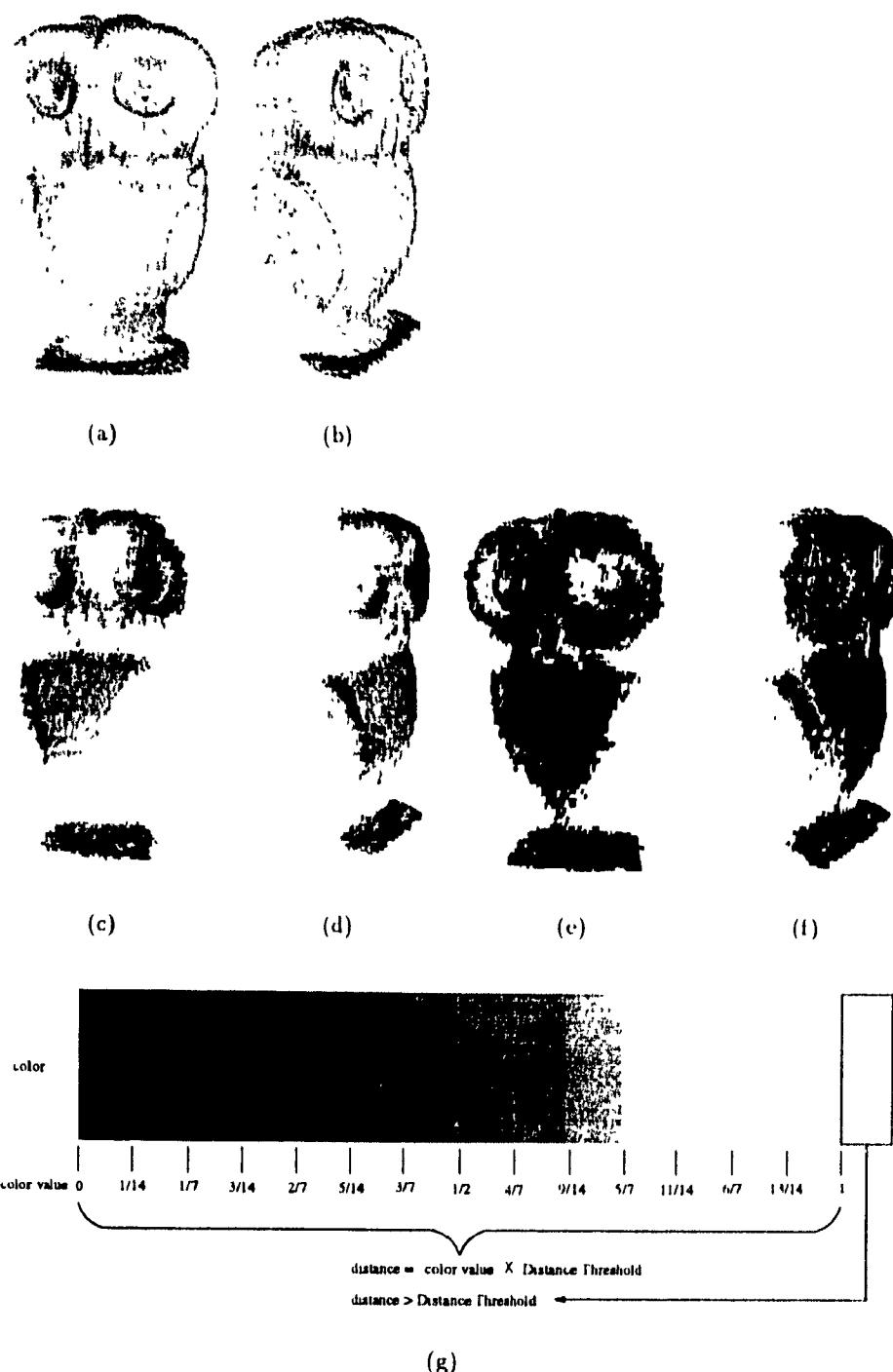


Figure 5.37: Figures (a) and (b) show the first (0°) and last (300°) view of the multi-view registration experiment conducted. Figures (c) and (d) show how simply doing a local registration between consecutive views results in a large registration error. When all the views are registered simultaneously, this error is eliminated and the fit is good all around. This is shown in figures (e) and (f) where we see that the two views overlap very closely. Figure (g) shows a scale to interpret the gray levels of the images.

model obtained are shown in Figure 5.38. The owl is rendered by fitting a small shaded plane at every point of the model surface. Figure 5.38 (a) shows all the points forming the model. Each point is formed by a small square plane of 0.3 millimeter in size. The model is shown on a black background in order to ease the visualization of the points. When each point is fit with a square plane of 1 millimeter in size, the complete surface of the owl is reconstructed. Figures 5.38 (b), (c) and (d) show the model seen from various orientations.

