

Object modelling by registration of multiple range images

Yang Chen and Gérard Medioni

We study the problem of creating a complete model of a physical object. Although this may be possible using intensity images, we here use 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 either assume this transformation to be known (which is extremely difficult for a complete model), or compute 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 modelling procedure, and illustrate them on real range images of complex objects.

Keywords: object modelling, 3D surface registration, range image registration

Creating models of physical objects is a necessary component machine of biological vision modules. Such models can then be used in object recognition, post 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 objects. Some researchers bypass the problem by using a model which consists of multiple views^{1,2}, but this is not always enough.

Institute for Robotics and Intelligent Systems, Departments of Electrical Engineering and Computer Science, University of Southern California, Los Angeles, CA 90089-10273, USA

Paper received: 22 July 1991

0262-8856/92/003145-11 © 1992 Butterworth-Heinemann Ltd

vol 10 no 3 april 1992

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.

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 solving the *correspondence* problem. This problem has been at the core of many previous research efforts: Bhanu² developed an object modelling system for object recognition by rotating object through known angles to acquire multiple views. Chien *et al.*³ and Ahuja and Veenstra⁴ 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 Aggarwal⁵. Vemuri and Aggarwal⁶ have used a base-plane pattern to determine the interframe rotation of objects by locating the pattern in the intensity images, which are taken at the same time as the range images. These techniques all have difficulties 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 cannot take advantage of the object surface structure in choosing vantage views. Ferrie and Levine⁷ obtained interframe correspondence by matching surface features. The accuracy of this method depends on the accuracy of the feature detection technique used. Potmesil⁸ developed a system for modelling the complete surface of an object by taking multiple range images 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. Recently, Besl⁹ presented a uniformed formulation for the so-called 'free-form surface matching' problem, and suggested various possible ways to solve the problem, with the conclusion that reliable detection of generic surface points, curves or regions are essential to the matching problem.

Our object modelling 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. see References 1, 10 and 11). 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 setup described by Sato and Inokuchi¹² 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 stripe 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* computed from floating point values.

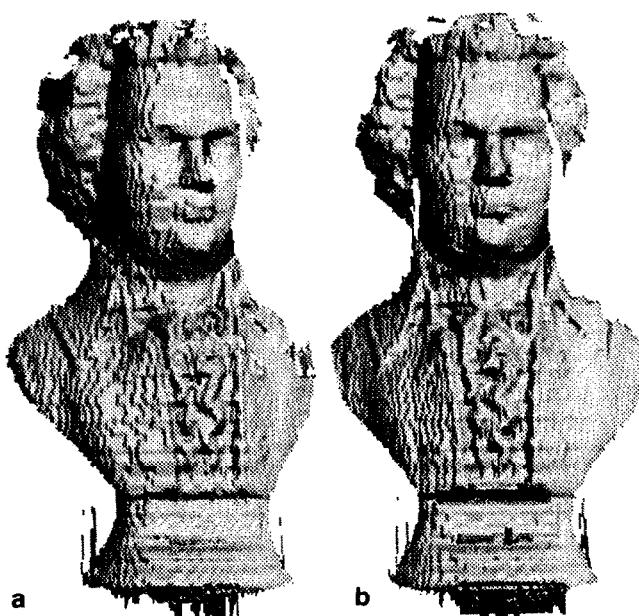


Figure 1. Mozart bust range images. (a) First view of the Mozart bust (217×422); (b) second view of the Mozart bust (221×420)

*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.

In the following sections, we discuss the issues involved in range image registration, including the definition of registration and the methods to achieve registration, the selection of control points, the registration algorithm, and the data integration strategy. Results for the registration and object modelling stages are finally presented.

RANGE IMAGE REGISTRATION

The need for range image or, more generally, surface 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. Surface registration technique can also be used in inspection, geometry analysis, terrain matching and modelling and medical image analysis⁹. Many object recognition systems also solve the problem of obtaining a transformation between the scene and the model for pose estimation and/or as a verification^{1, 10, 11, 13, 14}. All of these systems use some kind of feature correspondence to estimate the transformation. The accuracy of the derived transformation from these systems thus depends mostly on the position accuracy of the used features, and is often not good enough for model construction. In addition, feature extraction can be extremely difficult for objects with free-form surfaces. Kamgar-Parsi *et al.*¹⁵ have used a range image registration technique in mapping the ocean floor, where they achieved registration by matching elevation contours. However 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. We try to solve the problem in the context of object modelling, not to replace any of the systems mentioned above, but rather to improve the transformation given by some higher level systems, such as the ones mentioned. 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, we say that two views of a surface are registered if they coincide when one view is placed at a proper position and orientation relative to the other (here we use the term 'view' to mean the rendered representation of the object surface from a specific viewpoint). More precisely, two views of a surface are said to be in registration when *any* pair of points (\mathbf{p} , \mathbf{q}) from the two views representing the same surface point can be related to each other by *one* rigid spatial transformation, i.e. there exists a rigid transformation T , such that:

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

where P and Q are two views of the same surface, $T\mathbf{p}$ is the result of applying transformation T to \mathbf{p} , and T is a transformation, which can be expressed in matrix form as follows in homogeneous coordinates:

$$T = T(\alpha, \beta, \gamma, t_x, t_y, t_z) =$$

$$\begin{array}{ll} \cos\alpha\cos\beta & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma \\ \sin\alpha\cos\beta & \sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma \\ -\sin\beta & \cos\beta\sin\gamma \\ 0 & 0 \\ \cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma & t_x \\ \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma & t_y \\ \cos\beta\cos\gamma & t_z \\ 0 & 1 \end{array} \quad (2)$$

where α , β and γ are rotation angles about the x , y and z axes, respectively, and t_x , t_y and t_z are the translation components. In general, the transformation T needed to bring the two views in 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 equation (1) is satisfied. In practice, the problem can be expressed as solving T to minimize the following measure:

$$D(P, Q) = \sum \|T\mathbf{p} - f(\mathbf{p})\|$$

where f is a correspondence mapping function:

$$f: P \rightarrow Q \mid \forall \mathbf{p} \in P, f(\mathbf{p}) \in Q$$

The difficulty in solving for T is that the process is highly non-linear⁹ when f is unknown. 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.

