Proceedings of the 1991 IEEE
International Conference on Robotics and Automation
Sacramento, California - April 1991

Object Modeling by Registration of Multiple Range Images*

Yang Chen † and *Gérard Medioni*

Institute for Robotics and Intelligent Systems
Departments of EE-Systems and CS
University of Southern California
Los Angeles, CA90089-0273
E-mail: yangchen@iris.usc.edu

Abstract

We study the problem of creating a complete model of a physical object. Although this may be possible using intensity images, we use here range images which directly provide access to three dimensional information. The first problem that we need to solve is to find the transformation between the different views. Previous approaches have either assumed this transformation to be known (which is extremely difficult for a *complete* model), or computed it with feature matching (which is not accurate enough for integration). In this paper, we propose a new approach which works on range data directly, and registers successive views with enough overlapping area to get an accurate transformation between views. This is performed by minimizing a functional which does not require point to point matches. We give the details of the registration method and modeling procedure, and illustrate them on real range images of complex objects.

1 Introduction

Creating models of physical objects is a necessary component machine of biological vision modules. Such models can then be used in object recognition, pose estimation or inspection tasks. If the object of interest has been precisely designed, then such a model exists in the form of a CAD model. In many applications, however, it is either not possible or not practical to have access to such CAD models, and we need to build models from the physical object. Some researchers bypass the problem by using a model which consists of multiple views ([4], [2]), but this is not always enough.

If one needs a complete model of an object, the following steps are necessary:

1. data acquisition,
2. registration between views,
3. integration of views.

By *view* we mean the 3D surface information of the object from specific point of view. While the integration process is very dependent on the representation scheme used, the precondition for performing integration consists of knowing the transformation between the data from different views. The goal of registration is to find such a transformation, which is also known as the *correspondence* problem. This problem has been at the core of many previous research efforts: Bhanu [2] developed an object modeling system for object recognition by rotating object through known angles to acquire multiple views. Chien *et al.* [3] and Ahuja and Veenstra [1] used orthogonal views to construct octree object models. With these methods, the correspondence problem is solved once the data acquisition facilities are calibrated. A similar method was also used by Wang and Aggarwal [13]. Vemuri and Aggarwal [12] have used a base-plane pattern to determine the interframe rotation of object by locating the pattern in the intensity images taken at the same time as the range images. These techniques all have difficulty in constructing a complete object surface description, since either the methods limit the movement of the objects relative to the sensors, or the sensors can only be at certain locations and we can not take advantage of the object surface structure in choosing vantage views. Ferrie and Levine [5] obtained interframe correspondence by matching surface features. The accuracy of this method depends on the accuracy of the feature detection technique used. Potmesil [9] developed a system for modeling complete surface of an object by taking multiple range data and then matching (i.e., registering) them through heuristic search in the transformation parameter space. Although his matching technique is quite general, we feel that searching through the huge parameter space, even with some heuristics, is neither computationally tractable nor necessary.

Our object modeling system attempts to build a complete model for an object through the integration of multiple-view range images. We use the information about the range finder setup and find the inter-frame transformation of the range images through range image registration. We avoid the search through the transformation parameter space by assuming an initial approximate transformation for the registration algorithm, since we believe that this information is available from the range finder setup (e.g. rotation on a rotary table) or through high level feature matching, e.g., [11], [8] and [4]. The registration algorithm is an iterative process minimizing a least square error measure. Unlike most other registration techniques, ours does not require point-to-point correspondence, since we minimize the distance from points to planes. Our integration procedure is currently performed by converting each view into either a spherical or a cylindrical coordinate.

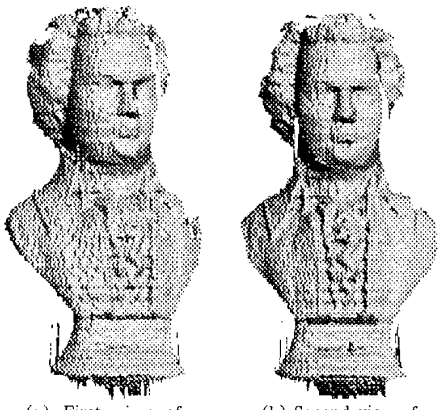
In our experiments, we acquire the range images with a range finder described in [10], which consists of a projector with a programmable liquid crystal mask and a CCD camera. The range finder works on the principle of space coding with projected pattern and triangulation. Figure 1 shows two range images taken of a Mozart bust in Cartesian image format (also called depth map or graph surface). For display purposes, they are shown in shaded intensity image form [1] computed from floating point values.