5.8 Summary

In this section, the results of various experiments were presented. First, the hardware setup used to conduct the various experiments was described. To acquire multiple range images of an object, two data acquisition systems were used. The first system consists of a fixed rangefinder camera combined with a precision turntable. The object is rotated by the table between each view acquisition made by the rangefinder. The second acquisition system consists of a robot arm on which the rangefinder camera has been mounted. The robot arm is used to position the camera around an object to permit the acquisition of different range views.

Before registration, range views are processed in order to remove all background surfaces and spurious data points and the initial transformation estimate between views is determined. A calibration tool is used with the precision turntable to determine the position of the axis of rotation of the turntable with respect to the rangefinder camera. This is then used to compute the initial transformation between views acquired with the turntable. When the range views are acquired with the robot arm, the transformation estimate between views is computed from the absolute position of the camera provided by the robot arm controller.

The sampling error of the rangefinder camera was examined. It was found that the measurement error was Gaussian distributed and that the mean absolute error was proportional to the distance between the rangefinder and the surface being sampled.

Two optimization methods, Genetic Algorithms and Very Fast Simulated Reannealing, were studied with regard to their respective performance. The optimum search parameters were determined for each optimization method and the two were compared in terms of speed of convergence and optimum solution reached. Very Fast Simulated Annealing was found to perform better than Genetic Algorithms in all cases.

