Registering Multiview Range Data to Create 3D Computer Objects

Gérard Blais Martin D. Levine

TR-CIM-93-16 November 8,1993

Centre for Intelligent Machines
McGill University
Montréal, Québec, Canada

Postal Address: 3480 University Street, Montréal, Québec, Canada H3A 2A7
Telephone: (514) 398-7115 FAX: (514) 398-7348
Electronic Mail: gblais@cim.mcgill.ca levine@cim.mcgill.ca

Registering Multiview Range Data to Create 3D Computer Objects

Gérard Blais Martin D. Levine

Abstract

This research deals with the problem of range image registration for the purpose of building surface models of three-dimensional objects. The registration task involves finding the translation and rotation parameters which properly align overlapping views of the object so as to reconstruct from these *partial* surfaces, an *integrated* 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 between the partial surfaces contained in two range images as produced by a set of motion parameters . This function computes a sum of Euclidean distances between a set of control points on one of the surfaces to corresponding points on the other. The strength of this approach resides in the method used to determine point correspondences across range images. It is based on reversing the rangefinder calibration process, 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.

A stochastic optimization technique, Very Fast Simulated Reannealing (VFSR), is used to minimize the cost function.

Dual-view registration experiments yielded excellent results in very reasonable computational time. A multiview registration experiment was also performed, but 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.

Contents

List of Figures

1. Introduction

In many practical applications it is desirable to generate three-dimensional models of real objects. Possible areas include object recognition, robot navigation, CAD input and computer graphics. For object recognition, a 3D modeling system could be used to create a 3D representation from multiple views of an object. This could be then be compared to a database of object models to identify it. For robot navigation, one may require a mobile robot to move about and map its environment. A model of the surroundings could be used by the robot for path planning and collision avoidance. In manufacturing, it is often desirable to input detailed shape specifications for an already existing part into a CAD program. A 3D modeling system could be used for this task as well. In certain computer graphics applications, the system described in this paper could be used for the creation of artificial scenes and objects representing real world objects (i.e., virtual reality).

To obtain surface information about a scene, the human visual system makes use of two eyes separated by a small distance and computes the depth(range) from the discrepancies between the images in both eyes. This is the principle behind stereo vision. It is a very complex process which is an active subject of research but still not yet completely understood. Besides stereo vision, humans also use certain optical information, such as shadows, to determine the shape of objects. This has spawned various areas of research collectively referred to as *shape-from-X* techniques. These attempt 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 modeling an object, is to obtain the surface information directly using a *rangefinder*.

A rangefinder image consists of a two-dimensional array of 3D points representing the surface of the 3D object. In this paper, the approach we took to modeling an object was to obtain multiple range images from various viewing positions (active vision) and then to piece them together to form the model. The registration of the range images was accomplished using a rigid 3D transformation (only translation and rotation was permitted). Because of the multiple range views, we need to determine many such transformations and express all views in a *unique* coordinate frame such that they are perfectly registered.

More formally, given N views of an object in a scene, each one describing the 3D structure of the object as seen from a particular viewpoint, we wish to find N rigid motion transformations $T_1, T_2, \ldots, T_N$, 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 $i (i = 1, \ldots, N)$ 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 $S_i' = T(S_i)$ 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 is the union of all transformed sets $S_1', S_2', ..., S_N'$, 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.

A novel approach for solving this problem of range image registration is presented in this paper. It is a relatively simple method and combines both (i)speed of execution and (ii)robustness to noise in the range data and positioning errors in the mechanical apparatus used for data acquisition. For example, the method is suitable for use with an eye-in-hand system, where the rangefinder is attached to the end of a robot arm, which is generally known to be a rather sloppy absolute positioning device.

To achieve accurate registration, a cost function is defined which indicates the quality of registration of two range views by a sum of distances between corresponding points in each view. These range views are registered by determining the 3D rigid transformation which minimizes the cost function. The novelty of this approach is a method for reversing the calibration process of the rangefinder which permits point correspondences between range views to be computed directly. This results in an extremely fast method for computing the distance between range views, as required by the evaluation of the cost function. Stochastic search is used to find the transformation which minimizes the cost function in a reliable manner, even in the presence of the multiple local minima present in the cost function.

Section 2 describes the different approaches which have been developed for solving the range image registration problem. In Section 3, the essential aspects of our registration method are presented and registration is formulated as an optimization problem. Section 4 discusses the testing of the method by performing several registration experiments. Finally, in Section 5, various aspects of the registration approach are discussed and possible improvement are suggested.

2. Previous Work

The methods used for range image registration can be divided into two main categories. The first avoids the registration problem altogether by relying on precisely calibrated mechanical 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. They can be viewed as *open loop* systems, where the registration transformation provided by the

acquisition apparatus is accepted blindly without verification. Some of these open loop methods are described in Section 2.1.

The second category involves methods that derive the registration transformation between range images from the information contained in the range images and other information provided by the acquisition system. In distinction to the first category, these can be viewed as *closed loop* systems where the transformation parameters are gradually updated and refined until the range views are precisely 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 is only a coarse approximation to 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 above, many researchers have circumvented the problem of searching for the appropriate motion transformation to register two range images by simply relying on accurate and precise hardware. For example, Sakaguchi et al. [10, 11] have utilized this technique to generate octree models of 3D objects. Both a precision turntable and an eye-in-hand system were used for data acquisition. The authors do not discuss how the registration of views obtained with the movable eye-in-hand system were performed. However, similar to the views acquired with the turntable, it is very likely that they employed the positional estimate of the robot arm in order to determine the transformation between range views.

Vemuri and Aggarwal [16] also relied on a calibrated system to obtain inter-frame transformations. They made use of a turntable (base plane) on which the object to be modeled was placed. A pattern was drawn on the base plane which permitted the inter-frame transformation to be deduced by observing the orientation of this pattern in the intensity image.

Similar to Sakaguchi et al., Potmesil [7] has used an octree representation for creating 3D models from *intensity* images. But instead of combining range views, conic volumes, generated from silhouettes of the object, were merged. To merge the different conic volumes, Potmesil used a camera calibration technique to find the position of the sensor with respect to a fixed frame of reference. Srivasta and Ahuja [14] have employed a similar method and tested their algorithm with artificial 3D data, for which they could directly derive the views from any desired position.

Again employing artificial data, Roth-Tabak and Jain [8] simulated a system in which a sensor was moved around its environment in order to build an internal representation of its surrounding world. Because they knew the exact position and orientation of the sensor with respect to the global coordinate system of the world, they could register all sensor views into a

common frame of reference.

In many cases, the inaccuracy of open loop systems is acceptable, but there are situations where more precise registration is required. This is true even when using turntables. In order to improve on model accuracy, closed loop systems must be considered.

2.2 Closed Loop Registration Techniques

Registration methods can make use of information contained in the views to be registered to derive the appropriate motion transformations relating them. Two possibilities exist. The first consists of methods that compare the differences in the *structure* of the surface across views. This comparison is performed over the entire surface or over a set of control points on the surface. The second consists of methods which register range views by matching features from one view with those from the other.

The approach taken by Potmesil [6] is to compare surface differences between range views and to find the rigid 3D transformation which minimizes those differences. The different views of the object were acquired so as to guarantee partial overlap. The matching algorithm used heuristic search to align overlapping surface segments in a common 3D coordinate system. A cost function was defined to measure the quality of the registration. It was assumed to be it/unimodal and measured orientation and shape differences between a set of control points in the first view and corresponding points in the second. Correspondence across views was established by a ray-casting procedure. The matching algorithm minimized these differences over all of the control points, thereby maximizing shape similarity.

Cheng and Medioni [4] employed an iterative method to register two range images. An initial estimate of the transformation was assumed to be available. Range views were registered by minimizing the distance from control points in one view to planes in the other. A line-surface intersection algorithm was used to determine the point in the second view which was intersected by a line originating from a control point in the first view and in the direction of the normal at that point. This is similar to the way Potmesil [6] established correspondences across range views. Once the intersection point was found, the equation of the plane, tangent to the surface at that point, was determined. The distance measure was defined as the Euclidean distance between the control point in view 1 and the corresponding tangent plane in view 2.

Our method is similar to that of Cheng and Medioni's [4] . The main difference is their use of a line-surface intersection search algorithm to determine point correspondence between range views. In our approach, correspondence is established directly through an inverse calibration function. Thus, the main advantage of our approach is that it eliminates this search by it/directly computing the location of the most likely correspondence point. This results in an extremely fast metric for the evaluation of inter-surface distance. This metric is then the basis for a cost

function which is optimized in order to register range views.

Szeliski [15] 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 come from the same surface. The major disadvantage of this approach is that the range data from two 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 described by Soucy [13] compares surface structures across range views to determine the best registration transformation. The latter may be nonrigid, meaning that the surfaces are allowed to deform in order to be properly registered. To register two range views, a curvature consistency algorithm was first applied to each image, so that the local curvature field varied smoothly over the whole image and sharp transitions are smoothed. The first view was then divided into a set of rectangular patches and a few (5 to 10) of these were selected for registration. Each of the selected patches was then fit to the surface of the second view. The basis of Soucy's approach is the minimization of a functional that measures the difference between a local neighborhood in one image and a corresponding one in an adjacent image [13]. Soucy demonstrated qualitatively that the functional became convex after a sufficient number of iterations of the curvature consistency algorithm. Then, assuming unimodality of the functional, a simple gradient descent algorithm was used to find the best motion parameters for each patch. Once all the selected surface patches were been fit to the second view, their motion parameters were propagated to their neighboring patches in the first view. The propagated parameters were adjusted for each patch so that the it touched the surface of the second view and the relative rotation minimized the difference metric. Then these patches propagated their motion parameters to their respective neighbors and so on, until all patches had motion parameters assigned to them.

The last stage imposed motion consistency. The relative position and orientations of the patches were adjusted so that they reflected as closely as possible the relative position and orientations that they had in the original view before the transformation. The positions and orientations of these patches were iteratively updated until the desired level of rigidity in the transformation of the original surface structure was obtained.

This aspect of the method is advantageous in the sense that, by controlling the amount of rigidity in the transformation, one is able to partially compensate for possible distortions introduced by the rangefinder during the acquisition of the range images. The disadvantage of this approach is the excessive amount of computing time.

