Registration and Integration of Multiple Object Views for 3D Model Construction

Chitra Dorai, *Member, IEEE*, Gang Wang,
Anil K. Jain, *Fellow, IEEE*, and Carolyn Mercer

Abstract—Automatic 3D object model construction is important in applications ranging from manufacturing to entertainment, since CAD models of existing objects may be either unavailable or unusable. We describe a prototype system for automatically registering and integrating multiple views of objects from range data. The results can then be used to construct geometric models of the objects. New techniques for handling key problems such as robust estimation of transformations relating multiple views and seamless integration of registered data to form an unbroken surface have been proposed and implemented in the system. Experimental results on real surface data acquired using a digital interferometric sensor as well as a laser range scanner demonstrate the good performance of our system.

Index Terms—Automatic 3D object modeling, free-form objects, registration, view integration, range images, digital interferometry.

1 Introduction

THERE is growing interest in transforming images to models, i.e., to construct geometric and descriptive models of 3D objects from sensed data for applications ranging from part inspection and manufacturing to surgical planning and entertainment industries. Models of existing objects may be unavailable because

1) CAD techniques were not used to create the objects or
2) use of the original CAD models is prohibited due to proprietary reasons.

Accurate models of existing free-form objects are required in emerging applications such as object animation and visualization in virtual museums and vision augmented environments. Another industrial application that is of major interest to us is image tiling and assembly for automated inspection of in-situ engine components whose surface information is acquired from various viewpoints using optical phase-measuring interferometry systems [1].

Automatic object model construction involves:

1) acquiring many object views,
2) registering the views, and
3) integrating and building geometric models of the data.

Data acquisition involves obtaining either intensity or depth data of an object from multiple viewpoints. In this paper, the term data refers to surface depth measurements that can be obtained using a laser range scanner or an interferometric sensor. Since accurate 3D spatial relations between different object views may not be easily and directly obtained in many situations, during registration, transformations that relate the views need to be determined to bring the object regions that are shared between them into alignment. Integration merges data from multiple views using the computed view transformations such that a single surface representation is created in a unique coordinate frame.

In this paper, we describe a complete system for registering and integrating multiple views of an object. New techniques for solving key problems, such as robust estimation of transformations relating multiple views and seamless integration of registered data to form an unbroken surface, have been proposed and implemented in our system. Our technique [2] registers object views pairwise in the presence of uncertainties and noise in surface data, and allows data to be obtained from any viewpoint without the need for a control device with six degrees of freedom of motion. The view integration scheme uses a weighted averaging technique for merging the registered views together to result in a smooth surface. Our system has been tested on depth data obtained using digital interferometry techniques as well as laser range scanners. Experimental results on real complex object surfaces demonstrate the feasibility of our system.

2 Previous Work

Many current research efforts approach object model construction via view registration and integration. Techniques to solve the registration problem fall into two categories:

1) the first kind [3], [4], [5] relies on precisely calibrated data acquisition devices to determine the transformations that relate the views and
2) the second involves developing techniques to estimate the transformations from the data directly [6].

The first type is often inadequate to construct a complete description of complex shaped objects because views are restricted to rotations or to some known viewpoints only. Therefore, we cannot make use of the object surface geometry in the selection of observer viewpoints to obtain measurements. The second category is general and involves searching a huge parameter space, and even with good heuristics, it may be computationally very expensive.

Chen and Medioni avoid the search in the view transformation space by assuming an initial approximate transformation for registration, which is improved with an iterative algorithm [7] that minimizes the distance from points in a view to tangential planes at corresponding points in other views. Besl and McKay [8] propose an *iterative closest point* algorithm for registration which requires the specification of an appropriate procedure to find the closest point on a geometric entity to a given point. Registering sparse range data has also been proposed as a motion estimation problem [9], although the algorithm has not been tested on real surface data. The view integration technique described by Soucy and Laurendeau [10] assumes that reliable view transformations are available and proceeds by building a set of triangulations with the subsets of the common surface segments between all pairs of views, and connecting them to output a global triangulation. Martin Rutihauser et al. [11] also start with triangulated surfaces but employ a modified Kalman estimator to merge the triangulations robustly. Blais and Levine [12] employ a combination of reverse calibration and stochastic search to estimate the pairwise view transformation. While most techniques attempt to register views sequentially, Bergevin et al. [13], [14] and Shum et al. [15] propose to register multiple views simultaneously. A related system for registration and integration of object views [16] uses a variant of the iterated closest point algorithm [8]. Our registration technique also uses a distance minimization algorithm to register a pair of views, but we do not impose the requirement that one surface be strictly a subset of the other. Another example of a reverse engineering system is REFAB [17] that allows an interactive specification of the types and locations of features in a mechanical part and generates the parameterizations of features to be converted into a usable CAD model.

- *C. Dorai, G. Wang, and A.K. Jain are with the Department of Computer Science, Michigan State University, East Lansing, MI 48824. E-Mail: {dorai,wanggang,jain}@cps.msu.edu.*
- *C. Mercer is with the Optical Technology Branch, NASA Lewis Research Center, Cleveland, OH 44135. E-mail: cmercer@lerc.nasa.gov.*

Manuscript received 6 June 1996; revised 29 Aug. 1997. Recommended for acceptance by B.C. Vemuri.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 105853.