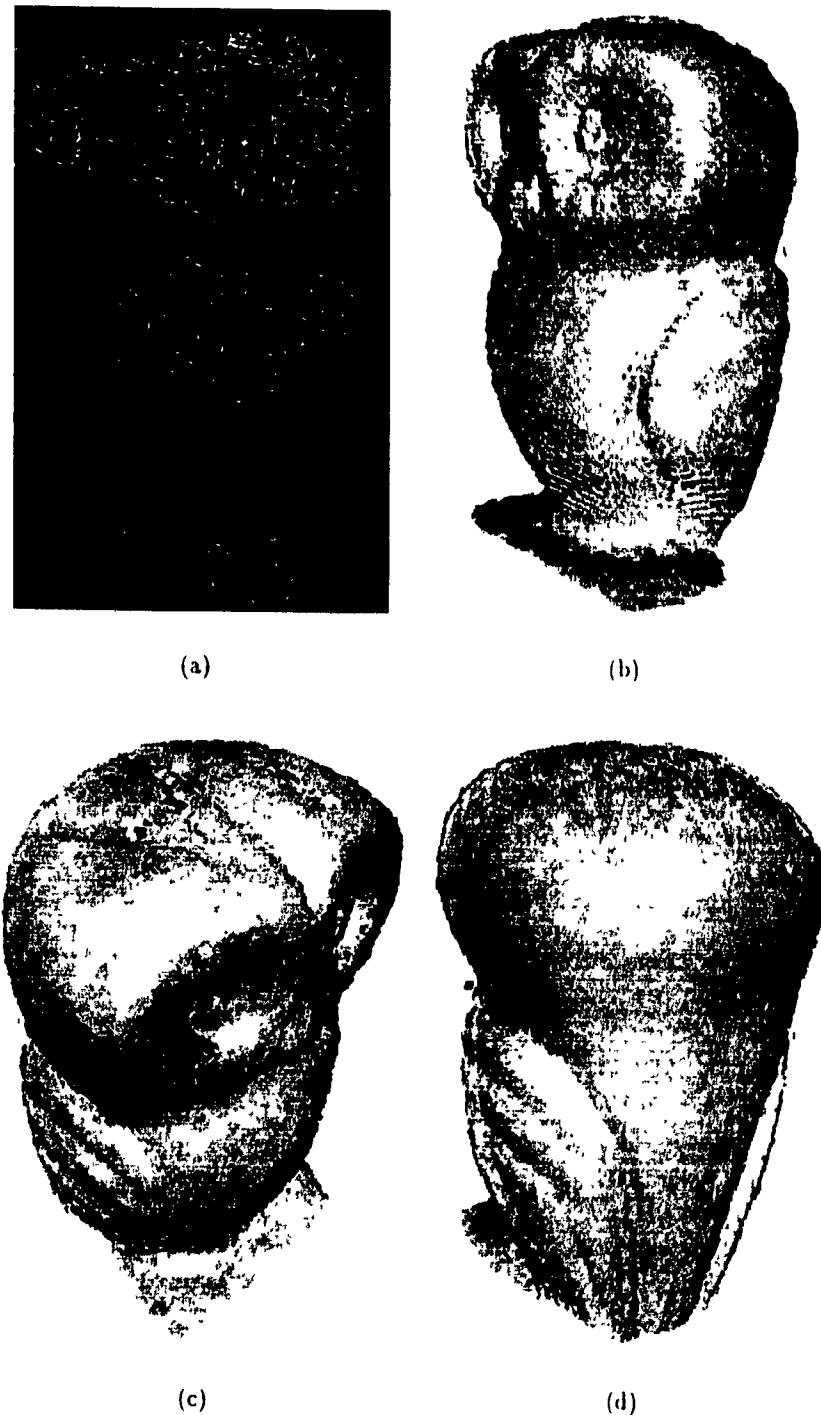


Figure 5.38: After the six range views of the owl have been registered, redundant data is removed and a 3D model is constructed. The program used to remove redundant data and generate the 3D model was provided by Kenong Wu. The model consists of a set of 3D points along with a normal specifying the orientation of the surface at that point. The surface of the model is rendered by fitting a small shaded plane at each point oriented along the normal for that point. Figure (a) shows the set of 3D points forming the model; the background was set to black in order to ease the visualization of the points. Figures (b), (c) and (d) show different viewpoints of the model.

5. Experiments and Results

Dual-view registration experiments were conducted with VFSR for range views of various 3D objects. The registration algorithm performed satisfactorily in all cases. The approach used for the registration of multiple range views was described. It was shown that multi-view registration performed by locally optimizing the cost function between pairs of range views could result in relatively large registration errors between the first and last image. To circumvent this problem, a global cost function is defined. The cost function consists of the sum of the cost of individual transformations between pairs of views. A multi-view experiment was performed with the global optimization approach and good results were obtained.

Chapter 6

Conclusions

A novel approach for the registration of range images has been presented in this thesis. The method relies on formulating the registration task as an optimization problem by defining a cost function measuring the quality of registration between two range views. To measure the registration quality provided by a given rigid 3D transformation, the cost function evaluates a sum Euclidean distances between control points in one view after transformation and corresponding points in the other view. The point correspondence between range views is established by inverting the set of calibration equations of the rangefinder. This inverse calibration provides us with a very fast way of determining point correspondences across range views.

In Chapter 4, two optimization techniques were described: Genetic Algorithm (GA) and Very Fast Simulated Reannealing (VFSR). 3D slices of the 6D cost function were plotted for the registration of a typical set of range images. In most plots the cost function exhibited evidence of multimodal behavior, showing one significant global minimum surrounded by a large number of minor local minima. Because of the presence of these multiple local minima, the use of a gradient descent optimization was dismissed. The optimization techniques (GA and VFSR) were chosen for their stochastic properties, which makes them inherently robust for finding a global optimum in the presence of multiple local optima.

As was shown in Chapter 5, the registration results obtained with Very Fast Simulated Reannealing were superior to those of the Genetic Algorithm. The quality of the registration obtained was better and the time required to converge to the minimum cost solution was shorter. As a consequence, further registration experiments were conducted with the use of VFSR. A series of dual-view registration experiments were first presented. The registration algorithm performed well in all cases, even when the optimization was initiated far from the optimum solution. The registration of the pipe, car and fruit views showed that the algorithm was capable of registering range views acquired with a eye-in-hand system where the initial transformation estimate provided by the robot arm is sometimes quite inaccurate. The registration of the teapot range views also showed the robustness of the algorithm to self-occlusion.