The approaches 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. For example, Cheng and Don [3] invoked the principle of invariance of distance measures under rigid body motion. They chose a triangle of points $\triangle P_1P_2P_3$ from the first view and then found the best matching triangle in the other. There are other methods which are partially related to the registration problem. For example, the one proposed by Shah and Jain [12] solved the correspondence problem in 2D by matching corners across image frames. Chen [2] has devised a technique for determining the pose of an object in a scene based on a known model. The sensory data were lines and the reference model was in the form of planes. To determine the pose of the object, a closed-form solution was found for a set of line-to-plane correspondences. So here again features are matched in order to determine a motion transformation. However, the problem with these techniques is that they rely on accurate feature extraction prior to registration and this process is very prone to error.

Finally, the reader might wish to consult Sabata and Aggarwal [9] for a review of the problem of estimating motion from a pair of range images. The method discussed in this paper has the advantages of the surface matching algorithms of Potmesil [6] or Cheng and Medioni [4], in the sense that no feature extraction step is necessary. However, the point correspondence problem required by these techniques is eliminated by using inverse camera equations to directly compute the position of corresponding points across range views.

3. Registration using Inverse Camera Calibration

3.1 Registration Method Overview

The rangefinder used for this research provides a two-dimensional sweep of the surface of a scene. A range image consists of a variable size rectangular array of depth values of points sampled on the surface of the scene. In this research, we have used a maximum size 256 by 256 range images. 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 the point, its coordinates (x, y, z) 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 indices i and j would specify which ray sampled each point. Figure 1 shows a surface sampled by the rangefinder.

The principal idea behind our method is to reverse this process whereby 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') in 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

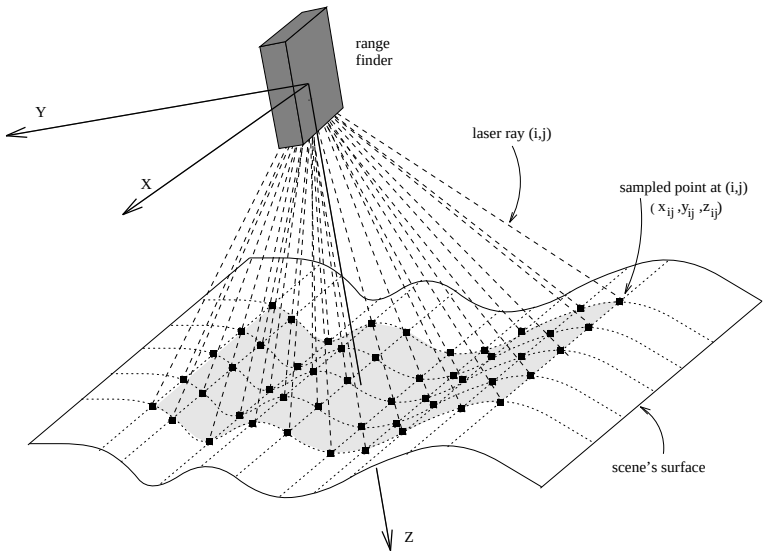


Figure 1: This figures shows how the rangefinder gathers information about the surfaces in a scene. A surface is sampled in a grid-like fashion and the z values obtained are stored in a two-dimensional array. The indices of the array represent the particular ray in the 2D scanning grid which sampled the given point. The value z_{ij} would thus be stored in location (i, j) of the range image array.

closest (Euclidean distance) to that transformed 3D point. Since every point in a range image is obtained by sampling the surface with a different ray, the point in image 2 associated with the (i, j) ray is thus taken as the corresponding point.

To perform the registration, control points(pixels) are selected from the first range view by uniform subsampling. These are mapped by a rigid 3D transformation T into the second view's reference frame. Each transformed control point is then associated with 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, is computed. The objective is to find that transformation T which minimizes this distance measure. A transformation estimate T_e, obtained from the acquisition apparatus, is used to constrain the number of possible transformations. Thus, a finite search space is delimited around the estimate T_e and only those transformations inside this search space are considered as potential solutions for the registration. The inverse calibration process is detailed in Section 3.3.

By minimizing the sum of Euclidean distances between all control points in one view and their respective corresponding points in the other, the distance between these views is minimized. Since the sum of distances is 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 3.2.

Note that for 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 corresponding indices 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 performing 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, correspondence being established using the inverse calibration.

3.2 Formulating the Registration Task as an Optimization Problem

The objective of the registration task is to find a transformation T which best represents the relative displacement and orientation between two range views. An optimization algorithm is used to search for the best transformation. Therefore, we must define some kind of measure by which transformations can be ranked based on the quality of registrations they produce.

We will now formally define the cost function needed for optimization. Let S_c be a set of control points taken from the total set of points in the first view. S_c is a subset of all sampled points in that view. Let T be the transformation which takes a point in the first view and expresses it in the reference coordinate frame of the second. If $\vec{\mathbf{p}}$ is a point in the first view, then $T(\vec{\mathbf{p}})$ is the same point expressed in the second view's coordinate frame. We specify a rigid 3D transformation by six motion parameters, consisting of three translations $t_x, t_y and t_z$, and three rotation angles r_x, r_y and r_z. From these motion parameters, we define a translation vector $\vec{\mathbf{d}}$ and a rotation matrix $\mathbf{R}$ as follows:

$$\vec{\mathbf{d}} = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} \tag{1}$$

$$\mathbf{R} = \begin{pmatrix} c_y c_z & -c_y s_z & s_y \\ c_x s_z + s_x s_y c_z & c_x c_z - s_x s_y s_z & -s_x c_y \\ s_x s_z - c_x s_y c_z & s_x c_z + c_x s_y s_z & c_x c_y \end{pmatrix} \tag{2}$$

where

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

With these definitions, $T(\vec{\mathbf{p}})$ is simply

$$T(\vec{\mathbf{p}}) = \mathbf{R}\vec{\mathbf{p}} + \vec{\mathbf{d}} \tag{3}$$

Let $C()$ be the correspondence function defined by the camera's inverse calibration equations defined in Section 3.3. If $\vec{\mathbf{q}}$ is a point in the first view's coordinate frame, then $C(\vec{\mathbf{q}})$ is the point in the second view whose associated ray is closest to point $\vec{\mathbf{q}}$. 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 in the second range image are found using the inverse calibration equations. The result is the 3D point in 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. This would be the case if the indices i and j represented a datum point that has been discarded during the preprocessing of the range images. Also, because a range image has a maximum of 256 by 256 sampled points, it is possible that the values of i and j computed from the inverse calibration equations are outside the allowed range of the indices, which must be between 1 and 256. 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:

$$cost(T) = \sum_{\vec{\mathbf{p}} \in S_c} d\left(T(\vec{\mathbf{p}}), C(T(\vec{\mathbf{p}}))\right) \tag{4}$$

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 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.

When the function $C()$ finds no corresponding point for a given control point in S_c and returns *undefined*, the L_2 norm is undefined. Therefore, the cost function given by Equation (4) 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.

As shown in Figure 2, 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.

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

$$d(\vec{p_1}, \vec{p_2}) = \begin{cases} \|\vec{p_1} - \vec{p_2}\| & ; \text{if } \vec{p_2} \text{ is defined and } \|\vec{p_1} - \vec{p_2}\| \leq \tau \\ \tau & ; \text{if } \vec{p_2} \text{ is defined and } \|\vec{p_1} - \vec{p_2}\| > \tau \\ 0 & ; \text{if } \vec{p_2} \text{ is undefined} \end{cases} \tag{5}$$

Because the distance measure returns 0 when no correspondence point is found, a transformation minimizing the number of correspondences would yield a minimum cost value. However, this is undesirable 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:

$$cost(T) = \frac{\sum_{\vec{\mathbf{p}} \in S_c} d\left(T(\vec{\mathbf{p}}), C(T(\vec{\mathbf{p}}))\right)}{\|S_c(T)\|} \tag{6}$$

There still exists a problem with the cost function defined in this way. Because no penalty is assigned to transformations yielding few correspondences, the cost function will nevertheless be a minimum when no correspondences are established between views. This issue can be handled by enforcing an overlap between 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 evaluated using the normal cost function defined by equation (6). 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.