Potmesil⁸ used a heuristic search in the transformation parameter space to match surfaces, but the second problem mentioned above still exists. In fact, there are other ways in which we can reduce our search effort. 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 discuss how the ideas in this section are implemented and how we can achieve registration without knowing the function f .

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 $\mathbf{q}_i \in Q$, $i = 1 \dots N$, we can easily find the transformation by

minimizing:

$$e = \sum_{i=1}^N \|T\mathbf{p}_i - \mathbf{q}_i\|^2 \quad (4)$$

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, i.e. we minimize:

$$e = \sum_{i=1}^N \|T\mathbf{p}_i - \mathbf{q}_i\|^2, \text{ with } \mathbf{q}_i = \mathbf{q} \mid \min_{\mathbf{q} \in Q} \|T\mathbf{p}_i - \mathbf{q}\| \quad (5)$$

This follows directly from equation (1), since if $\min_{\mathbf{q} \in Q} \|T\mathbf{p}_i - \mathbf{q}\| = 0$ for all $i = 1 \dots N$, equation (5) 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 case, at each iteration k , we use the previous value T^{k-1} to find \mathbf{q}_j^k :

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

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 (6), the problem becomes easier. Potmesil⁸ 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:

$$e^k = \sum_{i=1}^N \|T^k\mathbf{p}_i - \mathbf{q}'_j^k\|^2, \text{ with } \mathbf{q}'_j^k = (T^{k-1}\ell_i) \cap Q \quad (7)$$

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 ,[†] and ' \times ' stands for vector product.

The common problem with the above two iterative methods is that, at each iteration, they work with some 'control points' that do not necessarily correspond to the same points on the surface. 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 (5). Equation (5) can then be written as:

$$e = \sum_{i=1}^N \|T\mathbf{p}_i - \mathbf{q}'_j\|^2, \text{ with } \mathbf{q}'_j = \mathbf{q} \mid \min_{\mathbf{q} \in S_j} \|T\mathbf{p}_i - \mathbf{q}\| \quad (8)$$

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

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

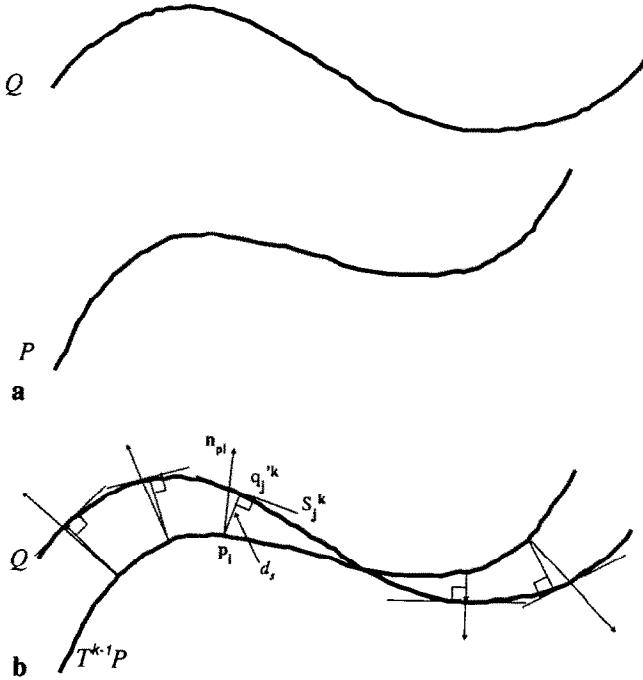


Figure 2. Distance measures between P and Q illustrated in the 2D case. (a) Q and P before T^{k-1} is applied; (b) distance to the tangent plane of Q

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}_j^k defined in equation (7) 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 2b):

$$e^k = \sum_{i=1}^N d_s^2(T^k \mathbf{p}_i, S_j^k) \quad (9)$$

with:

$$S_j^k = \{\mathbf{s} | \mathbf{n}_{\mathbf{q}_j}^k \cdot (\mathbf{q}_j^k - \mathbf{s}) = 0\}, \mathbf{q}_j^k = (T^{k-1} \ell_i) \cap Q$$

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}_j^k and ‘ \cdot ’ stands for scalar product. 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 Lowe¹⁶, who minimized point-to-line distance in object recognition.

Next, we first introduce the iterative registration algorithm. Then line-surface intersection algorithm, control point selection strategy will be discussed.

Registration algorithm

Assume that we have two surfaces P and Q , and an initial transformation T_0 which is applied to P . We

rewrite equation (9)

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

where:

- $T \circ T^{k-1} = T^k$ is the composite transformation,
- $S_i^k = \{\mathbf{s} | \mathbf{n}_{\mathbf{q}_j}^k \cdot (\mathbf{q}_j^k - \mathbf{s}) = 0\}$ 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,
- (‘ \cdot ’ and ‘ \times ’ stand for scalar and vector products, respectively.)

Our registration algorithm is to find the 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 $\mathbf{p}_i \in P$ ($i = 1 \dots N$) (see below for selection process) and compute the surface normals $\mathbf{n}_{\mathbf{p}_i}$ at those points. Let $T^0 = T_0$;
2. At each iteration k for $k = 1, 2, 3, \dots$, 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 $\mathbf{n}'_{\mathbf{p}_i}$ (see above);
 - Compute the tangent plane S_i^k of Q at \mathbf{q}_i^k ;
 - (b) Find the transformation T that minimizes e^k in equation (10) with a least squares method;
 - (c) Let $T^k = T \circ T^{k-1}$.