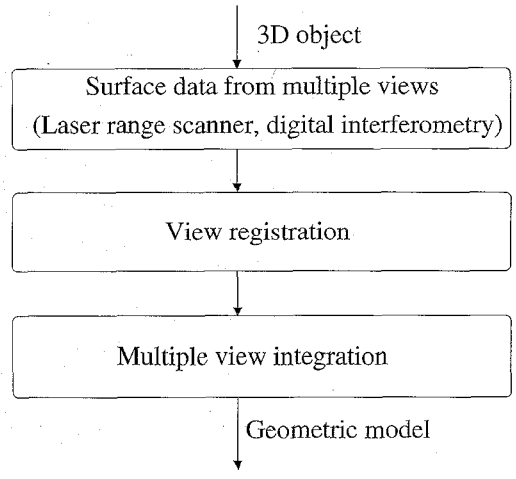


Fig. 1. Overview of an automatic 3D object model construction system.

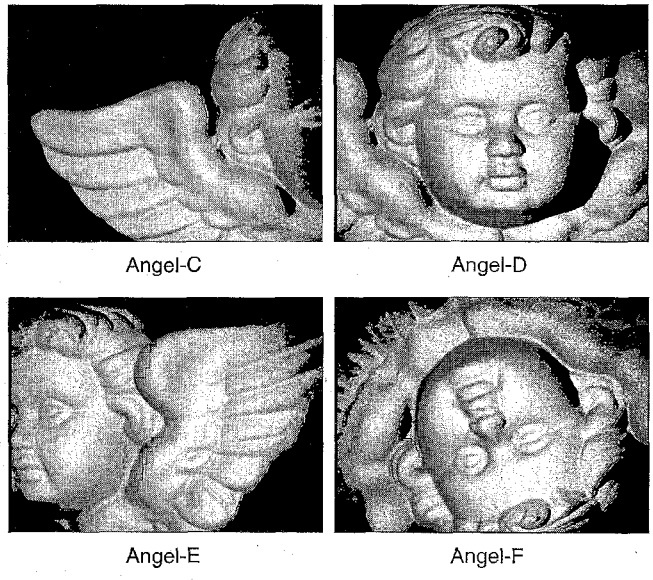


Fig. 2. Four different views of an angel sculpture obtained using a prototype fiber-optic projected fringe system designed and built at NASA Lewis Research Center.

3 System Overview

Our multiview integration system consists of several stages of processing as shown in Fig. 1. Details of each stage are presented in the following sections.

3.1 Data

Interferometric data was used in our study in addition to the data obtained used a laser range scanner because it is representative of data encountered in many industrial applications. Fig. 2 shows the data obtained from a fiberoptic projected-fringe system [1]. Surface data is rendered as pseudo intensity in Fig. 2 and points almost vertically oriented are shown darker. Projected fringe interferometers are characterized by the tradeoff between high accuracy and measurement area; the width of the measured area can only be up to 2,000 times larger than the depth resolution. Hence, for large objects, multiple views are required to get complete surface information.

3.2 Preprocessing

In order to improve the accuracy of registration and data integration, some preprocessing is required to remove the noise in the input images. We have implemented two preprocessing steps:

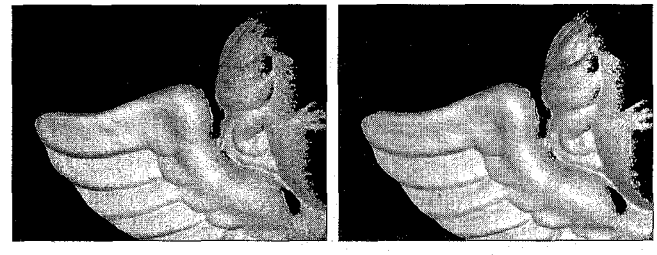


Fig. 3. Preprocessing of surface data. (a) Angel-C after removal of isolated pixels. (b) After median filtering.

1) removal of isolated pixels and
2) removal of noise through median filtering.

Experimental results indicate that a combination of these two preprocessing steps improves the accuracy of integration results noticeably.

The isolated pixel removing algorithm first segments the input range images into multiple *connected* regions and calculates the size of each *connected* region in units of pixels. It then removes all those isolated patches whose areas are less than a pre-defined threshold. For the angel images, the threshold was set to 10 percent of the number of pixels in each image. Fig. 3a shows the result of this processing step.

We adopted the median filtering technique for noise removal as it does not lose much surface detail while smoothing the input images. Experimental results show that applying a median filter to input range images after the removal of isolated pixels can remarkably improve the accuracy of image registration. The number of iterations required to achieve convergence during the iterative registration of views also decreases noticeably. Fig. 3b shows the result of a 3×3 median filter applied to the range image after the removal of isolated pixels. There was not much difference observed in the accuracy of the registration results when the window size of the filter was larger.