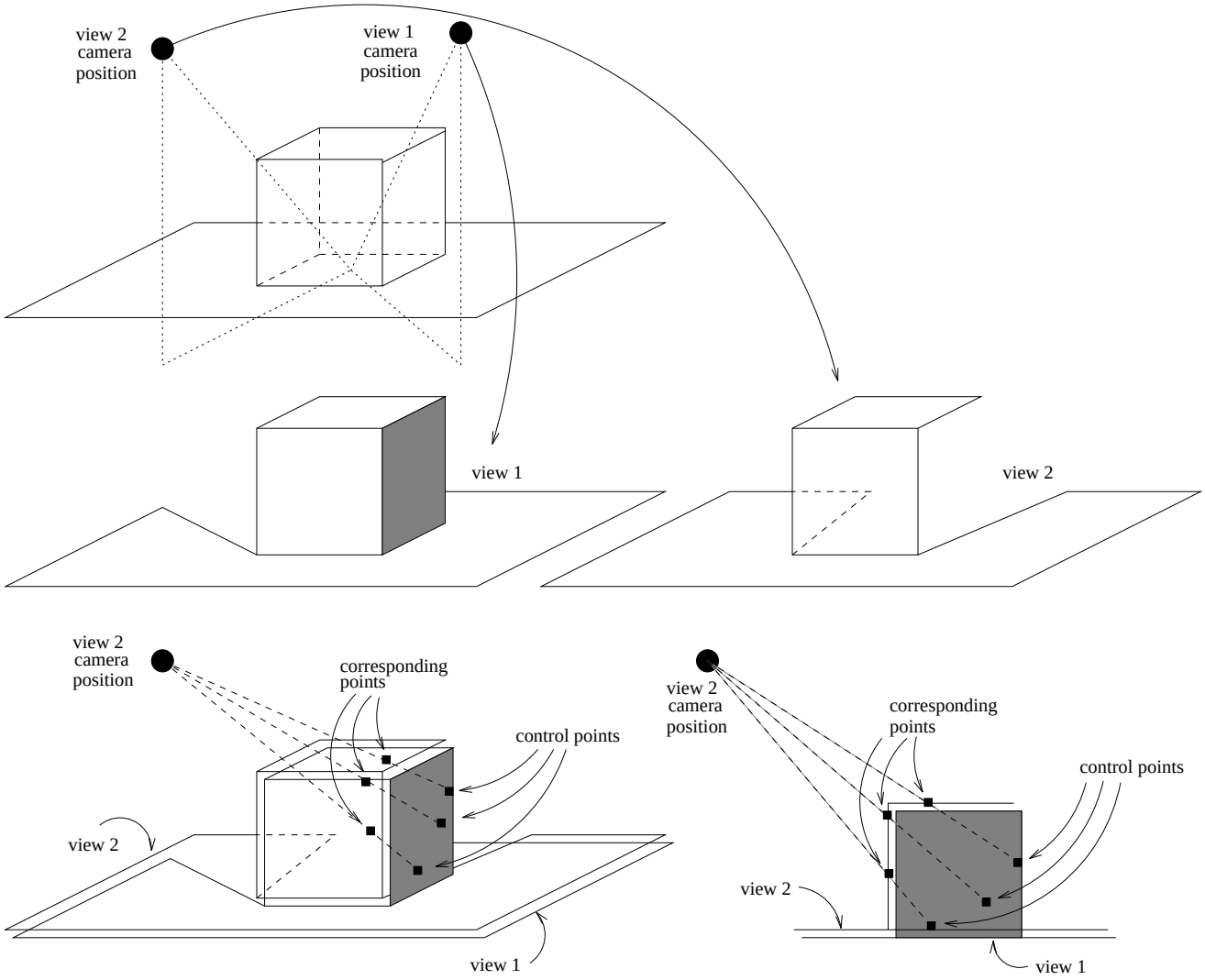


Figure 2: This figure illustrates the use for the distance threshold. Two range images are acquired from the same scene (a cube on a plane). The points sampled from the side of the cube (shown by a shaded region) in the first view are unique to view 1. View 2 is a straight-on view of the cube and thus does not contain any points sampled on the side of the cube. The cost function value 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 correspondences 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.

Let Ω be the overlap factor. For example, $\Omega = 0.3$ means that at least 30% overlap between range views is required . The overlap generated by a transformation T is simply the total number of correspondences $\|S_c(T)\|$ divided by the total number of control points $\|S_c\|$. Because of the threshold τ, and because control points without correspondences 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 in mind, we redefine the cost function as follows:

$$cost(T) = \begin{cases} \frac{\sum_{\vec{\mathbf{p}} \in S_c} d(T(\vec{\mathbf{p}}), C(T(\vec{\mathbf{p}})))}{\|S_c(T)\|} & ; \text{if } \frac{\|S_c(T)\|}{\|S_c\|} \geq \Omega \\ \tau \|S_c\| & ; \text{if } \frac{\|S_c(T)\|}{\|S_c\|} < \Omega \end{cases} \tag{7}$$

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

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

Note that by establishing point-to point correspondence across range images 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 is taken as

either the perpendicular distance or the distance to the closest point. One may argue that these give better indications of the distance between two views. However, as optimization progresses, and the registration between the two views improves, the distance along the scan lines will approach the perpendicular distance. Figure 3 illustrates how the point-to point correspondence is established between two range images.

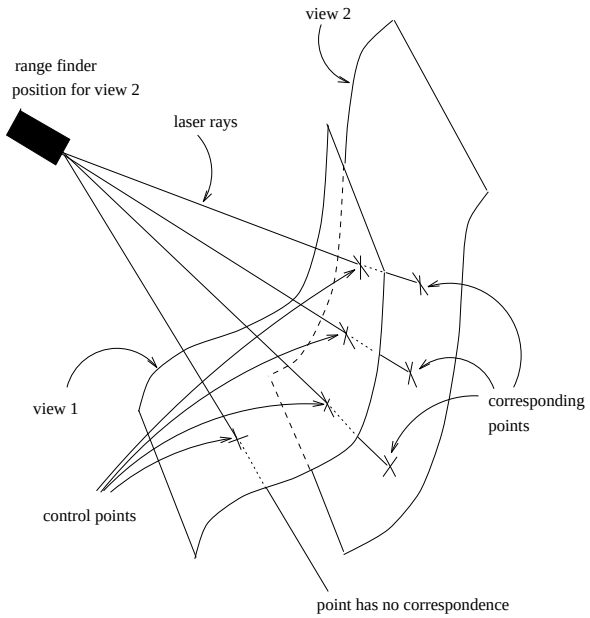


Figure 3: This figure shows how point-to-point correspondence is established across range images. After transformation, a control point in view 1 is put into correspondence with the point in view 2 having the closest ray.

Figure 4 shows the process involved in evaluating the fitness of a transformation. At the top , we see the acquisition of a typical range image where two images of a scene are obtained from two different viewing positions. The bottom shows how the registration quality of a transformation is evaluated in the optimization process. The first step consists of the selecting control points in the first range image. These are then expressed in the coordinate frame of the second range image. This is done by applying to all control points the motion transformation we wish to evaluate. Once the control points are mapped into view 2's reference frame, the corresponding points in the second range image are determined (through the inverse camera calibration equations). The cost function is then evaluated by summing the Euclidean distance between all control points and their corresponding points. Two transformation examples are shown. In the first, the transformation provides a good registration of the range images. In the second, the transformation is poor. Good registration yields small distances between control points and their correspondences and hence results in low cost.

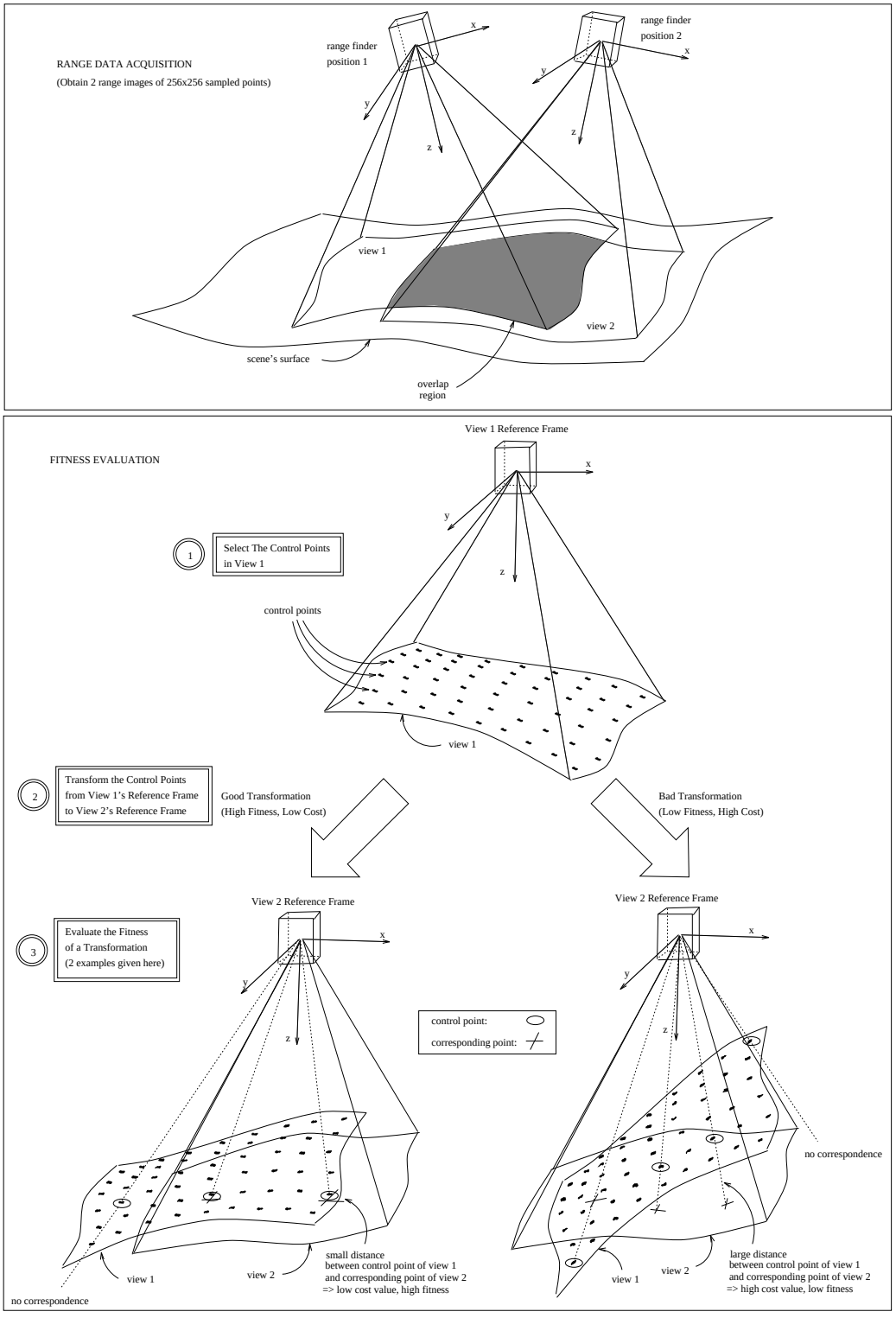


Figure 4: This figure illustrates the process of cost function evaluation. The top illustrates typical range data acquisition. The bottom shows how the cost function is evaluated for a transformation. Two cases are illustrated; a good (left) and a bad transformation (right).

3.3 Inversion of the Rangefinder Calibration Process

3.3.1 Coordinates Evaluation from Rangefinder Calibration

In order to determine the actual 3D coordinates (x, y, z) in space of a sampled point, the rangefinder must first be calibrated. This produces 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 two rotating mirrors (X-mirror and Y-mirror) used to sweep the laser beam over the surface.

Because the two-dimensional array of distances that constitutes the range image may contain a maximum of 256 by 256 sampled points, the indices can take values from 1 to 256.

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

1.
$$y = z \ TAN[\, j \,] \tag{8}$$

2.
$$\rho = \sqrt{y^2 + z^2} \tag{9}$$

3.
$$\alpha = ALPHA[\, i \,] \tag{10}$$

4.
$$x = \rho M[\, \alpha \,] + B[\, \alpha \,] \tag{11}$$

In the above equations, TAN, $ALPHA$, M and B are all calibration tables containing 256 entries. These are generated only once during the calibration of the rangefinder camera. Figure 5 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 (8), 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.