The convergence of the procedure is tested by checking:

$$\delta = \frac{|e^k - e^{k-1}|}{N'} \leq \epsilon_e, \quad (\epsilon_e > 0) \quad (11)$$

where ϵ_e is a threshold set via experiment, N' is the actual number of \mathbf{p}_i 's used, since some of them may not have counter parts 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 a variety of range images.

The time complexity for each iteration of this algorithm is linear to the number of control points used.

Line-surface intersection

In this section we present the algorithm for finding the intersection of a line with a digital surface, which is

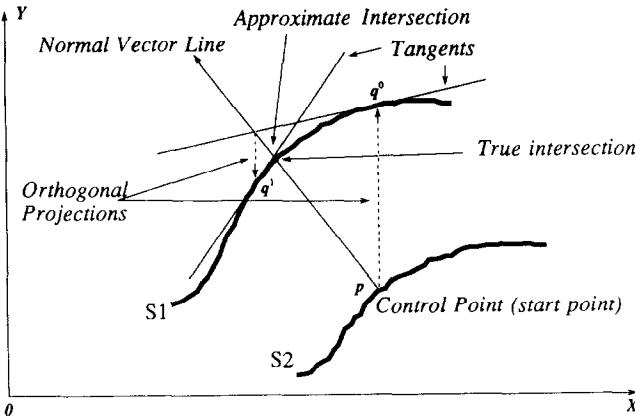


Figure 3. Intersecting a line with a digital curve

needed in equation (9). 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 $\ell = \{\mathbf{a} | \mathbf{n}_P \times (\mathbf{q} - \mathbf{a}) = 0\}$ (which passes through \mathbf{p} and in the direction of the surface normal vector \mathbf{n}_P of P at \mathbf{p}) with surface Q . Let the intersection point be $\mathbf{q} \in Q$. Since we do not have an analytical form for surface Q , we have to use an approximation. Our approach is to find the intersection of the line ℓ with the tangent plane to Q in the neighbourhood of prospective intersection points on Q instead. 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{p} is expected to be in the neighbourhood of \mathbf{q} . 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):

1. 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 $x^0 = x$, $y^0 = y$;
2. At each iteration k , for $k = 1, 2, \dots$, compute $\mathbf{q}^k = (x^k, y^k, z^k)^\tau$, the intersection of ℓ with the tangent plane of Q at $(x^{k-1}, y^{k-1}, Q(x^{k-1}, y^{k-1}))^\tau$;
3. We stop when $\|\mathbf{q}^k - \mathbf{q}^{k-1}\| \leq \epsilon_d$ ($\epsilon_d > 0$).

Similar to the Newton method for finding roots of a function of type $y = f(x)$, the convergence of this algorithm depends on the choice of the starting point, which is the orthographic projection of \mathbf{p} on Q along the z axis, and on the surface characteristics of the neighbourhood of the actual intersection point. So, if the neighbourhood of \mathbf{q} is smooth and the assumption about the initial approximate registration is true, then the projection of \mathbf{p} onto Q will be close to q , and we will have a high likelihood of a convergence. Typically, one intersection takes 3 to 5 iterations for the ϵ_d specified below. The algorithm does not work when either the projection of \mathbf{p} is out of the represented surface by Q or the surface normal is nearly perpendicular to the z axis.

The surface normals needed in computing line-

surface intersections are computed using surface fitting¹⁷. If the range image is represented as a Cartesian image $f(x, y)$, we first compute the first order derivatives $\partial f / \partial x$ and $\partial f / \partial y$ using the finite difference mask and compute the normal as follows:

$$\left(1, 0, \frac{\partial f}{\partial x}\right)^\tau \times \left(0, 1, \frac{\partial f}{\partial y}\right)^\tau$$

When the range images are specified as three separate dense images of 3D coordinates $x(u, v)$, $y(u, v)$ and $z(u, v)$, surface normals are computed as follows. The surface fitting technique is used on each of the x , y and z images to get partial differential estimates for each of them with respect to the parameters u and v . Then the surface normal is computed by:

$$\left(\frac{\partial x}{\partial u}, \frac{\partial y}{\partial u}, \frac{\partial z}{\partial u}\right)^\tau \times \left(\frac{\partial x}{\partial v}, \frac{\partial y}{\partial v}, \frac{\partial z}{\partial v}\right)^\tau$$

For a fixed window size in evaluating surface normals, the time complexity for each iteration of this line-surface intersection algorithm is constant.

As for the selection of ϵ_d , we have set it to the space sampling unit of the range image, which means that we terminate the iteration process when the computed intersection converges to within the neighbourhood of a spatial sampling unit. This is because our final goal of computing the intersection point \mathbf{q} is to find the tangent plane at \mathbf{q} according to the last section, and the exact location of \mathbf{q} is not very important. A second degree surface fitting can be used if more accurate intersection is needed.

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, and we only need them from one surface P , we can simply pick points on P on a regular grid, instead of every point on P to save computation time. On the other hand, we do want the control points to be in smooth areas for two reasons. First, the corresponding neighbourhoods on surface Q of the control points will likely be smooth and, consequently, the outcome of the line-surface intersection algorithm will be more reliable and stable. Second, the surface normal can be computed more reliably. This makes a major difference between our control point selection strategy and most other feature-based matching algorithms which seek points or features that represent certain geometry context for localization purpose. This is because our registration algorithm does not rely on the accurate correspondence of the control points or other features, but uses the constraints from the overall shape of the surfaces.