3.3 Registration of Views

Given a pair of range images P and Q (different views of the same 3D object), the goal of a registration algorithm is to find the best rigid transformation $\mathcal{T}$ that relates P and Q. These two views are said to be *registered* if the relation $\mathcal{T}\mathbf{p} = \mathbf{q}$ holds for any pair of points $(\mathbf{p}, \mathbf{q})$, $\mathbf{p} \in P$, $\mathbf{q} \in Q$, where $\mathbf{p}$ and $\mathbf{q}$ represent the same surface point. View P is referred to as the *source* image, and view Q, the *destination* image. An enhanced adaptation of the iterative distance minimization technique proposed by Chen and Medioni [7] is employed in our system to register a given pair of object views from their surface data directly.

3.3.1 Pairwise View Registration

The approach of Chen and Medioni [7] is based on the assumption that an approximate transformation between two views is already known, i.e., data from the views are approximately registered, and the goal is to refine the initial estimate to obtain a more accurate global registration. Nearly all distance minimization algorithms suppose that a coarse initial estimate of the transformation is either available from the knowledge of positions of the sensor or estimated using other feature-based techniques. Given two object views P and Q and a set of N points called the *control points* in P, Chen and Medioni [7] used the following objective function to estimate the view transformation $\mathcal{T} = (\alpha, \beta, \gamma, t_x, t_y, t_z)$ by iteratively minimizing the distances from the control points to the surface Q:

$$e^k = \sum_{i=1}^{N} d_s^2\left(\mathcal{T}^k \mathbf{p}_i, S_i^k\right) \qquad N \geq 3, \tag{1}$$

where $\mathcal{T}^k$ is the 3D transformation applied to a control point $\mathbf{p}_i \in P$ at the kth iteration, $l_i = \left\{\mathbf{a} \mid (\mathbf{p}_i - \mathbf{a}) \times \mathbf{n}_{\mathbf{p}_i} = 0\right\}$ is the line normal to

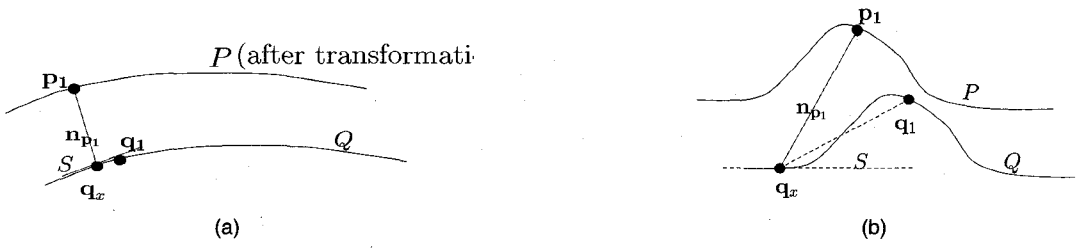


Fig. 4. Correspondence of control points. (a) Smooth surface. (b) Rough surface.

P at $\mathbf{p}_i$, $\mathbf{q}_i^k = (\mathcal{T}^k l_i) \cap Q$ is the intersection point of surface Q with the transformed line $\mathcal{T}^k l_i$, $\mathbf{n}_{\mathbf{q}_i}^k$ is the normal to Q at $\mathbf{q}_i^k$, $S_i^k = \left\{\mathbf{s} \mid \mathbf{n}_{\mathbf{q}_i}^k \cdot (\mathbf{q}_i^k - \mathbf{s}) = 0\right\}$ (where '$\cdot$' stands for the scalar product and '$\times$' for the vector cross product.) is the tangent plane to Q at $\mathbf{q}_i^k$ and d_s is the signed distance from a point to a plane.

The registration algorithm thus finds a transformation $\mathcal{T}$ that iteratively minimizes e^k using a least squares method. The tangent plane S_i^k serves as a local linear approximation to the surface Q at point $\mathbf{q}_i$. The intersection point $\mathbf{q}_i^k$ is an approximation to the actual *corresponding* point $\mathbf{q}_i$ which is unknown at each iteration k. An initial transformation $\mathcal{T}^0$ that approximately registers the two views is used to start the iterative process. By minimizing the distance from a point to a plane, only the direction along which the distance can be reduced is constrained. The convergence of the process can be established by verifying that the difference between the errors e^k at any two consecutive iterations is less than a prespecified threshold. The point of intersection of the normal line l_i and the surface Q is determined using iterative search near the neighborhood of a prospective intersection point computed using a reverse calibration procedure [2].

3.3.2 Verification of Control Point Correspondences

Given a set of control points in the *source* image, finding their *corresponding* control points accurately in the *destination* image is one of the core tasks of the registration algorithm. The effectiveness of the algorithm for establishing the correspondence depends upon the following two factors:

1) the accuracy of the initial transformation that drives the iterative process and
2) the smoothness of the surfaces to be registered.