From Equation (9) and Figure 5 it can be seen that ρ is simply the length of the line segment generated by the ray sampling the point 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. However, sometimes the mirror skips and advances by more than one position or does not move at all. As a consequence, the i index value will not always

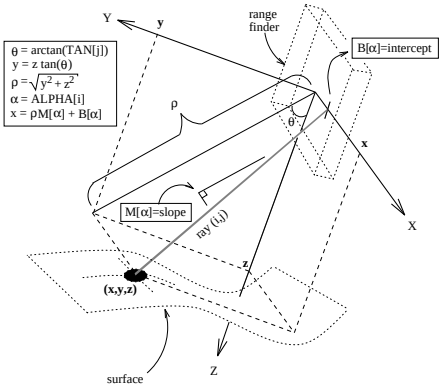


Figure 5: The geometric relationship between the 3D coordinates (x, y, z) and the different rangefinder parameters are shown in this figure. The coordinate reference frame is local to the rangefinder. The X-mirror is in its i^{th} discrete position and the Y-mirror in its j^{th} discrete position. The 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 the 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.

correspond to the actual position of the X-mirror. 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 (11) 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.

3.3.2 Reverse Calibration Process

By reversing the calibration process, we can compute the indices i and j from the coordinates (x, y, z) of a point in space. In other words, we wish 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 the range image. Therefore, if we can associate a ray with a 3D point, we can establish a point-to point correspondence between the given 3D point and the (i, j) point in the range image.

a) Computing the Y-mirror index j

The inversion of Equation (8) permits us to determine the value of j from the values of y and z. Given the coordinate values y and z we wish to determine the index value j such that Equation (8) is satisfied. 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 relationship emerges.

This can be exploited to simply express the table as the tangent of a linear equation:

$$TAN[j] = \tan(\theta_0 + \theta_1 j) \tag{12}$$

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 (8) and (12) we derive

$$j = \frac{\arctan(\frac{y}{z}) - \theta_0}{\theta_1} \tag{13}$$

In order to guarantee that the result obtained for j is always an integer, Equation (13) is rounded out to the closest integer value. We rewrite Equation (13) as follows

$$j = \left\lfloor \frac{\arctan(\frac{y}{z}) - \theta_0}{\theta_1} + \frac{1}{2} \right\rfloor \tag{14}$$

Equation (14) 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.

b) Computing the X-mirror index i

From Equation (10) and (11) we observe that in order to compute the index i, three calibration tables must be inverted. To invert Equation (10), the $ALPHA$ calibration table must be inverted; and to invert Equation (11), 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 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 to give:

$$ALPHA[i] = \alpha_0 + \alpha_1 i \tag{15}$$

Obtaining i from α is then achieved directly using the following equation

$$i = \frac{\alpha - \alpha_0}{\alpha_1} \tag{16}$$

b.2) Inverting the M and B Calibration Tables

We can now consider the problem of inverting Equation (11). 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 the Euclidean distance. The value of ρ is obtained directly from Equation (9).

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 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 (8), 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 value of z by that value:

$$y_{min} = \begin{cases} z_{max} \times TAN_{min} & \text{if } TAN_{min} < 0 \\ z_{min} \times TAN_{min} & \text{if } TAN_{min} \geq 0 \end{cases} \tag{17}$$

$$y_{max} = \begin{cases} z_{min} \times TAN_{max} & \text{if } TAN_{max} < 0 \\ z_{max} \times TAN_{max} & \text{if } TAN_{max} \geq 0 \end{cases} \tag{18}$$

Using Equation (9) and the min/max values of y and z as defined above, the min/max values of ρ are determined as follows:

$$\rho_{min} = \begin{cases} \sqrt{y_{max}^2 + z_{min}^2} & \text{if } y_{max} < 0 \\ \sqrt{0 + z_{min}^2} = z_{min} & \text{if } y_{min} < 0 \text{ and } y_{max} \geq 0 \\ \sqrt{y_{min}^2 + z_{min}^2} & \text{if } y_{min} \geq 0 \end{cases} \tag{19}$$

$$\rho_{max} = \begin{cases} \sqrt{y_{min}^2 + z_{min}^2} & \text{if } |y_{min}| > |y_{max}| \\ \sqrt{y_{max}^2 + z_{min}^2} & \text{if } |y_{min}| \leq |y_{max}| \end{cases} \tag{20}$$

Given ρ_{min} and ρ_{max}, we can then compute x_{min} and x_{max} using Equation (11) and the min/max values of the M and B calibration tables:

$$x_{min} = \begin{cases} \rho_{max} \times M_{min} + B_{min} & \text{if } M_{min} < 0 \\ \rho_{min} \times M_{min} + B_{min} & \text{if } M_{min} \geq 0 \end{cases} \tag{21}$$

$$x_{max} = \begin{cases} \rho_{min} \times M_{max} + B_{max} & \text{if } M_{max} < 0 \\ \rho_{max} \times M_{max} + B_{max} & \text{if } M_{max} \geq 0 \end{cases} \tag{22}$$

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, and hence, the more precise it 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} delimit 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:

$$x_d = \left\lfloor \frac{(x - x_{min})(x_{dim} - 1)}{x_{max} - x_{min}} + \frac{1}{2} \right\rfloor \tag{23}$$

$$\rho_d = \left\lfloor \frac{(\rho - \rho_{min})(\rho_{dim} - 1)}{\rho_{max} - \rho_{min}} + \frac{1}{2} \right\rfloor \tag{24}$$

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

$$x = \frac{x_d}{(x_{max} - x_{min})}(x_{dim} - 1) + x_{min} \tag{25}$$

$$\rho = \frac{\rho_d}{(\rho_{max} - \rho_{min})}(\rho_{dim} - 1) + \rho_{min} \tag{26}$$

To construct the table, the closest ray, one out of a possible 256 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 6 illustrates the relationship between the values in the $INVERSE\_MB$ table and the 256 rays defined by the slope 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 6 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

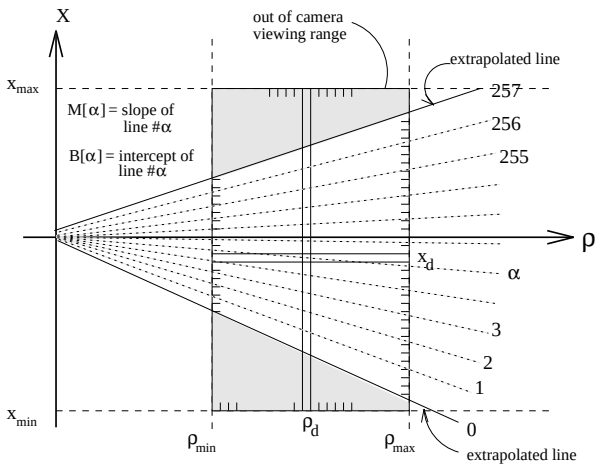


Figure 6: This figure shows the 256 rays defined by the slope and intercept values contained in the M and B calibration tables. The region delimited by x_{min}, x_{max}, ρ_{min} and ρ_{max} defines the extent of the $INVERSE\_MB$ table. Each entry in the table corresponds to the α value of the line closest 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. The table entries in these areas are set to 0 to reflect this.

ray is close enough to account for it. The other entries are set to an integer 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 additional 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 third order polynomials derived for M and B at 0 and 257.

Now, all the elements required to build the table are available. It 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 (25) and (26).

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 two 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.

3.3.3 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 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 by the laser rangefinder. Equations (8) to (11) describe how this 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 the range image array at location (i, j).

The reverse calibration process is done as follows:

1. Compute j from y and z using Equation (14).
2. Compute ρ using Equation (9).
3. Compute α from ρ and x by using the $INVERSE\_MB$ look-up table.
4. Compute i with Equation (10).

4. Experiments and Results

This section presents the experiments conducted in conjunction with this research. A brief description of the experimental setup and the data acquisition process is given in Section 4.1.

Considerable experimentation was carried out to determine the characteristics of the cost function For most range images tried, these usually indicated the presence of a single global optimum surrounded by multiple local optima. As a consequence, to ensure successful minimization for many different types of object surfaces, it became evident that we had to rely on a robust search method. A conventional gradient descent approach would be inadequate. Thus Very Fast Simulated Annealing (VFSR), a stochastic optimization method, was used to minimize the cost function [5]. Section 4.2 discusses the search parameters used for VFSR and the values of these parameters yielding optimum performance for registration.

Section 4.3 presents various dual-view registration experiments. Each consists of the registration of two range views obtained by sampling an object from two different viewing positions. For one object, the experiment is examined in detail, while only the final results are shown for the others. Finally, in Section 4.4, a multiview 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.

4.1 Experimental Setup and Data Acquisition

An eye-in-hand system was used for the acquisition of the range images. It consisted of a rangefinder camera attached to the end effector of a PUMA 560 robot arm. The robot is inverted and mounted on the ceiling to permit easier positioning of the camera for viewing objects at various angles.

An alternate method was a turntable. While maintaining the rangefinder camera in a fixed position, a precision turntable was used to accurately rotate objects so that sampling them from different viewpoints could be achieved. The position of the turntable can be specified as an absolute angular value in degrees.

We have determined by experimentation that the sampling error of the laser rangefinder is Gaussian distributed and that a linear relationship exists between the average sampling error (average of the absolute values) and the object distance[1]. In most experiments conducted for this research, data acquisition was performed with the rangefinder at a distance of around 40 centimeters from the object, sometimes more depending on the size and shape of the object. At this distance 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 an object plus some added noise. The latter is Gaussian distributed with a standard deviation proportional to the object distance. When registering two range views, we therefore expect the minimum average Euclidean distance between corresponding points in each view to be twice the mean absolute sampling error (the errors in each view get added). The cost function computes an approximation to this average distance. Therefore, when two views are properly registered, we would expect the minimum 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.

Before registration, it was necessary to preprocess each of the range images. This was done using simple segmentation to specifically remove background information. In addition spurious data points were eliminated.

Figure 7 shows a typical example of multiple view range data acquired using the turntable. Nine range views of an owl figurine acquired at 40 degree intervals are shown after preprocessing.

4.2 Search Control Parameters

The parameters which control the registration process can be divided into two categories. The first set affects the size of the search space and the general shape of the cost function. The second actually controls the VFSR minimization. These are briefly discussed below.