To check the smoothness of a surface in some neighbourhood, we can fit a smooth surface function to the neighbourhood using a least squares method and check the residual standard deviation of the fit. For implementation simplicity, we fit a plane. The threshold on the residual standard deviation is currently chosen through experiment, and it should not be a problem with some noise estimation procedure on the range data. One example is shown in Figure 4 for the

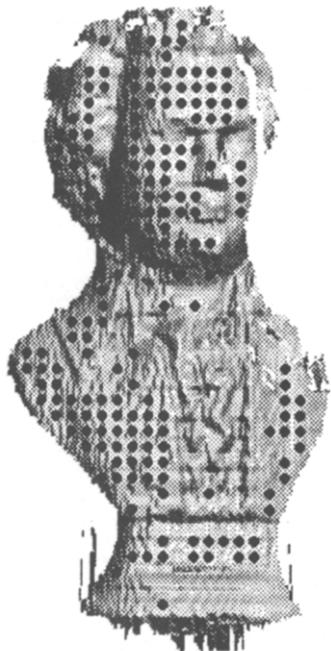


Figure 4. Example of computed control points for Figure 1a

chosen control points of the range image in Figure 1a, where the threshold for the residual standard deviation of a 9×9 window plane fitting was set to half of a spatial sampling unit.

Other issues relating to the selection of control points are the number of the control points and their distribution on the surface. Depending on the type of surface and how much the two range images overlap, the control points needed for a good registration usually range from 50 to 200, but more can be used if needed. Since more control points means more computation, we can begin with fewer control points and increase the number when we come close to convergence. Currently we have not adopted a measure to ensure a good distribution of control points, since it is a complicated issue relating to global surface properties and evaluation criteria. But it is important for a robust registration algorithm.

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

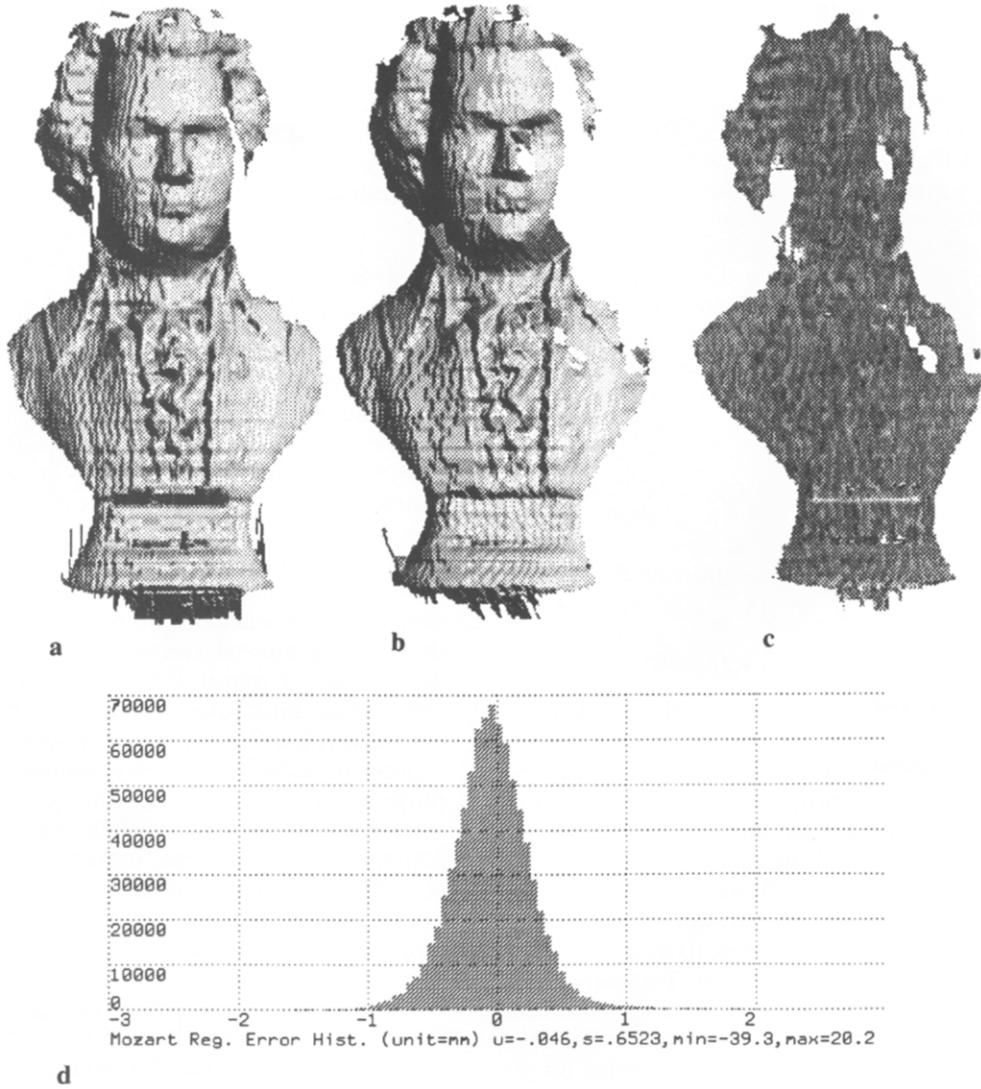


Figure 5. Registration results for the Mozart bust. (a) Second view of Mozart; (b) first view of Mozart after registration; (c) registration error image; (d) histogram of error image

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 used 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, once registration has been performed.

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 neighbourhood of 1 mm.