If the initial estimate is accurate and the object surfaces are smooth, most corresponding control points found by our technique are reasonable approximations to the true corresponding control points (see Fig. 4a). However, if the initial guess is not accurate, or the surface of the object is noisy and rough, many corresponding control points found may be far away from the actual corresponding control points (see Fig. 4b). In both the cases, $\mathbf{n}_{\mathbf{p}_1}$ is the surface normal to the surface P at control point $\mathbf{p}_1$ and $\mathbf{q}_x$ is the intersection of $\mathbf{n}_{\mathbf{p}_1}$ with the surface Q. Therefore, $\mathbf{q}_x$ would be considered the *corresponding control point* of $\mathbf{p}_1$. S is the tangential plane of Q estimated at the point $\mathbf{q}_x$ and $\mathbf{q}_1$ is the true corresponding point to $\mathbf{p}_1$ on the surface Q. The distance between $\mathbf{q}_1$ and $\mathbf{p}_1$ is $d_s(\mathcal{T}^k \mathbf{p}_i, S_i^k)$ desired in the registration algorithm described in Section 3.3.1, and would be minimized by the iterative process. On a smooth surface, the approximate corresponding point, $\mathbf{q}_x$ and the true corresponding point, $\mathbf{q}_1$ can lie closer to each other as shown in Fig. 4a whereas if the surface is noisy, they may be far away from one another as shown in Fig. 4b. Any inaccuracy in establishing the correspondence of control points leads to inaccurate estimation of d_s and hence the transformation parameters. We have designed a verification mechanism to check the validity of the corresponding control points found by the algorithm and in our system, only valid corresponding control point pairs are employed to update the transformation matrix.

The correctness of the correspondences of control points can be verified by the distance constraint imposed by rigid transformations. Given control points $\mathbf{p}_1$ and $\mathbf{p}_2$ on the surface P, and the corresponding control points $\mathbf{q}_1$ and $\mathbf{q}_2$ on the surface Q, the constraint

$$\| \mathbf{p}_1 - \mathbf{p}_2 \| = \| \mathbf{q}_1 - \mathbf{q}_2 \|$$

holds for all rigid transformations. Otherwise, $(\mathbf{p}_1, \mathbf{q}_1)$ and $(\mathbf{p}_2, \mathbf{q}_2)$ cannot be valid control point pairs at the same time and thus, $(\mathbf{p}_1, \mathbf{q}_1)$ and $(\mathbf{p}_2, \mathbf{q}_2)$ are not compatible with each other. Two pairs of control points $(\mathbf{p}_1, \mathbf{q}_1)$ and $(\mathbf{p}_2, \mathbf{q}_2)$ are defined to be compatible if and only if

$$-t \le \frac{\|\mathbf{p}_1 - \mathbf{p}_2\| - \|\mathbf{q}_1 - \mathbf{q}_2\|}{\max(\|\mathbf{p}_1 - \mathbf{p}_2\|, \|\mathbf{q}_1 - \mathbf{q}_2\|)} \le t \qquad (2)$$

where t is a predefined threshold (10 percent in our implementation).

Given two object views (surfaces) P and Q and a control point set $\{\mathbf{p}_1, \mathbf{p}_2, \cdots \mathbf{p}_n\}$ where $\mathbf{p}_i$ lies on P, the "approximate" corresponding control point set $\{\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_n\}$ is first determined at each iteration. The following method is used to check the correspondence of control point pairs $(\mathbf{p}_i, \mathbf{q}_i)$ for each $i \in \{1, 2, \cdots n\}$, and those found to be invalid are removed:

1) For any $i, j \in \{1, 2, \cdots, n\}$, and $i \neq j$, check whether the control point pairs $(\mathbf{p}_i, \mathbf{q}_i)$, $(\mathbf{p}_j, \mathbf{q}_j)$ are compatible with each other based on (2).
2) Remove the control point pair $(\mathbf{p}_i, \mathbf{q}_i)$ that is incompatible with the most number of control point pairs.
3) Iterate steps (1) and (2) until
   i) all the remaining control point pairs are compatible with each other, or
   ii) up to a preset maximum (50 percent of n) number of pairs is removed, or
   iii) the number of remaining control points is less than a predefined threshold (20 in our system).

Compared with the technique of Chen and Medioni [7], the improved algorithm augmented by the control point verification step was found to provide more accurate results as demonstrated by our experiments.

3.4 Seamless Integration of a Pair of Views

Given a pair of registered views, one of the goals in designing an integration algorithm is that the assembled data should be as smooth as possible without any loss of details present in the original raw data. Given the fact that there can be some imprecision in the estimated view transformation parameters obtained in the registration stage, and that the input data may be noisy, it is not