In the following sections, we discuss the issues involved

*This research was supported by DARPA contract F33615-87-c-1436, monitored by Wright Patterson Air Force Base, and by a grant from the Center for Manufacturing and Automation

†Mr. Chen was partly supported by "Pao Yu-kong and Pao Zao-long Scholarship"

[1] The pixel values of the shown images are proportional to the product of the normal vector of the underlying surface at the point and the directional vector of a predefined light source.



(a) First view of the Mozart bust (b) Second view of the Mozart bust

Figure 1: The range images of a Mozart bust

in range image registration, including the definition of registration and the methods to achieve registration (section 2.1 and 2.2), the selection of control points (section 2.3), the registration algorithm (section 2.4), and the data integration strategy (section 3). Results for the registration and object modeling stages are presented in section 2.5 and 3.3 respectively.

2 Range Image Registration

The need for surface (3D range image) registration arises when an accurate transformation is desired between two overlapping views of an object. This could be the case in 3D object recognition/localization, or in merging data from multiple views. Kamgar-Parsi *et al.* [6] have used a range image registration technique in mapping the ocean floor based on matching elevation contours of the ocean floor range images. Their method can only handle 2D transformations between range images to be registered, since it is based on 2D contour matching.

Here we present a new method for range image registration which works on the range images directly. From now on, we use the term range image registration and surface registration interchangeably, since in our domain of application, the surfaces are represented by range images. Intuitively, two views of a surface are said to be registered if they "coincide" when one view is placed at a proper position and orientation relative to the other. More rigorously, two views of a surface are said to be in registration when *any* pair of points $(\mathbf{p}_i, \mathbf{q}_j)$ from the two views representing the same surface point can be brought into coincidence by *one* rigid transformation. That is, there exists a rigid transformation T, such that

$$\forall \mathbf{p}_i \in P,\ \exists \mathbf{q}_j \in Q \ \mid\ \|T\mathbf{p}_i - \mathbf{q}_j\| = 0 \qquad (1)$$

or

$$D(P,Q) = \iint_{\Omega} \|T\mathbf{p}(u,v) - \mathbf{q}(f(u,v), g(u,v))\|^2 du dv = 0 \qquad (2)$$

where $\mathbf{p}(u,v) \in P$, $\mathbf{q}(u,v) \in Q$, P and Q are two views of the same surface, $(u,v) \in \Re \times \Re$ is the parameter space for P and Q, f and g are correspondence mapping functions (which means $\mathbf{p}(u,v)$ and $\mathbf{q}(f(u,v), g(u,v))$ represent the same surface point), $T\mathbf{p}_i$ is the result of applying T to $\mathbf{p}_i$ and Ω is the overlap region of P and Q.

In homogeneous coordinate, T can be expressed as

$$T = T(\alpha,\beta,\gamma,t_x,t_y,t_z) = \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma & t_x \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma & t_y \\ -s\beta & c\beta s\gamma & c\beta c\gamma & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (3)$$

where sx and cx stand for $\sin x$ and $\cos x$ respectively. So, in general, the transformation T needed to bring the two views into registration has 6 degrees of freedom. Thus, the task of registration is actually to search for such a transformation in the transformation parameter space so that $D(P,Q)$ in equation 2 reaches its minimum, which can be achieved by solving the following set of equations:

$$\frac{\partial D}{\partial \alpha} = 0, \frac{\partial D}{\partial \beta} = 0, \frac{\partial D}{\partial \gamma} = 0, \frac{\partial D}{\partial t_x} = 0, \frac{\partial D}{\partial t_y} = 0, \frac{\partial D}{\partial t_z} = 0 \qquad (4)$$

The difficulty is that the process is non-linear, and some iterative method must be used. Furthermore, $D(P,Q)$ may not be convex in general and there is no guarantee that a global minimum can be reached by an iterative procedure. Finally, we generally do not know the functions f and g.