In the first example we register the two range images in Figure 1. Figure 5b shows the first view of the Mozart bust after the registration is performed, Figures 5c and d show the registration error image and error histogram. The second test is shown in Figure 6 for the range images from a model tooth. The range images in the first example contain many detailed patterns and complex surface structures. But due to digitizing noise,

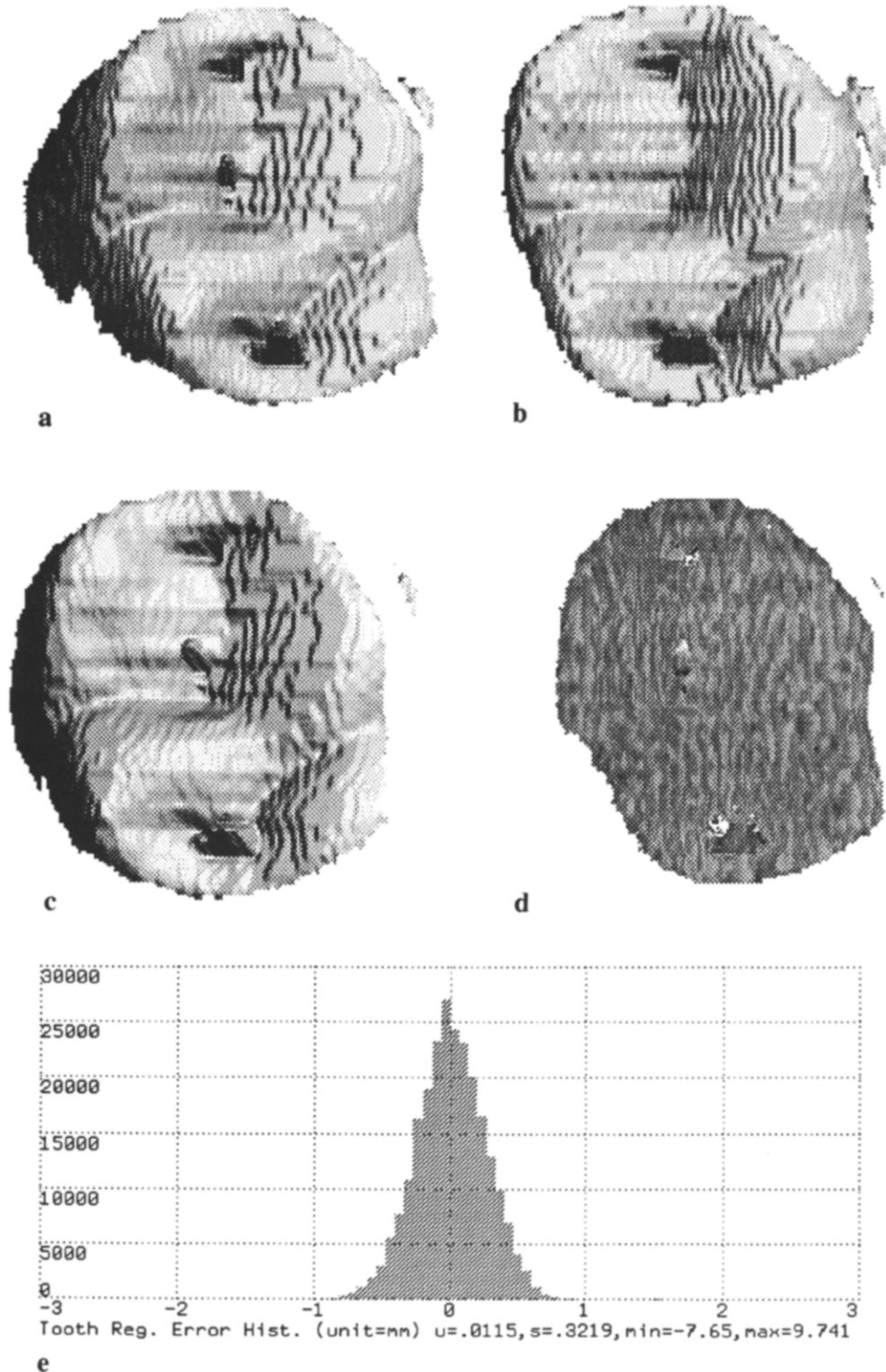


Figure 6. Registration results for the tooth. (a) First view of a model tooth (175 × 168); (b) second view of the tooth (163 × 168); (c) first view after registration; (d) registration error image; (e) histogram of error image

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 have 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 Figures 5 and 6. They are very comparable to the resolution of the original data. In fact, if the images are smoothed with a small Gaussian filter, an even better registration accuracy can be achieved. From the computed registration transformation, we found the rotation between the two Mozart images to be -15.06° while the actual rotation is -15° . For the second test shown in Figure 6, the computed rotation of the two images is -19.75° while the actual value is -20° .

The running times (elapse time) for the tests on a Symbolics 3620 Lisp Machine for the whole registration processes, from control point selection to transformation output, are as follows. It took 20 seconds for the Mozart image with 82 control points (78 of which found correspondence) and seven iterations. The tooth image took 15 seconds with 88 control points (72 of which found correspondence) and six iterations. In both cases ϵ_e in equation (11) were set to 0.01.

INTEGRATION OF MULTIPLE RANGE IMAGES

In this section we present one application of this range image registration algorithm to the modelling of compact objects. There is no attempt to obtain a high level description of the object in this study, although it should be possible, based on our results.

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 rotary table is enough for obtaining a wrap-around representation of the object without use of registration as was done by Vemuri and Aggarwal⁶, 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 modelling process goes as follows. We first put the object on a rotary 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 to use our registration algorithm, as assumed in the first part of this paper. This is done by asking the user to estimate the three rotation angles, and they do not have to be accurate. Then the registration algorithm is applied to bring the range images of the top and bottom views in registration with those of the side views. The details for merging data are described in the following sections.