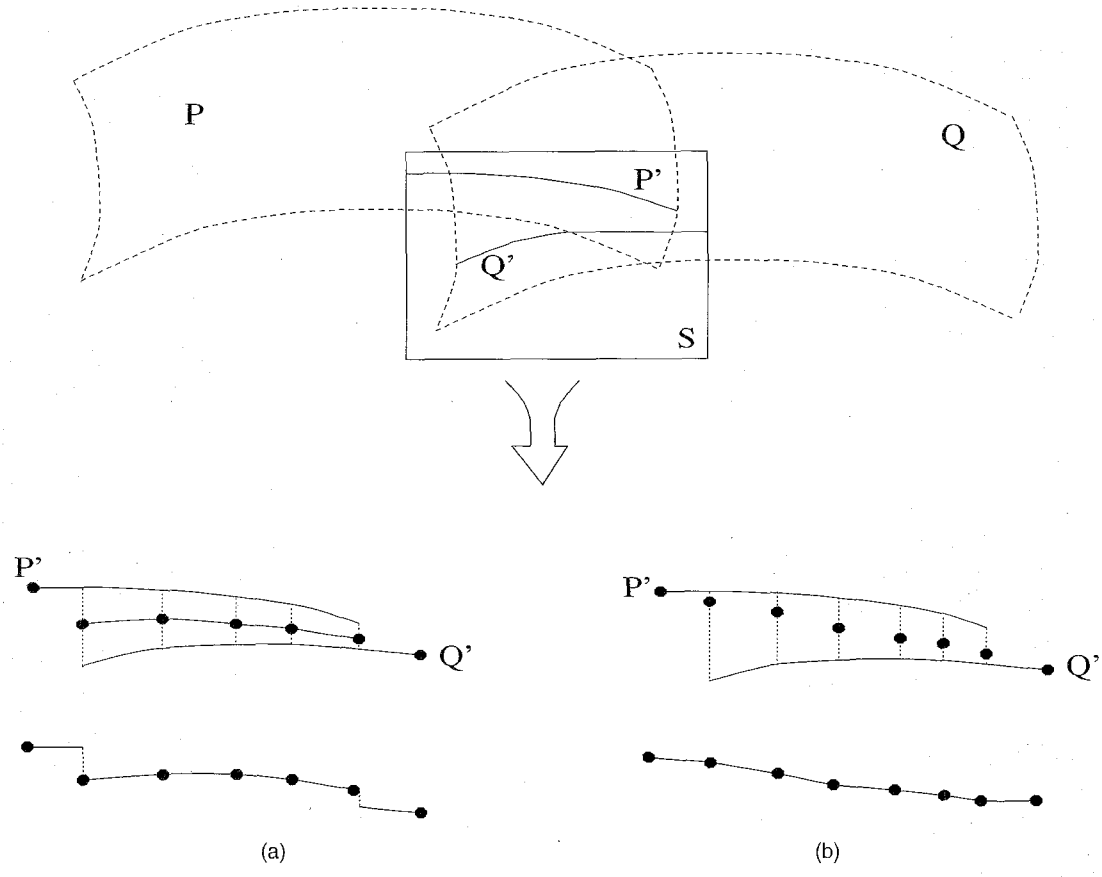


Fig. 5. Comparison of integration algorithms. (a) Integration using simple averaging. (b) Integration using weighted averaging.

quite easy to achieve this goal. Although a smoothing algorithm appears to be necessary to smooth the integrated data, surface details may be lost during blurring.

We have proposed and implemented a weighted averaging algorithm to integrate two registered range images. In order to compare the effectiveness of the weighted averaging algorithm, a simple (unweighted) averaging algorithm was also implemented. Fig. 5 illustrates both the integration algorithms. In Fig. 5, only a part of the surfaces P and Q that are to be integrated is overlapping. Plane S is an arbitrary plane that is parallel to the Z axis. Curves P' and Q' are the intersections of surfaces P and Q with plane S, respectively.

With the simple averaging algorithm, the depth value of each pixel in the merged image is calculated by averaging the depth values of the corresponding pixels in the *source* image P and the *destination* image Q, as shown in Fig. 5a. The drawback of this method is that it cannot guarantee the smoothness of the depth values on the view boundary. Fig. 7a shows the resultant image derived from assembling Angel-C and Angel-D by using the simple averaging algorithm. It demonstrates that the simple averaging algorithm cannot result in smooth integrated views.

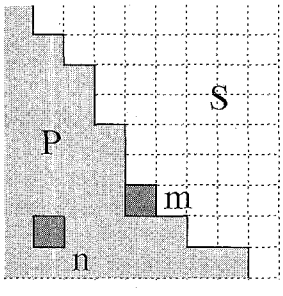


Fig. 6. Weighted averaging method.