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. These estimates are obtained

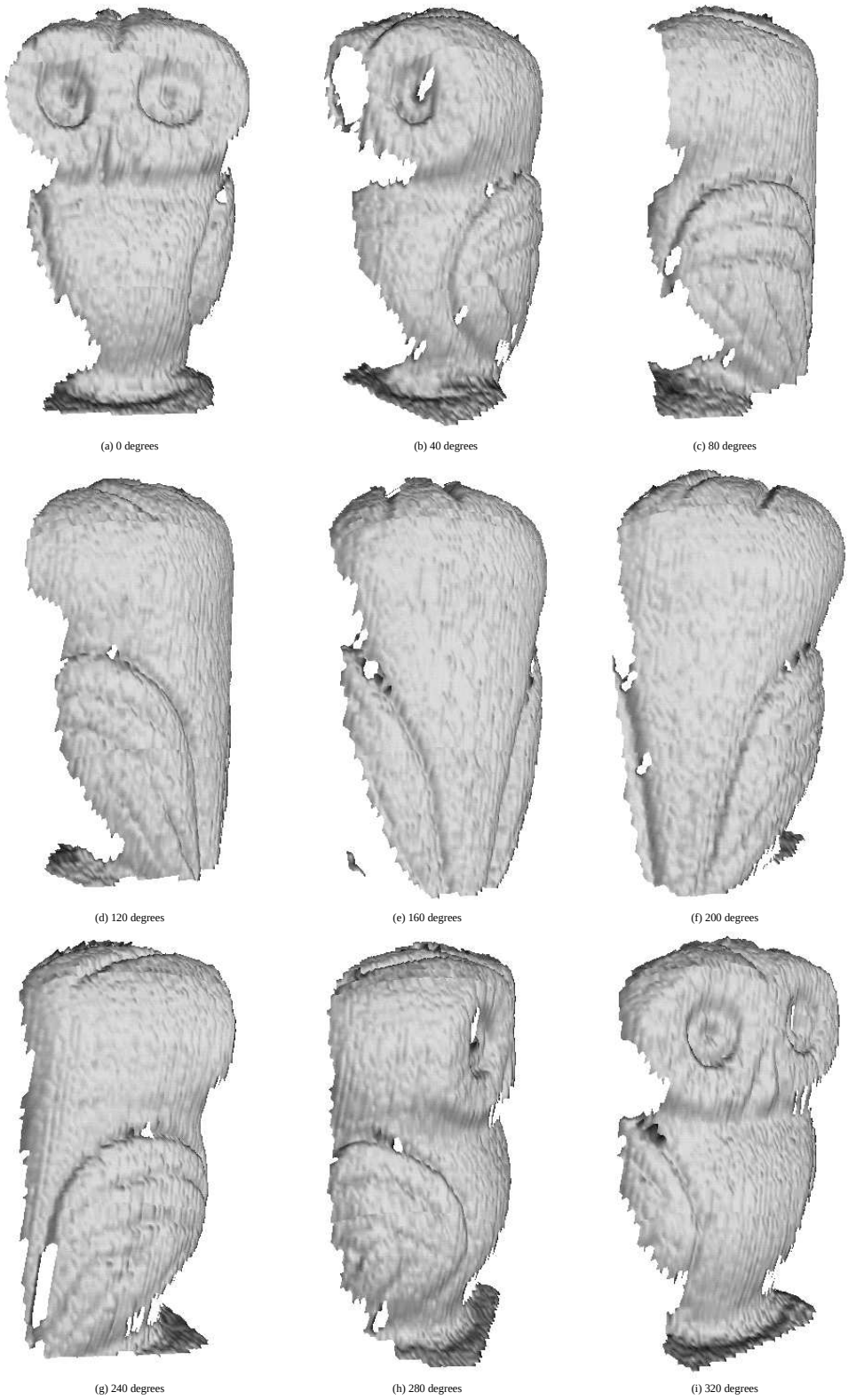


Figure 7: This figure shows nine range images of a small owl figurine seen from nine different viewpoints. The views were obtained by rotating the object on a precision turntable by 40 degree intervals. The views shown above were preprocessed in order to remove spurious data points and background surfaces, such as the platform of the turntable on which the object was resting, so that only the owl's surface remained.

from the positioning devices used for data acquisition (precision turntable or robot arm system) which are calibrated prior to acquisition. 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 bounds are set by specifying a search range $\Delta_{tx}, \Delta_{ty}, \Delta_{tz}, \Delta_{rx}, \Delta_{ry} and \Delta_{rz}$ around the estimate of each motion parameter. Using the transformation estimate, the lower and upper bounds for each parameter is computed by respectively, subtracting and adding the Δ range from the estimated parameter.

Appropriate settings of the search range were determined by experimentation with various range views obtained with the turntable and the robot arm. 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 search performance are related to the evaluation of the cost function. These were determined experimentally as follows (see Section 3.2 and [1]):

(i) control point sample interval (=5)
(ii) overlap factor (=30%)
(iii) distance threshold (=5 millimeters)

The experiments were done with different objects in order to obtain a representative idea of parameter sensitivity.

The second set of parameters controls the VFSR algorithm. The most important of these 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 Dr. Lester Ingber* and was adapted to the registration task discussed in this paper[5]. In this computer program, more parameters are provided to control search than the ones actually mentioned above. However, the ones cited are the primary parameters for controlling the annealing process.

It was determined through extensive and well controlled experiments that setting T_0, TRS and TAS to 1.0, 0.001 and 10000, respectively, yielded excellent registration results [1]. Search convergence using these parameters was relatively insensitive to the type of range views being registered . The experiments were performed for two very different objects(the owl and metal pipe) and the same settings were found to be optimum in both cases.

*Dr. Lester Ingber, Lester Ingber Research, P.O. Box 857, McLean, VA 22101.

4.3 Dual-View Registration Experiments

This section presents various experiments realized with the registration algorithm described previously. All the experiments discussed here consist of the registration of two range views obtained from different objects.

Two views were taken of a white owl figurine having a slightly textured surface. Both were obtained with the precision turntable. The rangefinder camera was held in a fixed position while the object was scanned, rotated clockwise by 40^{o}, and scanned again. The two views are shown in Figures 7 (a) and (b).

Because the turntable is very precise, the transformation estimate it provides is always very close to the optimal registration transformation. In order to perform a more rigorous test of the registration algorithm, the motion parameters of the estimate were altered when the turntable was used for data acquisition. A perturbation of 8 millimeters and 4 degrees was added to each of the translation and rotation parameters, respectively. The resulting transformation was used as the initial estimate for the registration. This was also done for the experiments with the teapot seen in Figures 10 (a) and (b).

The setting of the various search parameters used for the owl figurine and the results obtained for the registration are shown in Table 1. These and all subsequent experiments were done on a Silicon Graphics Iris 4D workstation. Figure 8 (a) shows the two views using the initial transformation estimate; figure 8 (b) shows the two views registered with the optimum transformation found by VFSR. In order to differentiate between the two views, one of them is rendered as a grid and the other as a shaded surface. An interweaving of the grid and shaded surfaces indicates high quality registration. This can be easily seen in figure 8 (b) where the mesh of the first view seems to be partially immersed in the shading of the second.

Another indication of registration quality is a display of the Euclidean distance measure between the two range views. As shown in Figure 9, the distance is represented by various shades of gray. A dark shaded point indicates that the distance between this point and its corresponding point in the other view is small. A light shade indicates a large distance. The minimum distance, 0 millimeters, is indicated by black and the maximum, the distance threshold, is indicated by light gray. Points without correspondence and those whose distance to their corresponding point in the other view is greater than the distance threshold are shown with the lightest gray. A gray level scale is provided (Figure 9 (e)) to indicate the distance between each point in one view and the other. The initial and the final distance between the views is shown. As an example, figures 9 (a) and (b) show the distance between the two views when the initial estimate is used and figures 9 (c) and (d) show the distance when the views are registered using the transformation found by the algorithm. The dark regions clearly indicate where the two views overlap. We can observe that the overlap region of each view is dark, indicating excellent registration.

The value of the cost function is a good indication of registration quality. Its value for the initial transformation estimate is 4.30 mm and the optimum found was 1.05 mm. This is lower than the expected cost of 1.25 mm due to the rangefinder sampling error (see Section 4.1) and is a good indication that the fit is near optimal.

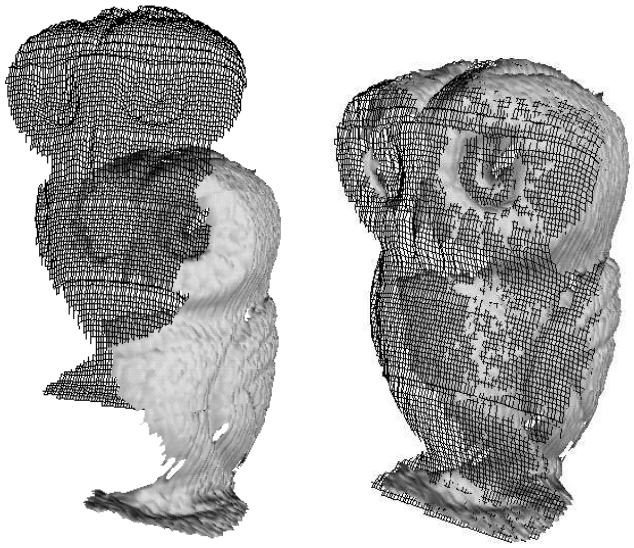
Other dual-view registration results obtained for various 3D objects are shown in Figure 10. The range views of the metal pipe, the fruits and the model car (Figure 10 (c) (d), (e) (f) and (g) (h), respectively) were acquired with the eye-in-hand robot system. This indicates the ability of the registration algorithm to handle the usually large errors occurring in the initial transformation estimate obtained with such a positioning system.

Registration Parameters Used For The Owl Figurine						
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 Parameters	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 perturbation)	-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					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					1.05 mm	
Number of Cost Function Evaluations					27044	
Registration Time (Note: varies with system load during experiment)					5 min. 57 sec.	

Table 1: Parameters used in the registration of owl range views and results obtained.