Potmesil [9] used a heuristic search in the transformation parameter space to match surfaces, but the second problem mentioned above still exists. Our approach is based on the assumption that an approximate transformation between two views is known before hand, i.e., they are approximately registered. The purpose of doing this is twofold. First, we argue that the goal of the surface registration algorithm is to find a finer, more accurate transformation between different descriptions/views of an object surface and that the initial approximate transformation can be obtained through high level matching or through the knowledge of the geometry of the data acquisition setup. Second, since we are not sure that a global minimum for $D(P,Q)$ can be found in general, a good starting point is very important.

In the following sections we will discuss how the ideas in this section are implemented and how we can achieve registration without knowing the functions f and g.

2.1 Choosing the Evaluation Function For Surface Registration

According to our definition of surface registration, if we know a set of N pairs of corresponding points, called control points, in two views, $\mathbf{p}_i \in P$ and $\mathbf{q}_i \in Q$, $i = 1...N$, we can easily find the transformation by minimizing

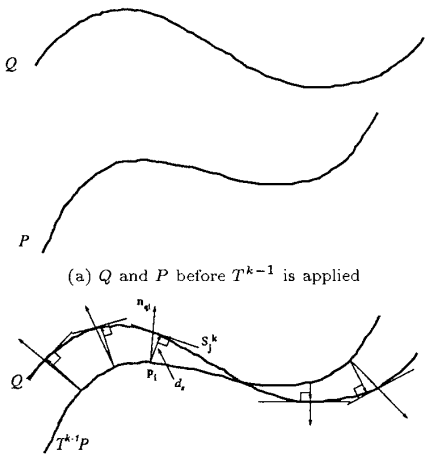
$$e = \sum_{i=1}^{N} \|T\mathbf{p}_i - \mathbf{q}_i\|^2 \qquad (5)$$

when the set size is larger than 3. Unfortunately, this correspondence information is difficult to obtain especially for non-structured surfaces.

Another way to this problem is to minimize the distances from points on one surface to the other. That is, we minimize

$$e = \sum_{i=1}^{N} \|T\mathbf{p}_i - \mathbf{q}_j\|^2, \text{ with } \mathbf{q}_j = \mathbf{q} | \min_{\mathbf{q}\in Q} \|T\mathbf{p}_i - \mathbf{q}\| \qquad (6)$$

This follows directly from equation 1, since if $\min_{\mathbf{q}\in Q} \|T\mathbf{p}_i - \mathbf{q}\| = 0$ for all $i = 1...N$, equation 6 will be zero. But this is very difficult to implement, since finding $\mathbf{q}_i$ is an optimization problem itself. Approximation of the algorithm by an iterative method can be used if we know an initial transformation T^0 that brings P in near registration with Q. In this



(a) Q and P before T^{k-1} is applied

(b) The distance from P to the tangent planes of Q

Figure 2: Distance measures between P and Q illustrated in the 2D case

case, at each iteration k, we use the previous value T^{k-1} to find $\mathbf{q}_j^k$:

$$e^k = \sum_{i=1}^{N} \|T^k \mathbf{p}_i - \mathbf{q}_j^k\|^2, \text{ with } \mathbf{q}_j^k = \mathbf{q} | \min_{\mathbf{q} \in Q} \|T^{k-1}\mathbf{p}_i - \mathbf{q}\| \quad (7)$$

With this approach, however, we need to perform the minimization on a digital surface to find $\mathbf{q}_j^k$, as is usually the case. If we use an approximation point instead of the $\mathbf{q}_j^k$ defined in equation 7, the problem becomes easier. Potmesil [9] has used the distance between the surfaces in the direction normal to the first surface as a registration evaluation function. Following this idea, we have

$$e^k = \sum_{i=1}^{N} \|T^k \mathbf{p}_i - \mathbf{q}'^k_j\|^2, \text{ with } \mathbf{q}'^k_j = (T^{k-1}\ell_i) \cap Q \quad (8)$$

where $\ell_i = \{\mathbf{a} | (\mathbf{p}_i - \mathbf{a}) \times \mathbf{n}_{\mathbf{p}_i} = 0\}$ is the line normal to P at $\mathbf{p}_i$, $\mathbf{n}_{\mathbf{p}_i}$ is the surface normal of P at $\mathbf{p}_i$ and $(T\ell_i) \cap Q$ stands for the intersection point of line ℓ_i (after transformed by T) with surface Q. [2]

The common problem with the above two iterative methods is that, at each iteration, they work with some "control points" that are not necessarily true correspondence points. The consequence is a slower convergence, since the constraints imposed by different pairs of control points can be mutually incompatible before the registration is obtained.