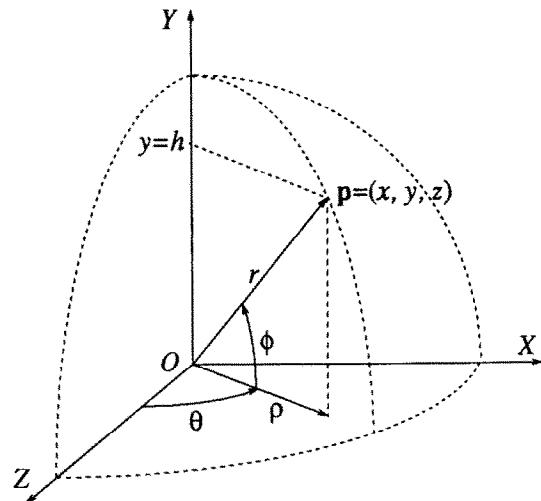


Figure 7. Relationship between Cartesian, cylindrical and spherical coordinate systems. Cylindrical coordinates: $\mathbf{p}_c = (\theta, \rho, h)$; spherical coordinates: $\mathbf{p}_s = (\theta, \phi, r)$

Object-centred 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-centred model for a class of compact objects. The object is described in a cylindrical or spherical coordinate system (based on the type of object, e.g. for an elongated shape we use cylindrical coordinates) centred in the object (see Figure 7). 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 from *a priori* knowledge (e.g. for side views) or from registration. They are then further reparameterized through interpolation into cylindrical or spherical coordinates (see Medioni *et al.*¹⁸ for the reparameterization method). 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.

Global registration

For this specific modelling 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 the neighbouring view, we register it with the merged data from all previous 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 neighbouring views can be avoided.

Results and discussion

We present results for two different objects. One is a free-form shaped wood block, 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 surface 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.

In Figure 8 the original wood block intensity image and some of the range images (shaded) used in the modelling process are shown. The integrated results are shown in Figure 9, where the model is shown in a spherical coordinate map, reconstructed intensity images and wireframe plots. In Figure 10, the results for the model tooth are shown. In these examples, we have used eight side views and 0–8 top and bottom views for each object with 45° of rotation angle between successive side views for simplicity.

From the results, we can see that, although the original range images (as shown in Figure 8) exhibit heavy digitization noise, the final integrated models are generally smooth, partly due to the averaging effect of the integration of multiple range images. The shape of the object surfaces in the ridge and valley areas has been kept very well, which shows the good performance of the registration algorithm since, otherwise, the data from different views would not 'fit' together, and the ridges and valleys on the object surface would have been blurred. The spherical coordinate maps shown in Figures 9 and 10 contain a few small uncovered areas, especially in the pole areas. This is due to the fact that the poles are special points for our representation, and

to the process of mapping range images into spherical maps. Other small uncovered areas can be solved by a more careful selection of the individual views in taking range images.

CONCLUSION AND FUTURE RESEARCH

We have presented a new method for constructing a complete surface model for a compact object. This new method is based on registering range images from multiple views of the object followed by view integration. The range image registration algorithm is based on minimizing a distance measure function derived from the definition of 3D surface registration. The modelling algorithm has been successfully applied to the modelling of objects with complex surface structures. The registration algorithm works directly on range data and provides an accurate registration.

The drawback of this modelling process is that the representation scheme may not be powerful enough to directly accommodate more complex objects. Currently the registration algorithm must be given an initial approximate transformation.

Future research will address the development of more powerful schemes for intermediate representation, to handle complex objects not representable by a cylindrical/spherical coordinate system. Incorporation of some surface matching system such as that presented by Stein and Medioni¹⁰ to provide the registration algorithm with the needed initial transformation, will enable us to build a fully automatic modelling system.