4.4 Multiview Registration

The registration of two range views is the first step in the integration of a set of range views. The goal of multiview registration is to construct a representation of the *complete* surface of an



(a) (b)

Figure 8: Figure (a) shows the relationship 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.

object by registering multiple range views. The process consists of combining the various range views of an object into a unique coordinate frame. Once this is accomplished, we can use the result for generating models and other higher level tasks, such as object recognition and robot grasping.

The most straightforward way of performing the registration of multiple views of a 3D object is to register the 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 on, until view 5 overlaps view 0, thus completing a circuit around the object. With these six range views, we could 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 (27).

$$\begin{aligned} T_{50} = T_{05}^{-1} &= (T_{45} \circ T_{34} \circ T_{23} \circ T_{12} \circ T_{01})^{-1} \\ &= T_{01}^{-1} \circ T_{12}^{-1} \circ T_{23}^{-1} \circ T_{34}^{-1} \circ T_{45}^{-1} \end{aligned} \tag{27}$$

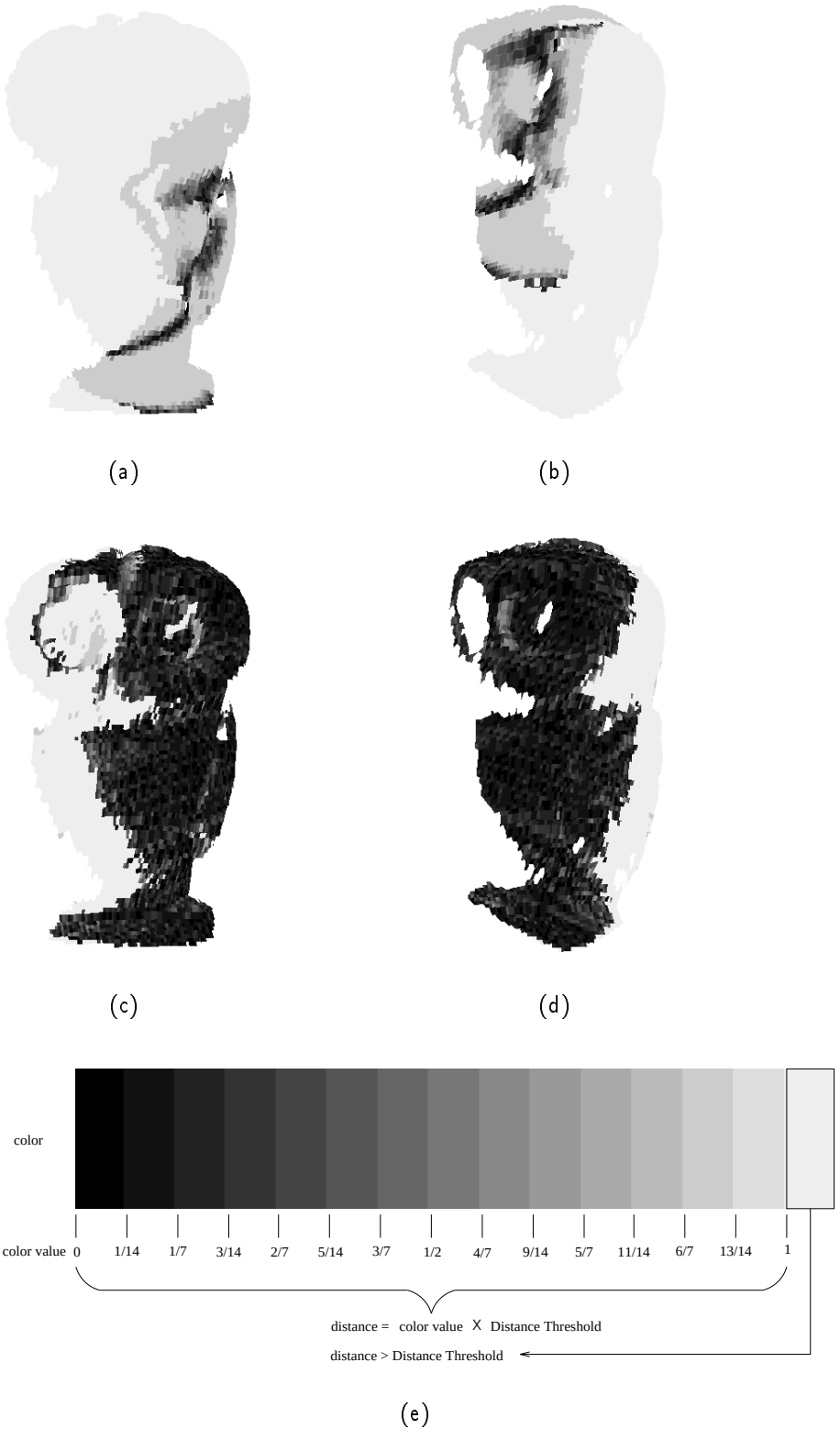


Figure 9:Figures (a) and (b) show the distance between the views when the initial transformation estimate is used. Figures (c) and (d) show the distance between the views when registered by the optimum transformation found by the algorithm. Figure (e) presents the scale for interpreting the gray levels in the above images. Figures (c) and (d) are very dark indicating excellent registration.

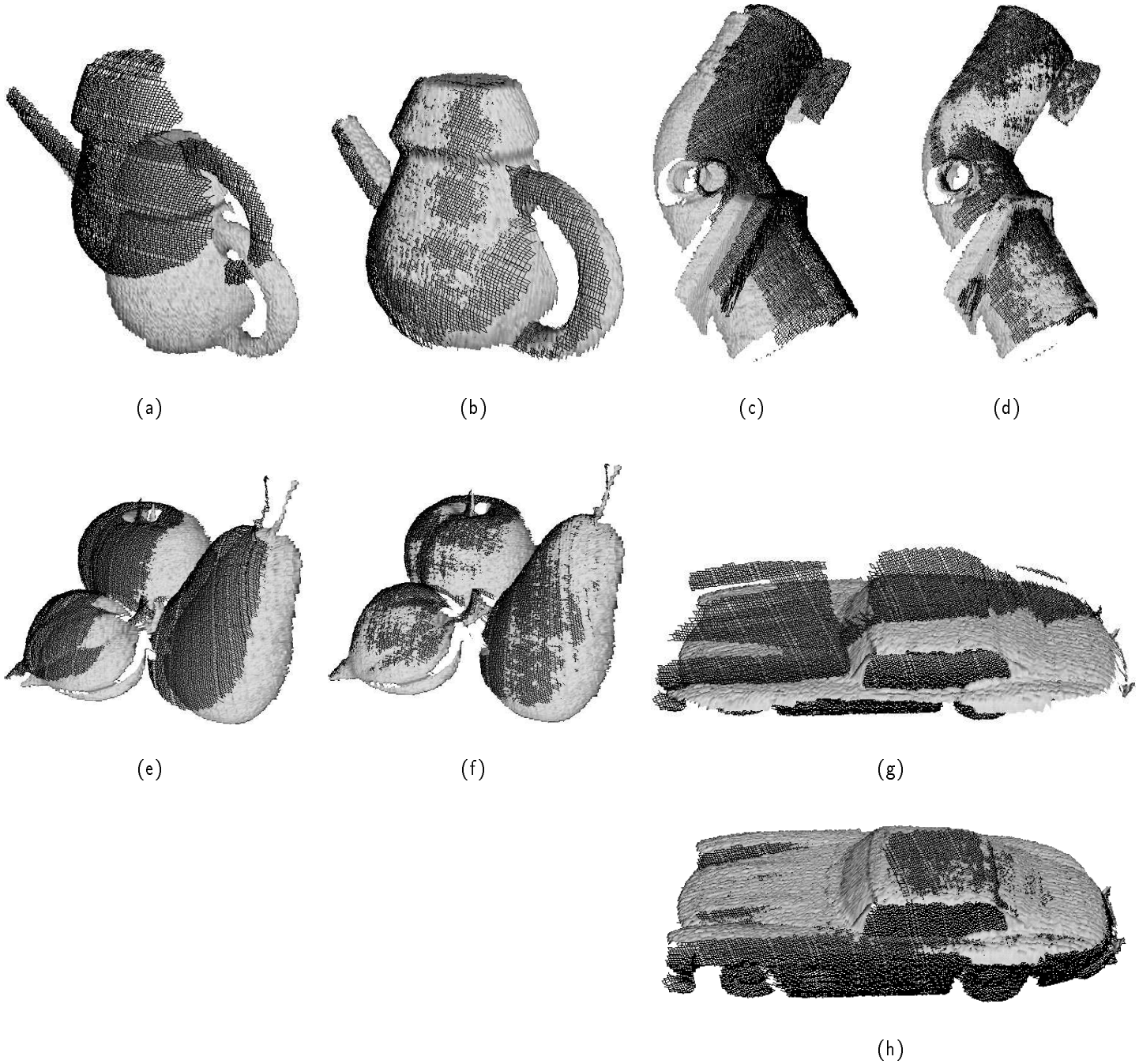


Figure 10: Composite image of four different dual-view registration experiments. Figures (a) and (b) show two views of a teapot before (related by the transformation estimate) and after registration (related by the optimum transformation found by the algorithm), respectively. Similarly, figures (c) and (d), (e) and (f), and (g) and (h) show the registration of two views of a metal pipe, fruits, and a model car, respectively.

where $T_a \circ T_b$ is the composite of transformations T_a and T_b.

The main problem with this approach is that, even though the five transformations found between pairs of views might be optimal, global registration will not be. A typical problem 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 may be well registered by T_{01}, but a very small error still exists. The same is true of all the other transformations. This small error in each intermediate transformation is compounded, so that by the time the last view wraps around to the first, a large error has accumulated.

To avoid this problem, the views can be registered simultaneously and the error between the first and last views can be taken into proper consideration. Of course, this makes more sense from a theoretical point of view as well. The experiment conducted here uses this approach. A new cost function is defined for the global registration of multiple range views as the sum of the individual costs of all *direct* 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. It is completely defined by a set of direct transformations. For example, transformation T_{50} is an indirect transformation and is completely defined by five direct transformations as shown in Equation (27). The new cost function to be minimized is thus

$$globalcost = cost(T_{01}) + cost(T_{12}) + cost(T_{23}) + cost(T_{34}) + cost(T_{45}) + cost(T_{50}) \quad (28)$$

where $cost(T)$ is the cost function defined in Equation (7). With this new cost function, a small error in all of the direct transformations is reflected as a large error in the indirect transformation. Thus global optimization ensures that all views fit together properly, not just in pairs.