Our approach is to approximate Q using its tangent plane S_j at $\mathbf{q}_j$ of equation 6. Equation 6 can then be written as:

$$e = \sum_{i=1}^{N} \|T\mathbf{p}_i - \mathbf{q}'_j\|^2, \text{ with } \mathbf{q}'_j = \mathbf{q} | \min_{\mathbf{q} \in S_j} \|T\mathbf{p}_i - \mathbf{q}\| \quad (9)$$

where S_j is the tangent plane of Q at $\mathbf{q}_j$. As mentioned earlier, we do not know where $\mathbf{q}_j$ is. But if we have an initial T^0 as mentioned above, then we can start an iterative process as an approximation. In this case we can use the $\mathbf{q}'^k_j$ defined in equation 8 as an approximation to the $\mathbf{q}_j$ at each iteration. Since the distance from a point to a plane can be expressed as a linear function of the coordinates of the point, the iterative procedure can be formulated as follows (see Figure 2(b)):

$$e^k = \sum_{i=1}^{N} d_s^2(T^k \mathbf{p}_i, S_j^k) \quad (10)$$

with

$$S_j^k = \{\mathbf{s} | \mathbf{n}_{\mathbf{q}j}^k \cdot (\mathbf{q}'^k_j - \mathbf{s}) = 0\}, \mathbf{q}'^k_j = (T^{k-1}\ell_i) \cap Q$$

where d_s is the signed distance from a point to a plane and $\mathbf{n}_{\mathbf{q}j}^k$ is the surface normal vector of Q at $\mathbf{q}'^k_j$. Now we do not work with specific correspondence points, so we get rid of the problem of slower convergence. In fact, by minimizing the distance from a point to a plane, we only constrain the direction in which this distance can be reduced. The point has two other degrees of freedom, in which it can move in accordance with the constraints imposed by other points and planes. Thus, global optimization (in our case, minimization of the sum of distances) can be achieved more quickly, which has been verified in our experiments. This is an extension of the idea used by Lowe [7], who minimized point-to-line distance in object recognition.

Next, we introduce the line-surface intersection algorithm, control point selection strategy, and then the iterative registration algorithm.

2.2 Line-Surface Intersection

In this section, we present the algorithm for finding the intersection of a line with a digital surface. Let P and Q be the two views of the surface in consideration, and $\mathbf{p} \in P$. In implementing the idea of registration from the last section, we need to find the intersection of the line $\ell = \{\mathbf{a} | \mathbf{n}_\mathbf{p} \times (\mathbf{p} - \mathbf{a}) = 0\}$ (which passes through $\mathbf{p}$ and in the direction of the surface normal vector $\mathbf{n}_\mathbf{p}$ of P at $\mathbf{p}$) with surface Q. Let the intersection point be $\mathbf{q} \in Q$. Our approach is to find the intersection of the line ℓ with the tangent plane to Q in the neighborhood of prospective intersection points on Q. This is an iterative process and we need to have a prospective intersection to start with.

As an example, let us consider the case where P and Q are represented in a Cartesian coordinate system, i.e., we have $P = P(x, y)$ and $Q = Q(x, y)$, and the value of P or Q represents the distance from some reference plane. Under the assumption of approximate initial registration, $\mathbf{q}$ is expected to be in the neighborhood of $\mathbf{p}$. In the following algorithm the initial prospective intersections are chosen by projecting $\mathbf{p}$ orthographically along the z axis onto Q. Our intersection algorithm works as follows (see Figure 3 for an illustration of the process for a typical 2D case).

Let $\mathbf{p} = (x, y, P(x, y))^\tau$ be a point on P, and ℓ a line normal to P at $\mathbf{p}$. Let $x^0 = x$, $y^0 = y$; At each iteration k, for $k = 1, 2, ...$, compute $\mathbf{q}^k = (x^k, y^k, z^k)^\tau$, the intersection of ℓ with the tangent plane to Q at $(x^{k-1}, y^{k-1}, Q(x^{k-1}, y^{k-1}))^\tau$, and we stop when $\|\mathbf{q}^k - \mathbf{q}^{k-1}\| > \epsilon_d$ ($\epsilon_d > 0$).