A multi-view registration experiment was presented in the last section of Chapter 5. It was observed that when range views were registered in pairs, an accumulated error in each intermediate transformation would result in a large registration error between the first and last range views. When the range views were simultaneously registered with the redefined cost function, this error was eliminated. The main problem with this approach is that with the simultaneous registration of many range views, the convergence time increases exponentially with the number of views registered. Because of this, the approach used for multi-view registration is not practical for applications that require results in a short time period. Nevertheless, the algorithm displayed a remarkable performance during dual-view registration experiments. It could therefore be a useful tool in areas such as the ones mentionned in the introduction (object recognition, robot navigation, CAD input system and computer graphics). To make the multi-view registration useful for practical applications, it is recommended that future research be conducted to find a parallel version of the approach which would then combine the speed of dual-view registration with the registration quality obtained with the simultaneous registration. A surface model constructed from the six registered views of the owl figurine showed the feasability of creating a higher level model from the set of 3D points of all range views obtained after registration.

Future improvements of the algorithm and prospective research in the domain of range image registration can consist of the following:

1. The data acquisition process could be automated by computing the next viewing position directly from the range data according to some attentional criterion (see Whaite [32, 33]).
2. Instead of the normal annealing schedule of VFSR, a better termination condition should be found for the registration. Here are some suggestions:
 - Terminate search once a specified percentage of points is within a certain distance threshold (for example, stop when 90% of control points are within 0.5 mm from the other range view.).
 - Terminate search when the value of the cost function is below a certain desired value. For example, one could make use of the known distribution of the rangefinder sampling error and the average scanning distance in order to determine when to stop the optimization search. The search could be stopped when

the value of the cost function is less than the mean absolute sampling error plus some tolerance factor.

3. Look at extracting control points based on their strategic importance instead of using a regular sampling interval. This intelligent selection of control points could result in better registration results and a faster convergence if less control points were necessary to attain the same (or better) registration quality.
4. Study the use of higher level surface information such as normal, principal curvature direction and curvature when comparing the difference between corresponding points in two views. This might yield better results than simply computing the Euclidean distance between points. The cost function could be modified to take into account these structural differences between corresponding points [27, 34].
5. When doing multiple view registration, it might be a good idea to integrate views gradually into a model and then register the new range views directly with this model. Merging range views and eliminating redundant data in order to create a surface model reduces the amount of information one has to carry and might speed up the registration process. To register a new range view with a partially complete model, one could simply minimize the distance between the new range view and a set of control points selected on the surface of the model.
6. Study a parallel implementation of the algorithm. When registering multiple views, the evaluation of the cost function requires measuring the distance between two views at a time. Thus the evaluation could be divided such that separate processes were responsible for evaluating the cost function for separate pairs of range views.

The research presented in this thesis is yet another attempt at solving the complex problem of range image registration. It is hoped that the various ideas that have been put forward will contribute to a better understanding of the registration problem and will pave the way to a practical system for 3D modeling.

Bibliography

- [1] Colin Archibald and Shahram Amid. Calibration of a wrist-mounted range profile scanner. In *Vision Interface '89*, London, Ontario, June 1989. Canadian Image Processing and Pattern Recognition Society.
- [2] Homer H. Chen. Pose determination from line-to-plane correspondences: Existence condition and closed-form solutions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):530–541, June 1991.
- [3] Jin-Chang Cheng and Hon-Son Don. A graph matching approach to 3-d point correspondences. *International Journal of Pattern Recognition and Artificial Intelligence*, 5(3):399–412, 1991.
- [4] Yang Cheng and Gérard Medioni. Object modeling by registration of multiple range images. *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, 1991.
- [5] C. H. Chien and J. K. Aggarwal. Volume/surface octrees for the representation of three-dimensional objects. *Computer Vision, Graphics and Image Processing*, 36:100–113, 1986.
- [6] David E. Goldberg. *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley Pub. Co., 1989.
- [7] Bruce Hajek. A tutorial survey of theory and applications of simulated annealing. In *Proceedings of 24th Conference on Decision and Control*, Ft. Lauderdale, Florida. December 1985.
- [8] Thomas C. Henderson. Efficient 3-d object representation for industrial vision systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(6):609–618, November 1983.
- [9] A. Hill and C. J. Taylor. Model-based image interpretation using genetic algorithms. *Image and Vision Computing*, 10(5):295–300, June 1992.