A multiview registration experiment was performed for the owl figurine using a total of five range views. 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 2.

Figure 11 illustrates the problem arising when the range views are registered two at a time. In this case, view 0^o was registered with view 60^o, view 60^o with view 120^o , 120^o with 180^o, 180^o with 240^o, and finally 240^o with 300^o. Figure 11 (a) shows all the views together as seen from above. View 0^o is rendered as a grid surface and all the other views are rendered as shaded surfaces. This helps to distinguish views 0^o and 300^o from one another. We observe that all the views appear to be well registered except for the last (300^o) and first views (0^o). The large

Parameters Used For Multiview Registration Of The Owl						
Cost Function Parameters	Control point sample interval			5		
	Overlap factor			0.6		
	Distance Threshold			5.0		
VFSR Parameters	T_0			1.0		
	TRS			0.001		
	TAS			10000		
Search Space Parameters	t_x	t_y	t_z	r_x	r_y	r_z
T_e (from calibration)	-88.974091	-189.533615	152.127502	-49.844002	12.405737	-27.003699
Search Range	Δ_{tx}	Δ_{ty}	Δ_{tz}	Δ_{rx}	Δ_{ry}	Δ_{rz}
	2.0	2.0	2.0	1.0	1.0	1.0
Initial Cost					4.30 mm	
Results obtained using pairwise local registration						
Optimum Transformations	t_x	t_y	t_z	r_x	r_y	r_z
$T_{0^o-60^o}$	-80.19219	-190.2726	144.5746	-48.30938	10.68813	-26.0654
$T_{60^o-120^o}$	-87.16365	-188.0008	144.1832	-47.94806	11.73128	-27.74214
$T_{120^o-180^o}$	-87.70704	-189.8285	149.0761	-49.23435	11.84017	-28.28247
$T_{180^o-240^o}$	-84.26648	-187.5514	143.0579	-47.77332	11.47543	-26.6186
$T_{240^o-300^o}$	-88.60028	-187.3025	145.4442	-48.2851	12.15576	-27.7237
Which Local Registration		$T_{0^o-60^o}$	$T_{60^o-120^o}$	$T_{120^o-180^o}$	$T_{180^o-240^o}$	$T_{240^o-300^o}$
Optimum Cost		1.58 mm	1.49 mm	1.26 mm	1.14 mm	1.38 mm
Number of Cost Function Evaluations		10720	29383	33911	18457	25773
Registration Time		4m 56s	2m 24s	5m 33s	2m 56s	2m 21s
Results obtained using global						
Optimum Transformations	t_x	t_y	t_z	r_x	r_y	r_z
$T_{0^o-60^o}$	-90.71757	-189.848	152.031	-49.70864	12.76964	-27.34013
$T_{60^o-120^o}$	-90.84272	-191.3197	154.0251	-50.10959	12.79042	-26.89138
$T_{120^o-180^o}$	-90.41573	-189.4754	150.6458	-49.32046	12.42318	-27.7174
$T_{180^o-240^o}$	-89.9021	-190.2839	152.058	-49.82738	12.46874	-27.62393
$T_{240^o-300^o}$	-87.61896	-191.2735	152.6673	-49.89927	12.21386	-26.53895
Optimum Cost					1.55 mm	
Number of Cost Function Evaluations					10654822	
Registration Time					83 hr. 17 min. 53 sec.	

Table 2: Parameters used for global registration of the owl views and results obtained.

gap between these two views results from an accumulated error in all transformations between the first and last view. Figure 11 (b) is the same group of registered range views seen from the side. Figure 11 (c) is an enlargement of a section of figure (b) to more vividly demonstrate the large gap between the first and last view.

Figure 12 shows the result obtained when global optimization is performed to register the six range views. As can be seen from Table 2, the time required to perform this experiment is extremely large. This is due to the fact that the annealing time increases exponentially with the number of dimensions. When registering range views pairwise, the search space is six-dimensional (one for each motion parameter). When six range views are registered simultaneously, the search space has 30 dimensions (5 transformations of 6 parameters each). Because of this, the annealing time is very large. To reduce this computing time, the algorithm could actually be partially parallelized. The global cost function defined in Equation (28) 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, thereby significantly reducing the overall time.

Nevertheless, the results obtained with global registration are much superior to pairwise local registration. We can now see that the first and the last view are properly registered. As in Figure 11, the registered views are shown from above (a), from the side (b), as well as an enlargement of a section of figure (b). This time, there are no gaps between the first view (shown as a grid) and the last.

Figure 13 is a comparison of the registration quality obtained with pairwise local and global registration. The cost function was evaluated between each pair of range views. With pairwise registration, even though each local fit is good, we can clearly see the large registration error present between the 300^o and 0^o views of the owl, as indicated by the high cost function (3.18 mm).

Finally, the views were processed to remove any redundant data. Overlapping surfaces were eliminated by replacing them by their averages. The result obtained is a set of points representing the complete surface of the owl figurine. Different views of the model obtained are shown in Figure 14. The owl is rendered by fitting a small shaded plane at every point of its model. Figure 14 (a) shows all the points forming the model, each indicated by a small square plane. Figures 14 (b), (c) and (d) illustrates the model seen from various orientations.

5. Conclusions

This paper presents a novel approach for the registration of range images. The method relies on formulating the registration task as an optimization problem by defining a cost function which measures the quality of registration between two range views. To do this for a specific rigid

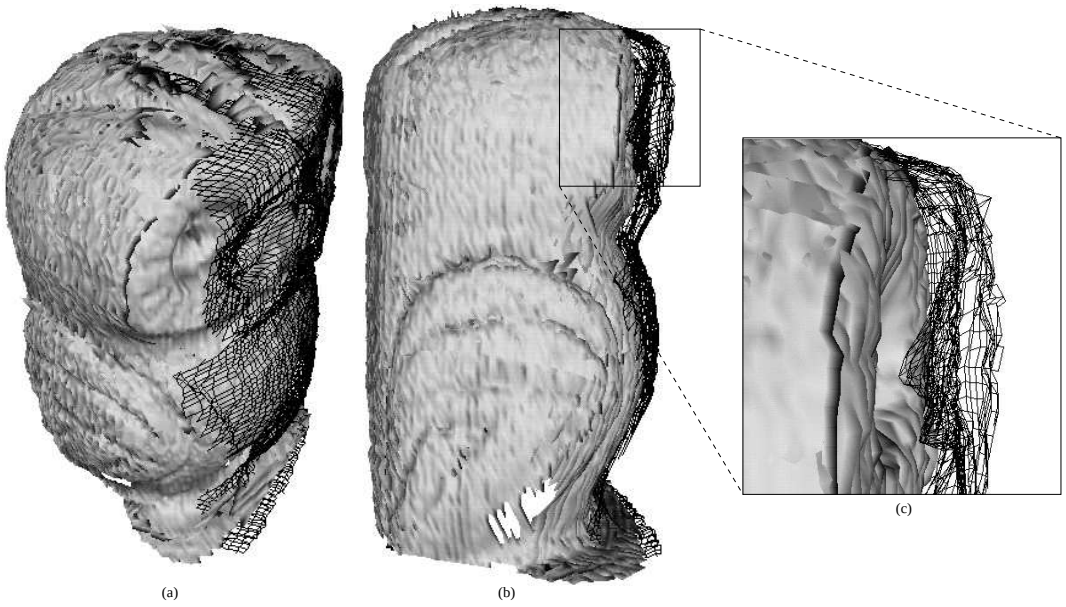


Figure 11: This figure illustrates the problem arising when registering multiple views without performing global optimization. Local optimization is performed to register range views two at a time. In figure (a) we observe that, although the fit between consecutive views is optimized, the first (shown 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), emphasizing the large distance between the first and last view.

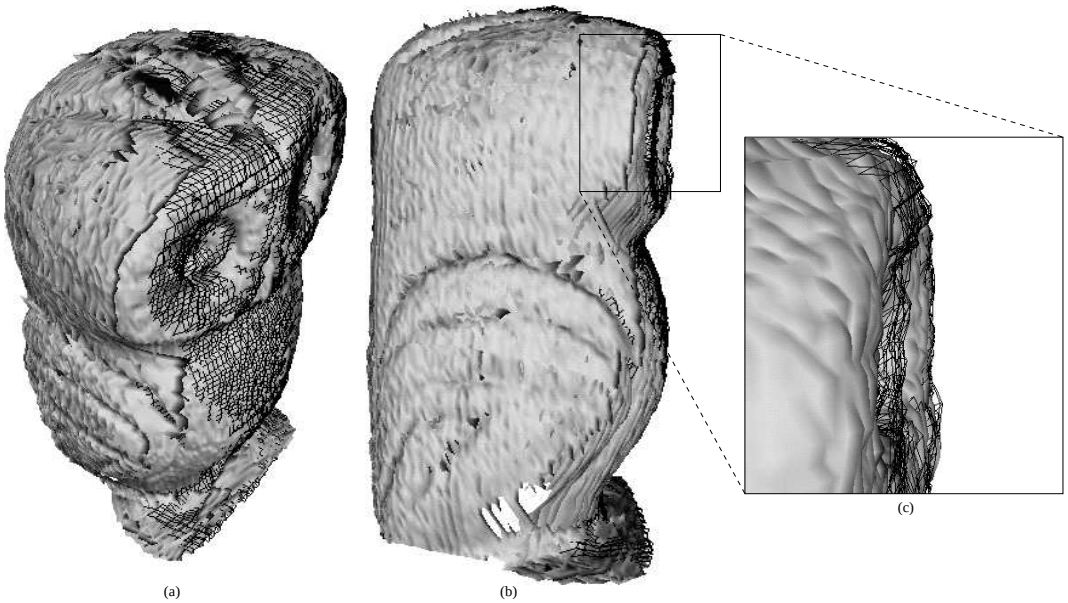


Figure 12: This figure shows the results obtained when registering multiple views using global optimization. Six range views were registered simultaneously. The result is shown seen from above (figure (a)) and from the side (figure (b)). Figure (c) is an enlargement of a section of figure (b) illustrating how closely the first (shown as a grid) and last view are registered.