This algorithm works well as long as the neighborhood in consideration is relatively smooth and converges in a matter of several (usually less than 5) iterations.

As for the selection of ϵ_d, we have set it to the space sampling unit of the range image, since our final goal to compute the intersection point $\mathbf{q}$ is to find the tangent plane at $\mathbf{q}$ according to the last section, we do not really need to get the exact location of $\mathbf{q}$.

[2] Here we always use the intersection point closest to P.

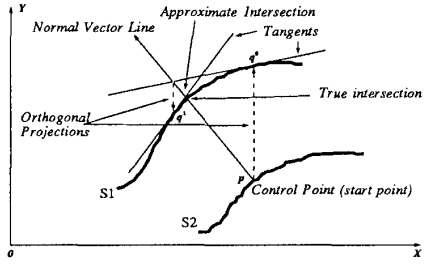


Figure 3: Intersecting a line with digital a curve

2.3 Control Point Selection

Next, we discuss the selection of control points on P. Since our control points do not need to represent meaningful surface features, we can simply pick points on P on a regular grid, instead of every known point on P to save computation time. On the other hand, we do want the control points to be in a smooth area so that the corresponding neighborhoods on surface Q will likely be smooth, and, consequently, the intersections will be found if they exist. To check the smoothness of a surface in some neighborhood, we can fit a smooth surface function to the neighborhood using least squares and check the residual standard deviation. For simplicity, we fit a plane.

2.4 The Registration Algorithm

Assume that we have two surfaces P and Q, and that we have already computed an initial transformation T_0. We rewrite equation 10 in section 2.1 as follows:

$$e^k = \sum_{i=1}^{N} d_s^2(T \circ T^{k-1}\mathbf{p}_i, S_i^k) \tag{11}$$

where

$T \circ T^{k-1} = T^k$,
$S_j^k = \{\mathbf{s}|\mathbf{n}_{\mathbf{q}j}^k \cdot (\mathbf{q}_j^k - \mathbf{s}) = 0\}$ is the tangent plane to Q at $\mathbf{q}_j^k$,
$\mathbf{n}_{\mathbf{q}j}^k$ is the normal to surface Q at $\mathbf{q}_j^k$,
$\mathbf{q}_j^k = (T^{k-1}\ell_i) \cap Q$ is the intersection point of Q with line $T^{k-1}\ell_i$,
$\ell_i = \{\mathbf{a}|(\mathbf{p}_i - \mathbf{a}) \times \mathbf{n}_{\mathbf{p}i} = 0\}$ is the line normal to P at $\mathbf{p}_i$,
$\mathbf{p}_i \in P$ is a point on P,
d_s is the signed distance from a point to a plane.

Our registration algorithm is to find the update transformation T which minimizes e^k in the above equation with a least squares method iteratively. The algorithm goes as follows:

1. Select a set of control points $\mathbf{p}_i \in P$ $(i = 1..N)$ and compute the surface normals $\mathbf{n}_{\mathbf{p}i}$ at those points. Let $T^0 = T_0$;
2. At each iteration k, repeat the following until the process converges (see below for explanation):
   (a) For each control point $\mathbf{p}_i$,
   - Apply T^{k-1} to both the control point $\mathbf{p}_i$ and the normal $\mathbf{n}_{\mathbf{p}i}$ to get $\mathbf{p'}_i$ and $\mathbf{n'}_{\mathbf{p}i}$;
   - Find the intersection $\mathbf{q}_i^k$ of surface Q with the normal line defined by $\mathbf{p'}_i$ and $\mathbf{n'}_{\mathbf{p}i}$;
   - Compute the tangent plane S_i^k of Q at $\mathbf{q}_i'$

   (b) Find the transformation T that minimizes e^k in equation 11 with a least squares method, let $T^k = T \circ T^{k-1}$;

The convergence of the procedure is tested by checking

$$\delta = \frac{\|e^k - e^{k+1}\|}{N'} \leq \epsilon_e, (\epsilon_e > 0) \tag{12}$$

where ϵ_e is a threshold set via experiment, N' $(N' < N)$ is the actual number of $\mathbf{p}_i$'s used, since some of them may not have counter part in Q. Although we could have checked e^k directly, since δ is much less sensitive to noise while e^k is a direct reflection of the noise level, a reasonable ϵ_e serves for variety of range images.