- [10] D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J. Physiol.*, (160):1561–1573, 1962.
- [11] D. H. Hubel and T. N. Wiesel. Functional architecture of macaque monkey visual cortex. *Proceedings Royal Society London*, 198 B:1–59, 1977.
- [12] Lester Ingber. Very fast simulated reannealing (vfsr). *Mathematical and Computer Modeling*, 12(8):967–973, 1989.
- [13] Lester Ingber and Bruce E. Rosen. Genetic algorithms and very fast simulated reannealing: A comparison. *Mathematical and Computer Modeling*, 16(11):87–100, 1992.
- [14] S. Kirkpatrick, C. D. Gelatt, and Jr. M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598), May 1983.
- [15] F. Blais M. Rioux and J.-A. Beraldin. Laser range finder development for 3-d vision. In *Vision Interface '89*, London, Ontario, June 1989. Canadian Image Processing and Pattern Recognition Society.
- [16] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *J. Chem. Phys.*, 21(6), 1953.
- [17] Pierre Parent and Steven W. Zucker. Trace inference, curvature consistency, and curve detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(8), August 1989.
- [18] Michael Potmesil. Generating models of solid objects by matching 3d surface segments. *8th International Joint Conference on Artificial Intelligence*, 2, 1983.
- [19] Michael Potmesil. Generating octree models of 3d objects from their silhouettes in a sequence of images. *Computer Vision, Graphics and Image Processing*, 40, 1987.
- [20] Gerhard Roth and Martin D. Levine. A genetic algorithm for primitive extraction. *Proceedings of the Fourth International Conference on Genetic Algorithms*, July 1991. (San Diego).
- [21] Gerhard Roth and Martin D. Levine. Geometric primitive extraction using a genetic algorithm. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 1992. Champaign, Illinois: IEEE Computer Society Press.

- [22] Y. Roth-Tabak and R. Jain. Building an environment model using depth information. Technical Report CSE-TR-07-88, University of Michigan, September 1988.
- [23] Bikash Sabata and J.K. Aggarwal. Estimation of motion from a pair of range images: A review. *Computer Vision, Graphics and Image Processing*, 54(3):309–324, November 1991.
- [24] Yasuo Sakaguchi, Hirokazu Kato, Kosuke Sato, and Seiji Inokuchi. Generation of 3-d models based on image fusion of range data. *IAPR Workshop on Machine Vision Applications*, pages 147–150, November 1990.
- [25] Yasuo Sakaguchi, Hirokazu Kato, Kosuke Sato, and Seiji Inokuchi. Acquisition of entire surface data based on fusion of range data. *IEEE Transactions, E* 74(10):3417–3422, October 1991.
- [26] Mubarak A. Shah and Ramesh Jain. Detecting time-varying corners. *Computer Vision, Graphics and Image Processing*, 28:345–355, 1984.
- [27] Gilbert Soucy. View correspondence using curvature and motion consistency. Master's thesis, McGill University, 1992.
- [28] Sanjay K. Srivastava and Narendra Ahuja. An algorithm for generating octrees from object silhouettes in perspective views. *IEEE Computer Society Workshop on Computer Vision*, pages 363–365, 1987.
- [29] Richard Szeliski. Estimating motion from sparse range data without correspondence. *IEEE 2nd International Conference on Computer Vision*, pages 207–216, December 1988.
- [30] B. C. Vemuri and J. K. Aggarwal. 3-d model construction from multiple views using range and intensity data. *Proceedings CVPR / IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 435–437, 1986.
- [31] K. L. Watkin, L. H. Baer, S. Mathur, R. Jones, S. Hakim, I. Diouf, B. Nuwayhid, and S. Khalife. Three-dimensional reconstruction and enhancement of freely acquired 2d medical ultrasonic images. *Proceedings of the 1993 Canadian Conference on Electrical and Computer Engineering*, 1993. to be published.

6 Conclusions

- [32] P. Whaite and F.P. Ferrie. From uncertainty to visual exploration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-13(10):1038-1050, October 1991.
- [33] P. Whaite and F.P. Ferrie. Uncertain views. In *CVPR92*, Champaign, Illinois, June 15-18 1992. IEEE-CS, IEEE-CSP, to appear.
- [34] Kenong Wu. Parametric geons. Technical Report 93-13, McGill University, Research Center for Intelligent Machines, August 1993.