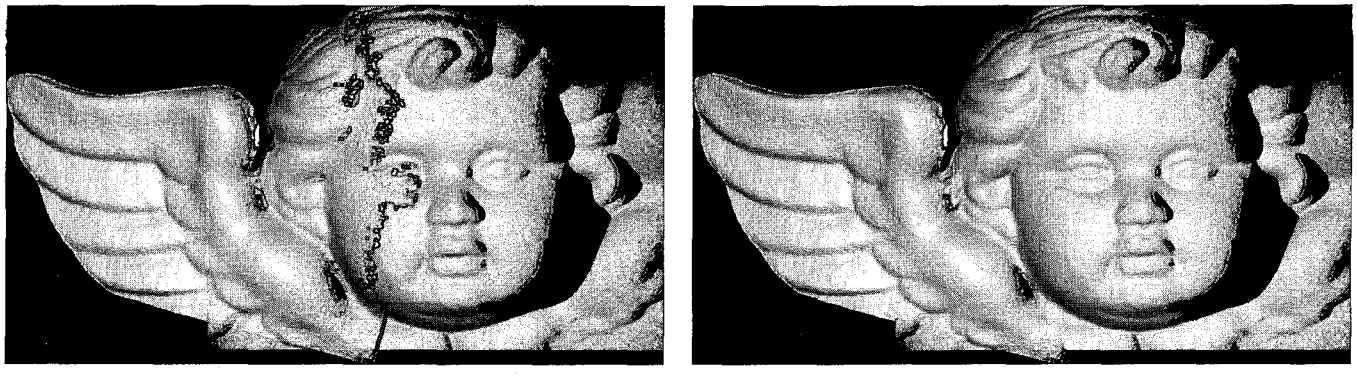
The weighted averaging algorithm, on the other hand is devised to deal with this problem. Each pixel in the input image is associated with a weight, which is essentially the distance of the pixel to the nearest boundary. As shown in Fig. 5b, a smooth transformation of data can be achieved on the view boundary by calculating the depth value in the merged image using the weighted averaging algorithm. Fig. 6 provides the details of the weighted averaging algorithm. Pixel m lies on the boundary of object P and S is the background. In our implementation, the weight associated with m is set to be 1. The nearest boundary pixel to pixel n in Fig. 6 is m, which is 3 pixels to the right and 1 pixel above n. Then, the weight of n is computed as $1 + 3 = 4$. Note that the weights associated with all the background pixels are defined to be 0. For any pixel x that is part of the visible surface of the object, its weight $W(x)$ is recursively defined as:

$$W(x) = \min_{\forall a \in N(x)} \big(W(a) + 1\big), \tag{3}$$

where the function $N(x)$ returns the neighborhood of pixel x. In our experiments both four-neighborhood and eight-neighborhood schemes were tested. Since there was no noticeable difference found in the integration results, only those results obtained using the four-neighborhood based scheme are reported in this paper.

Once the weight associated with each pixel in the overlapping regions of both the *source* and *destination* images is determined, the depth value Z_q of any pixel q in the integrated image is calculated based on the equation:

$$Z_q = \frac{Z_s \times W(s) + Z_d \times W(d)}{W(s) + W(d)}, \tag{4}$$



(a) (b)

Fig. 7. Integration of Angel-C and Angel-D. (a) Result of simple averaging. (b) Result of weighted averaging.

where s and d are the corresponding pixels of q in the *source* and *destination* images, respectively. Fig. 7b shows a marked improvement in the result of our weighted averaging algorithm over that of the simple averaging algorithm shown in Fig. 7a. The merger of views, Angel-E and Angel-F obtained using the weighted integration scheme is shown in Fig. 8.

3.5 Multiple View Integration

We have also devised a technique to assemble a simple 3D model of an object from its multiple views by integrating the information present in all the views so that new object views can be generated from this 3D model. Given multiple views $\{S_1, S_2, \cdots S_n\}$ and the corresponding transformation matrices $\{\mathcal{T}_1, \mathcal{T}_2, \cdots \mathcal{T}_n\}$ that relate these views to one another, a 3D model can be constructed as a 3D point set in the following manner:

$$M_1 = \bigcup_{i=1}^{n} (\mathcal{T}_i \circ S_i) \tag{5}$$

in which S_i is the set of 3D points obtained from the range image of the ith view. However, this method has a severe drawback. Consider points $\{\mathbf{p}_{i_1}, \mathbf{p}_{i_2}, \cdots \mathbf{p}_{i_k}\}$, $\mathbf{p}_{i_j} \in S_j$, $j = 1, 2, \cdots k$ that represent the same point $\mathbf{p}_i$ on the object surface visible in multiple views. These points after the transformation, may not map to a single surface location since the transformation matrix $\mathcal{T}_i$ estimated at the registration stage may not be very accurate. In other words, points $\mathcal{T}_j \circ \mathbf{p}_{i_j}$, $j = 1, 2, \cdots k$ may not reside exactly at the same 3D location. Therefore, inconsistencies may be introduced in the resultant surface model. To avoid this problem, we decompose the Cartesian 3D space into multiple equal-sized voxels $\{V_{ijk}\}$. For a given voxel V_{ijk}, we obtain a single 3D point $\mathbf{p}_{ijk}$ that represents all the points that fall into V_{ijk}:

$$\mathbf{p}_{ijk} = \frac{\sum_{\forall \mathbf{p} \in S\mathbf{p}_{ijk}} W(\mathbf{p}) \times \mathbf{p}}{\sum_{\forall \mathbf{p} \in S\mathbf{p}_{ijk}} W(\mathbf{p})} \tag{6}$$

where $S\mathbf{p}_{ijk} = \{\mathbf{p} \mid \mathbf{p} \in M_1, \mathbf{p} \in V_{ijk}\}$ and $W(\mathbf{p})$ is the weight associated with point $\mathbf{p}$. Note that each point (pixel) in an input range image is associated with a weight, which is essentially the distance of the pixel to the nearest boundary. The resultant surface model is $M_w = \{\mathbf{p}_{ijk}\}$.