Finally, the least squares algorithm for minimizing e^k in equation 11 is generally non-linear, but it can be converted to linear least squares problem if we know that the final T^k differs from T_0 by a small amount of rotation. Since the elements in T of equation 11, as defined by equation 3, can be approximated by linear terms because of the small α, β and γ.

2.5 Test Cases With Range Image Registration

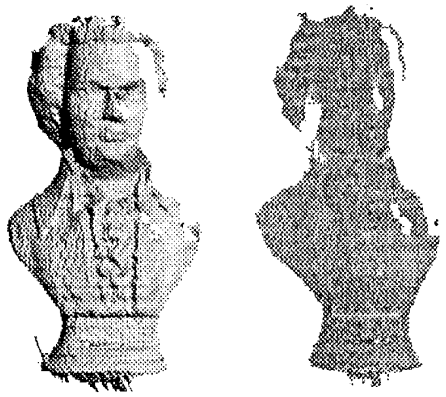
We present some examples of the range image registration algorithm. The range images used in the examples are Cartesian images with the background area already identified. As before, all range images are shown as shaded intensity images for display purposes. The actual images are 32-bit floating point number images. The two range images used in each example are taken under the same conditions, but the objects in the images were rotated 15 to 20 degrees about the y axis. In these tests, we simply use the identity matrix as the initial transformation. The error images are computed as the distances from the surface of the first image to the second in the surface normal direction of the first surface, after the found registration transformation has been applied to the first image. The spatial resolutions of the range images and the error images are all 0.5 mm, while the accuracy of the range-finder is in the neighborhood of 1mm.

In the first example, we register the two range images in Figure 1. Figure 4(a) shows the first view of the Mozart bust after the registration is performed, Figure 4(b) and (c) show the registration error image and error histogram. The second test is shown in Figure 5. The range images in the first example contain many detailed patterns and complex surface structures. But due to digitizing noise, surface features can not be detected reliably, thus matching surface features for registration may not produce satisfactory results. On the contrary, the range images in the second example contain few interesting features. Besides, the surface type is very special in that it is very difficult to define a good surface segmentation, and therefore difficult to match high level surface descriptions.

Our registration algorithm works very well on these examples, as can be seen from the error images and error histograms in Figure 4 and 5. They are very comparable to the resolution of the original data. In fact, if the images were smoothed with a small Gaussian filter, an even better registration accuracy could have been achieved. From the computed registration transformation, we found the rotation between the two Mozart images to be $-15.06°$ while the actual rotation was $-15°$. For the second test shown in Figure 5, the computed rotation of the two images is $-19.75°$ whereas the actual rotation was $-20°$.

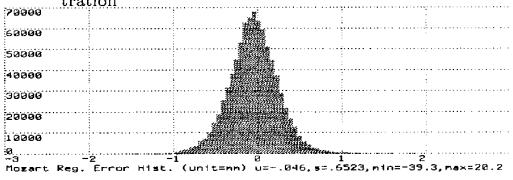
3 Integration of Multiple Range Images

In this section, we present one application of this range image registration algorithm to the modeling of compact objects. There is no attempt to obtain a high level description



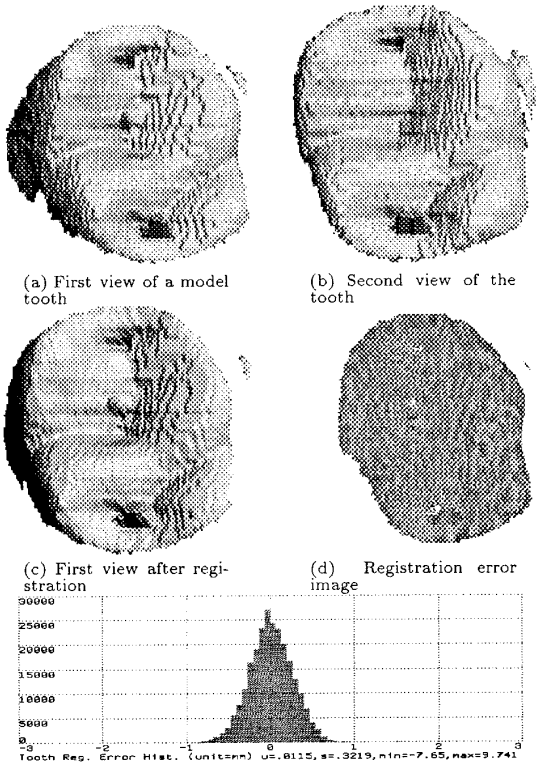
(a) First view of Mozart after registration