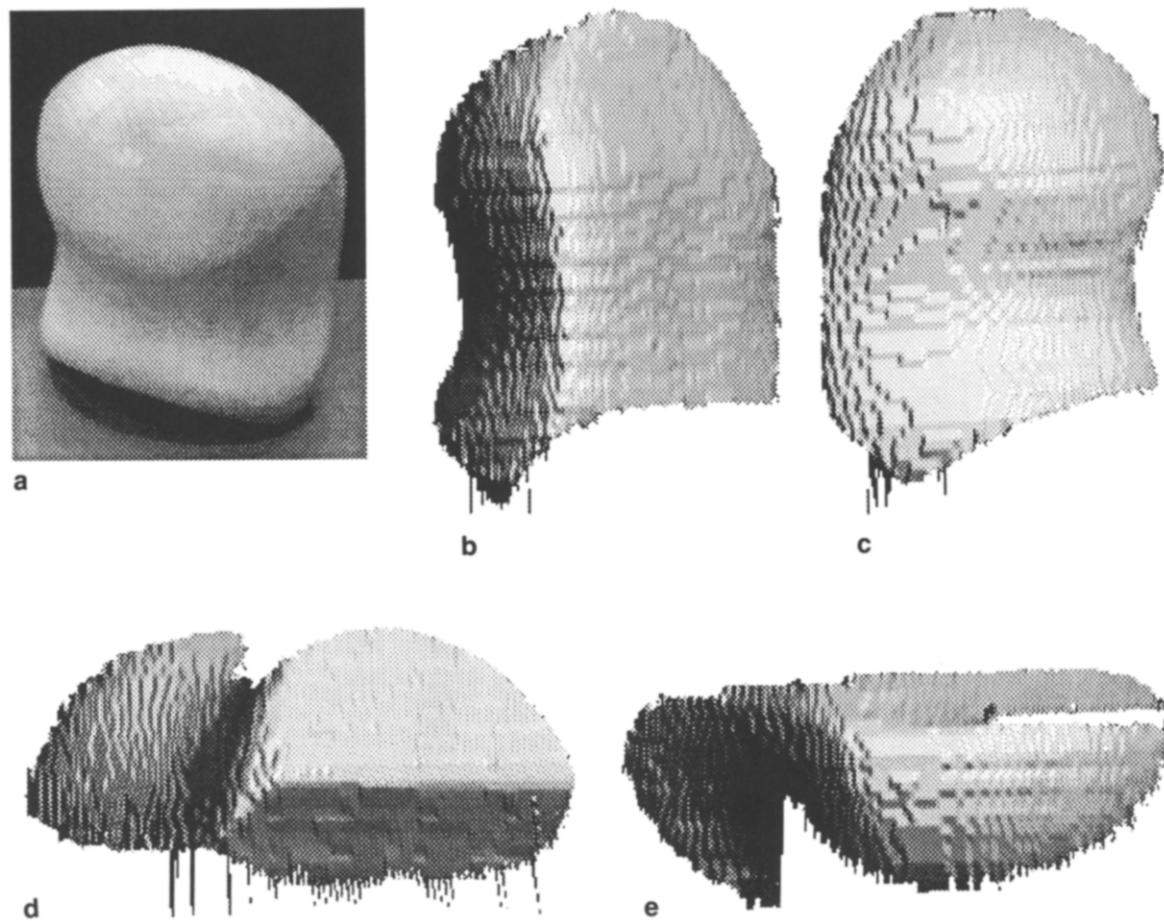


Figure 8. Wood block and four of the range image views used for model construction. (a) Intensity image of the wood block; (b) range image view 1; (c) range image view 2; (d) range image view 3; (e) range image view 4

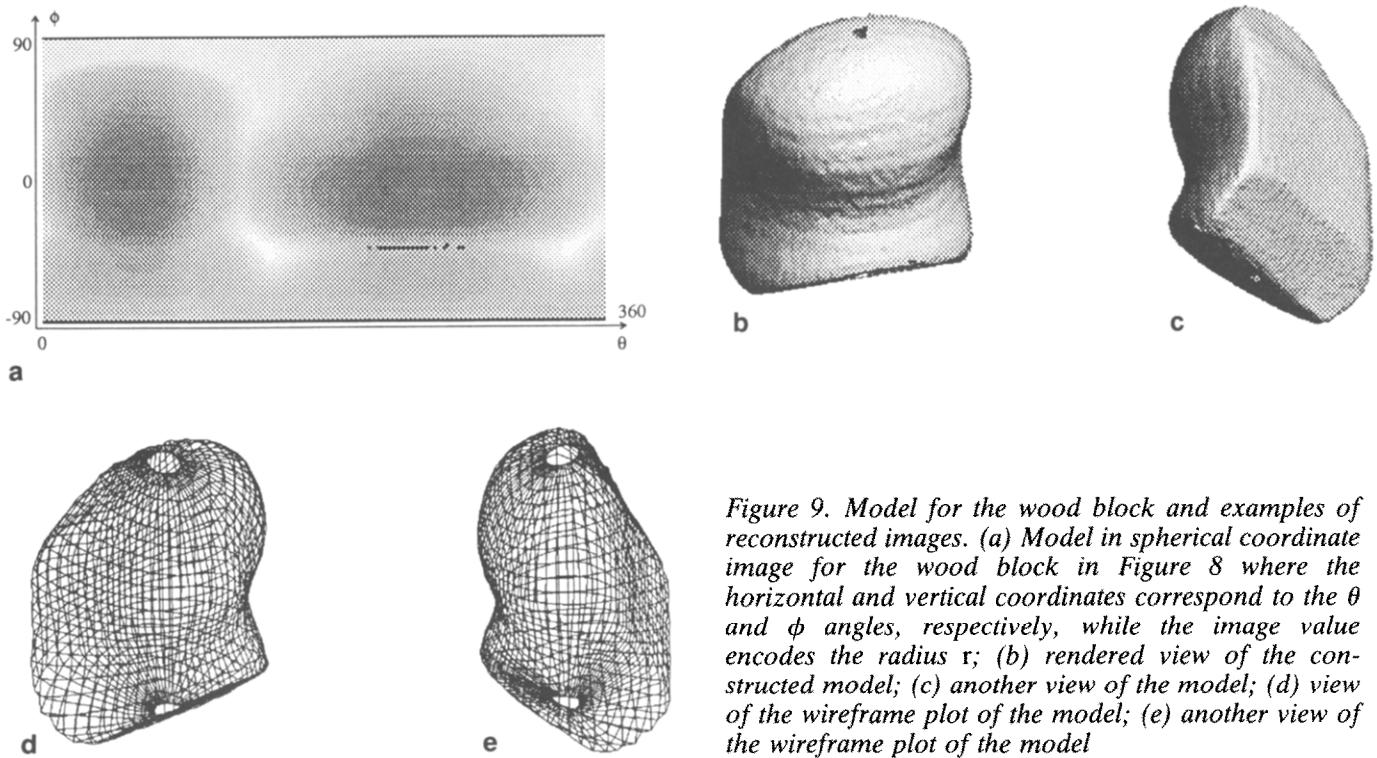


Figure 9. Model for the wood block and examples of reconstructed images. (a) Model in spherical coordinate image for the wood block in Figure 8 where the horizontal and vertical coordinates correspond to the θ and ϕ angles, respectively, while the image value encodes the radius r ; (b) rendered view of the constructed model; (c) another view of the model; (d) view of the wireframe plot of the model; (e) another view of the wireframe plot of the model