Once the 3D model M_w is obtained, a user can visualize the object from various new viewpoints. For a given viewpoint, there exists an equivalent transformation $R = \{\alpha, \beta, \gamma, t_x, t_y, t_z\}$, based on which a rigid transformation matrix $\mathcal{T}$ can be computed. The visible point subset $V_{set} \subseteq \mathcal{T} \circ M_w$ can be determined based on the depth value of each point in $\mathcal{T} \circ M_w$.

4 Experimental Results

We tested the performance of our prototype system on surface data obtained using a Technical Arts White laser range scanner available at Michigan State University and a digital interferometric sensor available at NASA Lewis Research Center. We registered and integrated the angel images shown in Fig. 2 successfully using the pairwise integration scheme to obtain the final integrated view shown in Fig. 9.

In order to compare the registration accuracy of our algorithm with the registration algorithm [7] proposed by Chen and Medioni, we used the same input images for both the algorithms. Table 1 shows the registration results in which Angel-C and Angel-D were used as the *source* and *destination* images, respectively; the initial guess was set to ($\alpha = 0$, $\beta = 0$, $\gamma = -15$, $t_x = -3.5$, $t_y = 1$, $t_z = -0.5$) by

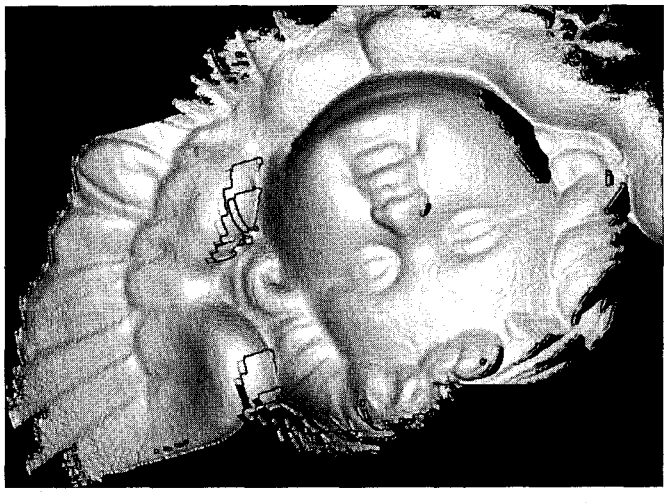


Fig. 8. Integrated view of Angel-E and Angel-F.

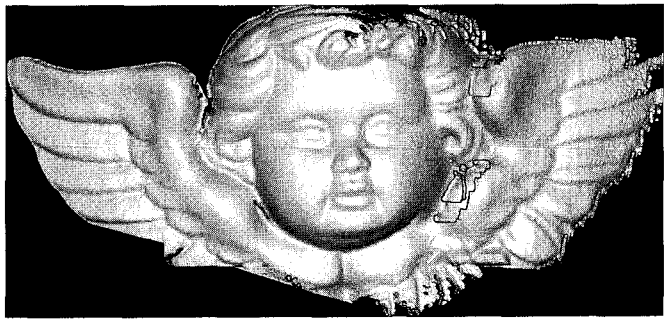


Fig. 9. Result of merging all angel views.

TABLE 1
REGISTRATION RESULTS

Registration method	*with* preprocessing and control point verification	*without* preprocessing and control point verification
3D transformation parameters	$\alpha = 3.692$ $\beta = -1.7056$ $\gamma = -14.5857$ $t_x = -3.3472$ $t_y = 1.2833$ $t_z = -0.5704$	$\alpha = 3.6922$ $\beta = -1.0148$ $\gamma = -14.8978$ $t_x = -3.3511$ $t_y = 1.3246$ $t_z = -0.5207$
Registration error, δ_r	0.0374	0.0515

visually inspecting the two views. When the range images contain the entire object surface and the rotations of the object in the views are primarily in the plane, then a good initial guess for the iterative algorithm can be determined automatically. We compute the approximate rotation and translation from the major (principal) axes of the object from the range data [2] to serve as an initial guess for our iterative registration procedure, since we assume that we do not have the prior knowledge of the sensor placement. If computational time is not a constraint on the system's performance, a reasonable estimate of the initial transformation can be obtained using genetic algorithms or simulated annealing techniques.

All rotations shown in Table 1 are measured in degrees and all translations as well as the registration errors are in centimeters. Our iterative registration scheme reached convergence after four iterations. With the Chen and Medioni method, the results improved gradually during the first five iterations; however, the result of the sixth iteration became worse than that of the fifth. This oscillatory behavior is often observed when the iterative procedure is near the final convergence. Table 1 shows only the result of the fifth iteration of Chen and Medioni's method. By using the verification of control point correspondences (Section 3.3.2), our system avoids the oscillation of values of the transformation parameters.