(b) Registration error image



(c) Histogram of the error image

Figure 4: Registration results for the Mozart bust



(a) First view of a model tooth

(b) Second view of the tooth

(c) First view after registration

(d) Registration error image

(e) Histogram of error image

Figure 5: Registration results for the tooth

of the object in this study, although it should be possible, based on our results.

In order to get a description of the whole object surface, multiple range images of the object from different vantage points are needed. While putting an object on a turn table is enough for obtaining a wrap-around representation of the object without use of registration, as was done in [12], it is seldom the case that all surface areas can be covered in this manner. But when taking range images from different object poses, the exact spatial relationship between them is lost. This is where our registration algorithm can be used.

Our modeling process goes as follows. We first put the object on a turn table to take a set of 4 to 8 side view range images of the object, and then the object is laid down and the range images of the top and bottom areas of the object are taken. The number of side views depends on the complexity of the object surface structure. For the range images taken from the top and bottom views, we need the approximate relationship between them and those of the side views in order to use our registration algorithm, as assumed in the first part of this paper. This is currently done by asking the user to estimate the 3 rotation angles, and they do not have to be accurate. Then the registration algorithm is applied to bring the range image of the top and bottom views in registration with those of the side views.

3.1 Object-Centered Representation

The original range images come as three separate dense images $x(u,v)$, $y(u,v)$ and $z(u,v)$ in the parametric space of the sensor. Our goal is to build an object-centered model for a class of compact objects. The object is described in a cylindrical or spherical coordinate system, centered in the object. This kind of representation provides us with an intermediate representation towards a higher level description of the object.

Once the object coordinate frame has been selected, the range image from each view of the object is transformed to this coordinate frame with transformations derived from the registration process. They are then further reparameterized through interpolation into cylindrical or spherical coordinates Since all data are now in the same parameter space, additional data from different views can be merged into one by simple average in the overlapping areas. The actual implementation also includes decisions about avoiding outliers.

3.2 Global Registration

For this specific modeling procedure, we can work globally for better overall results. That is, when we merge a current view range image, instead of registering it with only a neighboring view(s), we register it with the merged data from all previously processed views to find out the needed transformation. In this way, the information from all the previously merged views is used, and the possible error accumulation due to successive registrations with range images of neighboring views can be avoided.

3.3 Results and Discussion

We present modeling results for two objects. Each object measures about 10cm on their longest sides. One is a free-form shaped wood blob, and the other is a plaster model of a tooth. Both objects are free-form objects with few distinct surface features. Thus, matching detected surface features for registration would prove very difficult. The sur-