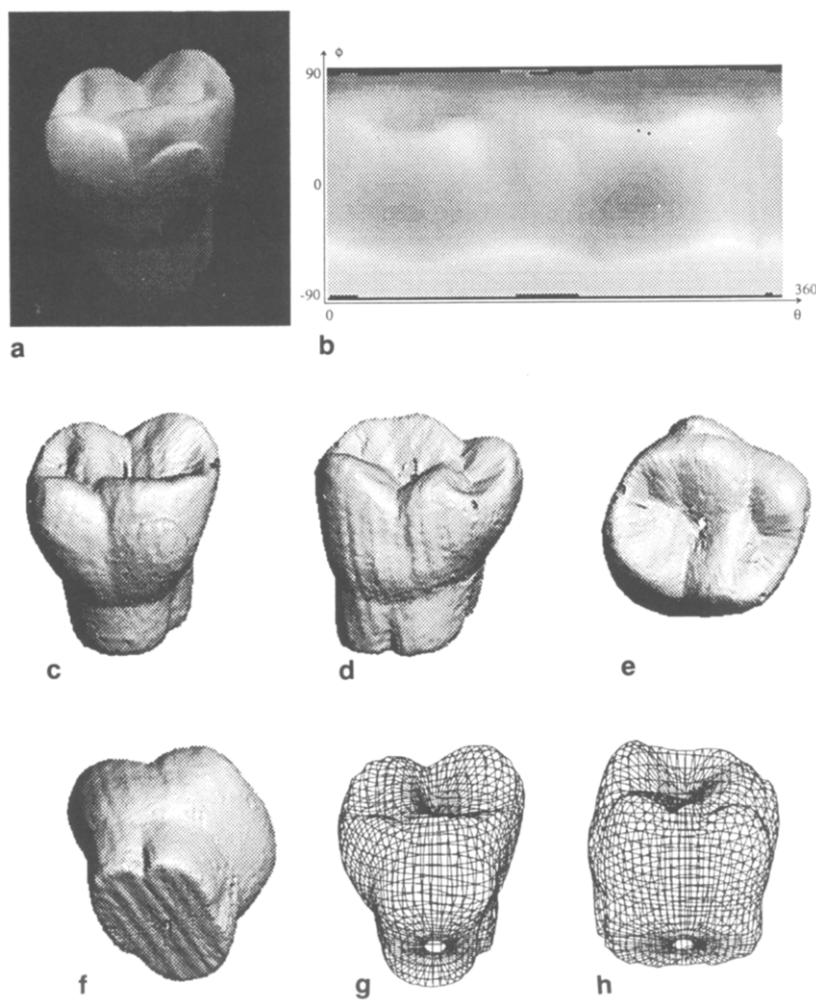


Figure 10. Plaster tooth and the derived model for it. (a) Intensity image of the plaster tooth; (b) model in spherical coordinate image for the model tooth; (c)–(f) rendered images of model; (g) one view of the wireframe plot of model; (h) another view of the wireframe plot of model

ACKNOWLEDGEMENTS

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. 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.

REFERENCES

- 1 **Fan, T-J, Medioni, G and Nevatia, R** 'Recognizing 3-d objects using surface descriptions', *IEEE Trans. PAMI*, Vol 11 No 11 (November 1989) pp 1140–1157
- 2 **Bhanu, B** 'Representation and shape matching of 3-D objects', *IEEE Trans. PAMI*, Vol 6 No 3 (May 1984) pp 340–351
- 3 **Chien, C H, Sim, Y B and Aggarwal, J K** 'Generation of volume/surface octree from range data', *Proc. Conf. on Comput. Vision & Pattern Recogn.*, Ann Arbor, MI (1988) pp 254–260
- 4 **Ahuja, N and Veenstra, J** 'Generating octrees from object silhouettes in orthographic views', *IEEE Trans. PAMI*, Vol 11 No 2 (1989) pp 137–149
- 5 **Wang, Y F and Aggarwal, J K** 'Integration of active and passive sensing techniques for representing three-dimensional objects', *IEEE Trans. Robotics & Automat.*, Vol 5 No 4 (August 1989) pp 460–471
- 6 **Vemuri, B C and Aggarwal, J K** '3-D model construction from multiple views using range and intensity data', *Proc. Conf. on Comput. Vision & Pattern Recogn.*, Miami Beach, FL (1986) pp 435–437
- 7 **Ferrie, F P and Levine, M D** 'Integrating information from multiple views', *Proc. IEEE Workshop on Comput. Vision*, Miami Beach, FL (December 1987) pp 117–122
- 8 **Potmesil, M** 'Generating models for solid objects by matching 3D surface segments', *Proc. Int. Joint Conf. on Artif. Intell.*, Karlsruhe, Germany (August 1983) pp 1089–1093
- 9 **Best, P J** 'The free-form surface matching problem', in **H Freeman (ed)**, *Machine Vision for Three-Dimensional Scenes*, Academic Press, New York (1990) pp 25–71
- 10 **Stein, F and Medioni, G** 'Toss – a system for efficient three dimensional object recognition', *Proc. DARPA Image Understanding Workshop*, Pittsburgh, PN (September 1990)
- 11 **Parvin, B and Medioni, G** 'A constraint satisfaction network for matching 3d objects', *Proc. Int. Conf. on Neural Networks* (vol II), Washington, DC (June 1989) pp 281–286
- 12 **Sato, K and Inokuchi, S** 'Range-imaging system utilizing nematic liquid crystal mask', *Proc. IEEE Int. Conf. on Comput. Vision*, London, UK (1987) pp 657–661
- 13 **Bolles, R C and Horaud, P** '3DPO: a three-dimensional part orientation system', *Int., J. Robotics Res.* Vol 5 No 3 (1986) pp 3–26
- 14 **Faugeras, O D, Ayache, N and Faverjon, B** 'A geometric matcher for recognizing and positioning 3-D rigid objects', *Proc. Conf. Artif. Intell. & Applic.*, Denver, CO (1984)
- 15 **Kamgar-Parsi, B, Jones, L and Rosenfeld, A** 'Registration of multiple overlapping range images: scenes without distinctive features', *Proc. Conf. on Comput. Vision & pattern Recogn.*, San Diego, CA (June 1989) pp 282–290
- 16 **Lowe, D G** *Perceptual Organization and Visual Recognition*, Kluwer Academic Publishers, New York (1985)
- 17 **Beaudet, P R** 'Rotationally invariant image operators', *Proc. Int. Joint Conf. on Pattern Recogn.*, Kyoto, Japan (1978) pp 579–583
- 18 **Medioni, G, Saint-Marc, P and Jezouin, J-L** 'A versatile pc-based range finding system', *IEEE Trans. Robotics & Automat.*, Vol 7 No 2 (April 1991) pp 250–256