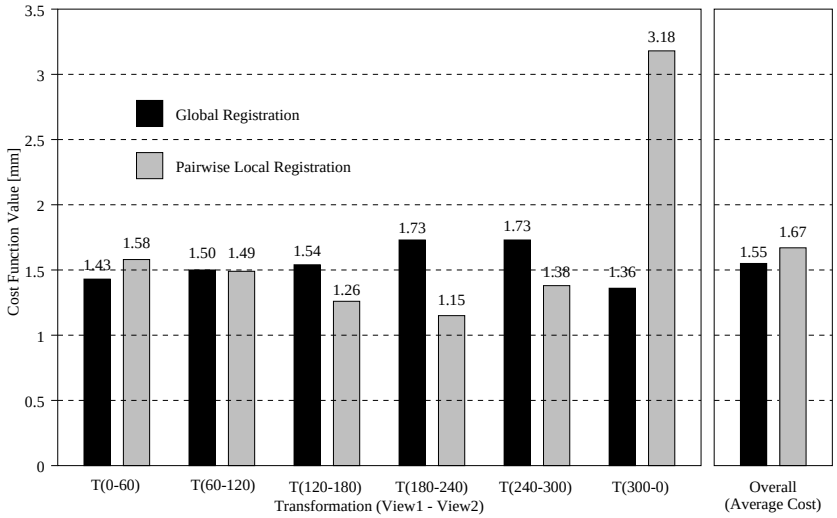


Figure 13: This figure compares the respective registration quality obtained with global (black bars) and pairwise local (gray bars) registration. The graph demonstrates that, even though registration quality is locally better when range views are registered in pairs, the overall fit is better when global registration is performed. We can see the large registration error between the first (0^o) and last (300^o) view of the owl, where the cost function evaluated between these two views yields a value of 3.18 mm. However, when global registration is performed, this error is eliminated and the overall cost function is lower (1.55 mm).

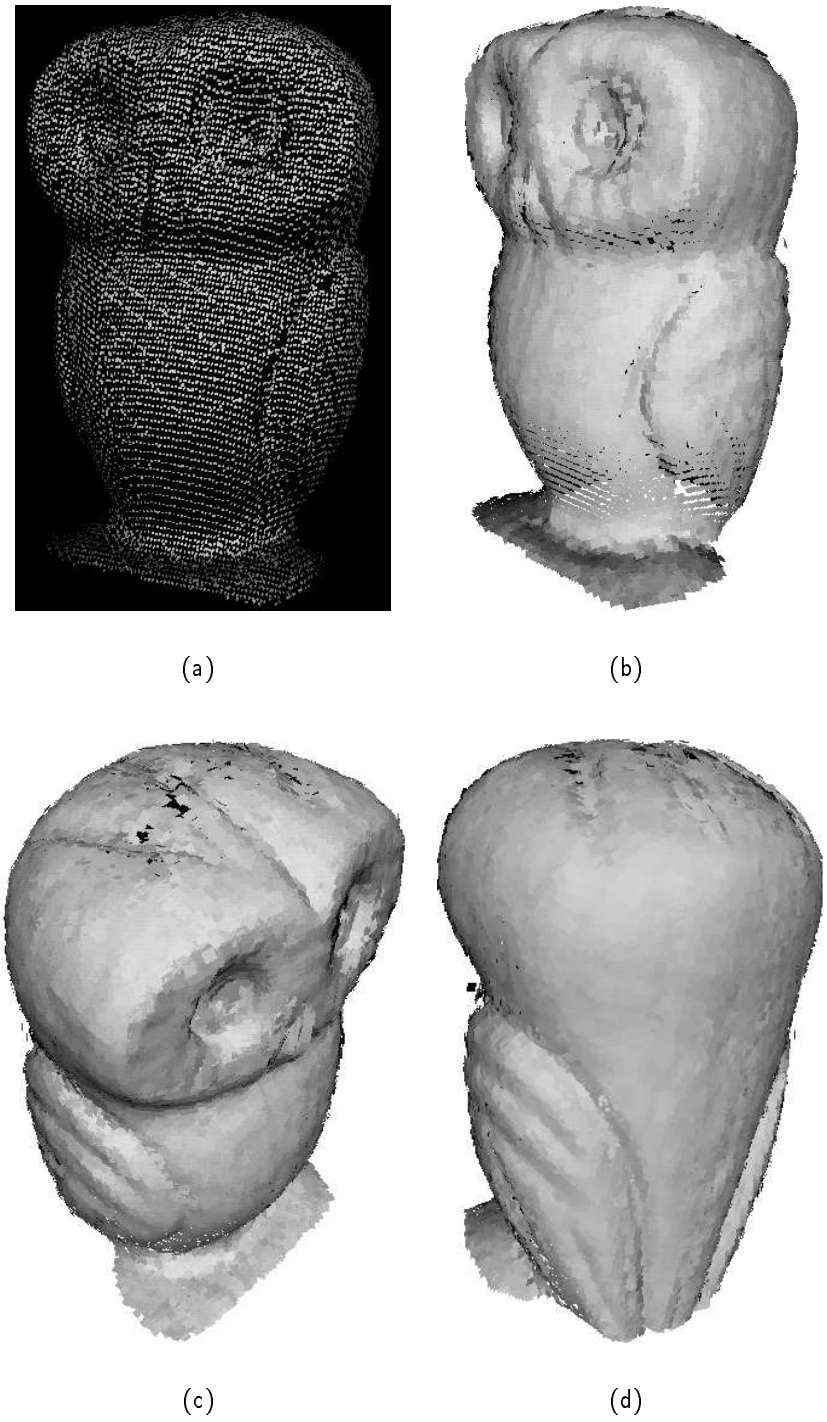


Figure 14: After the six range views of the owl have been registered, redundant data is removed and a 3D model is constructed. The model consists of a set of 3D points, along with a normal specifying the orientation of the surface at that point. The surface of the model is rendered by fitting a small shaded plane at each point, oriented along the normal at 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.

3D transformation, the cost function evaluates the sum of Euclidean distances between control points in one view after transformation and corresponding points in the other view. Point correspondence between range views is rapidly established by inverting the set of calibration equations of the rangefinder.

Using experimentation, it was determined that the cost function exhibited multimodal behavior, showing one significant global minimum surrounded by a large number of minor local minima. Because of these multiple local minima, the use of conventional gradient descent optimization was dismissed. The VFSR optimization technique was chosen for its stochastic properties, which makes it inherently robust for finding a global optimum in the presence of such multiple local optima.

A series of dual-view registration experiments was presented first. The registration algorithm performed well in all cases, even when optimization was initiated significantly far from the optimum solution. The registration of the metal 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 usually quite inaccurate. The registration of the teapot range views shows the robustness of the algorithm to self-occlusion.

A multi-view registration experiment was presented in the last section of Section 4. 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 a redefined cost function, this error was eliminated. However, the main problem with this approach is that the convergence time increases exponentially with the number of views registered. Because of this, multiview registration is not quite practical for applications that require results in a short time, although parallel processing could significantly reduce the computational time. Nevertheless, the algorithm displayed remarkable performance. A surface model constructed from six registered views of the owl figurine showed the feasibility of creating a complete model from the set of 3D points of all range views obtained after registration.

Future improvements of the algorithm could involve 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 [17, 18]).

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 search could be stopped when the value of the cost function was less than the mean absolute sampling error plus some tolerance factor.

3. It may be possible to extract the control points based on their strategic importance instead of using a regular sampling interval. This intelligent selection of control points could produce better registration results and faster convergence if less control points were necessary to attain the same (or better) registration quality.

4. When performing multiple view registration, it might be advantageous to integrate views gradually by registering the next range view directly with the already existing 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.

5. A study of a parallel implementation. When registering multiple views, the evaluation of the cost function requires measuring the distance between two views at a time. This could be divided so that separate computers were responsible for evaluating the cost function for separate pairs of range views.

The research presented in this paper is an attempt at solving the complex problem of multiview range image registration. It is hoped that this new approach will soon lead to a practical system for 3D modeling.

Acknowledgements

We thank Dr. Lester Ingber for kindly providing the VFSR (ASA) computer code as well as participating in valuable discussions. We would also like to thank Marc Bolduc, Samir Shah, Pierre Trmblay, Gilbert Soucy and Kenong Wu for their advice, expertise and technical assistance. G.B. would like to thank NSERC for its financial support in the form of an NSERC Postgraduate Scholarship. M.D.L. would like to thank CIAR and PRECARN for their financial support. This research was partially supported by the Natural Sciences and Engineering Research Council of Canada by a Strategic Research Grant.

References

[1] Gérard Blais. Creating 3d computer objects by integrating multiview range data. Master's thesis, McGill University, 1993.

[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] Lester Ingber. Very fast simulated reannealing (vfsr). *Mathematical and Computer Modeling*, 12(8):967–973, 1989.

[6] Michael Potmesil. Generating models of solid objects by matching 3d surface segements. *8th International Joint Conference on Artificial Intelligence*, 2, 1983.

[7] Michael Potmesil. Generating octree models of 3d objects from their silhouettes in a sequence of images. *Computer Vision, Graphics and Image Processing*, 40, 1987.

[8] Y. Roth-Tabak and R. Jain. Building an environment model using depth information. Technical Report CSE-TR-07-88, University of Michigan, September 1988.

[9] 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.

[10] Yasuo Sakaguchi, Hirokazu Kato, Kosuke Sato, and Seiji Inokuchi. Generatio of 3-d models based on image fusion of range data. *IAPR Workshop on Machine Vision Applications*, pages 147–150, November 1990.

[11] 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.

[12] Mubarak A. Shah and Ramesh Jain. Detecting time-varying corners. *Computer Vision, Graphics and Image Processing*, 28:345–355, 1984.

[13] Gilbert Soucy. View correspondence using curvature and motion consistency. Master's thesis, McGill University, 1992.

[14] 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.

[15] Richard Szeliski. Estimating motion from sparse range data without correspondence. *IEEE 2nd International Conference on Computer Vision*, pages 207–216, December 1988.

[16] 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.

[17] 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.

[18] P. Whaite and F.P. Ferrie. Uncertain views. In *CVPR92*, Champaign, Illinois, June 15-18 1992. IEEE-CS, IEEE-CSP. to appear.