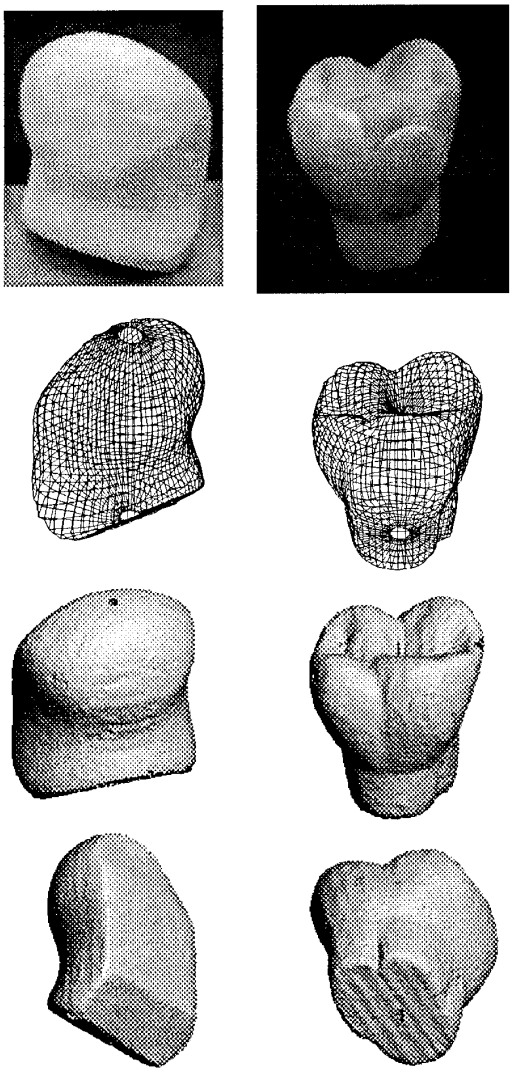


Figure 6: The wood blob, the plaster tooth and the derived models for them

face structures of these objects (especially the model tooth) are very special in that it is very difficult to define a reliable segmentation to achieve high level description.

Two examples are shown in figure 6. The first row shows the intensity images of the orignal objects. The second row shows the wireframe plot of the acquired models. The third and fouth rows show the rendered images from the acquired models. In these examples, we have used 8 side views and 6 to 8 top and bottom views for each object with 45° of rotation angle between successive side views for simplicity.

As can be seen, the final integrated models are generally smooth partly due to the averaging effect in the integration of multiple range images. The smoothness is also due to the good performance of the registration algorithm, since, otherwise, the data from different views would not "fit" together and ridges and valleys on the object surface would have been blurred.

3.4 Conclusion and Future Research

We have presented a new method for constructing a complete surface model for compact objects. This new method is based on registering range images from multiple views of the object before view integration, using a registration algorithm developed in this research. The range image registration algorithm is based on minimizing a distance measure function derived from the definition of 3D surface registration.

The drawback of this modeling process is that the representation scheme may not be powerful enough to directly accommodate more complex objects.

Future research will address the development of more powerful schemes for intermediate representation, to handle complex objects not representable by cylindrical/spherical coordinate system. Incorporation of some surface matching system such as [11] to provide the registration algorithm with needed initial transformation, will enable us to build a fully automatic modeling system.

Acknowledgment

The authors would like to thank Dr. Philippe Saint-Marc for many valuable discussions we had during this research, and for helping the authors in implementing the algorithms.

References

[1] Narendra Ahuja and Jack Veenstra. Generating Octrees from Object Silhouettes in Orthogaphic Views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-11(2):137–149, 1989.

[2] Bir Bhanu. Representation and Shape Matching of 3-D Objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(3):340–351, May 1984.

[3] C. H. Chien, Y. B. Sim, and J. K. Aggarwal. Generation of Volume/Surface Octree from Range Data. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 254–260, 1988.

[4] T.-J. Fan, G. Medioni, and R. Nevatia. Recognizing 3-d objects using surface descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(11):1140–1157, November 1989.

[5] F.P. Ferrie and M.D. Levine. Integrating information from multiple views. In *Proceedings of the IEEE Workshop on Computer Vision*, pages 117–122, December 1987.

[6] Behrooz Kamgar-Parsi, Jeffrey L. Jones, and Azriel Rosenfeld. Registration of Multiple Overlapping Range Images: Scenes without Distinctive Features. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 282–290, San Diego,CA, June 1989.

[7] David G. Lowe. *Perceptual Organization and Visual Recognition*. Kluwer Academic Publishers, 1985.

[8] B. Parvin and G. Medioni. A constraint satisfaction network for matching 3d objects. In *Proceedings of the International Conference on Neural Networks*, volume II, pages 281–286, Washington, D.C, June 1989.

[9] Michael Potmesil. Generating Models for Solid Objects by Matching 3D surface Segments. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1089–1093, Karlsruche, West Germany, August 1983.

[10] Kosuke Sato and Seiji Inokuchi. Range-Imaging System Utilizing Nematic Liquid Crystal Mask. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 657–661, 1987.

[11] F. Stein and G. Medioni. Toss - a system for efficient three dimensional object recognition. In *Proceedings of the DARPA Image Understanding Workshop*, Pittsburgh, Pennsylvania, September 1990.

[12] B. C. Vemuri and J. K. Aggarwal. 3-D Model Construction from Multiple Views Using Range and Intensity Data. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 435–437, 1986.

[13] Y. F. Wang and J. K. Aggarwal. Integration of Active and Passive Sensing Techniques for Representating Three-Dimensional Objects. *IEEE Transactions on Robotics and Automation*, 5(4):460–471, August 1989.