In all our experiments, we used registration error (δ_r) as an indication of the accuracy of registration. It was computed as the average distance from the surface of the *source* image to the *destination* image in the direction of the Z axis, once the *source* image has been transformed using the transformation parameters determined by the registration process. In most cases, only parts of the *source* and *destination* images are overlapped. Therefore, δ_r is calculated only in the overlapping regions. The value of δ_r is primarily determined by two factors:

1) the accuracy of registration error and
2) the noise in the input images.

Although it is difficult to quantify the contribution of each factor to the final value of δ_r, it can still be used to compare the relative merits of different registration techniques, since the noise in the input images is the same. In the case of angel images, the distance between the two eyes is about 1.2 cm (140 pixels) and the range image of Angel-D spans about 1.4 cm of depth. The error in registering Angel-C and Angel-D is about 3 percent of the depth span (about 4 pixels). Results from comparisons using images of different objects indicate that accurate registration results from our improved registration method lead to better integration of data especially when the input images are complex and noisy. These tests also show that our method requires a fewer number of iterations to achieve convergence. Given a coarse correct initial guess, our system usually takes about 30 seconds to register two range images whose sizes are 640×480 on a SPARCstation 10 with 32MB RAM.

By registering and integrating nine different views of a wooden Buddha sculpture shown in Fig. 10, a 3D surface model was constructed which allows a user to rotate and view the Buddha from new viewing directions (see Fig. 11) which are distinct from those that were used to generate the input data. The thickness of the laser beam generated by the range scanner is about 3mm. As a result, many of the fine details on the head of the Buddha were lost or were not measured accurately while acquiring the data using the Technical Art White scanner. This loss of detail cannot be properly compensated during the construction of the 3D surface model. Thus, on the resultant 3D surface model, there are a few small "holes" (black pixels) on the Buddha's head. The input views contain altogether 142,448 3D surface points and each pixel in an image represents an actual size of $0.026'' \times 0.045''$. The voxel size selected was 0.04" during view integration. The resultant surface model contains 47,720 3D points. These experimental results demonstrate that our system is capable of seamlessly merging surface information from multiple range images. We believe that our system is of practical value for many applications due to its accuracy and robustness.

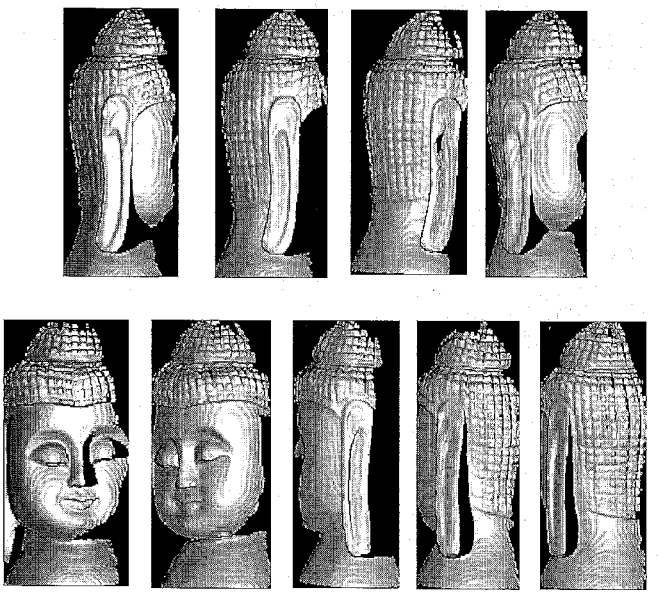


Fig. 10. Nine views used to reconstruct a 3D surface model of the Buddha.

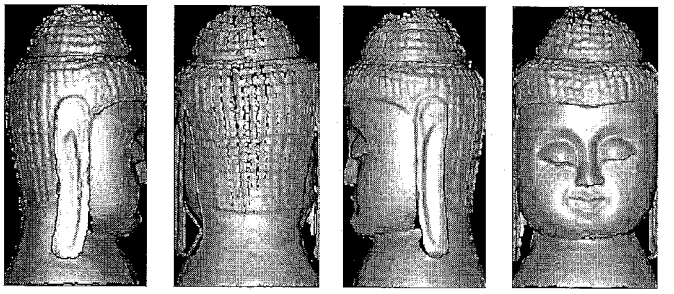


Fig. 11. Four new views generated using the reconstructed 3D model of the Buddha.

5 DISCUSSION

When multiple object views are registered and integrated in stages, the entire 3D model can be constructed easily. However, this approach may lead to accumulated registration errors. A possible way to avoid this accumulated registration error is to register and integrate all the views *simultaneously* using a star network [13], [14] supplemented by our improved registration technique.

During view registration, the extent of surface overlap directly influences the number of corresponding points that can be established for the control points and thus the accuracy of the estimated transformation. With very small overlapping regions in the views registered, our system could only retain a few corresponding pairs that were used in estimating the transformation. If acquiring data is not a problem, many closely overlapping views can be obtained and the transformation accuracy can be improved. Our system is not limited to any fixed number of views; however computational costs will place practical limits.

The voxel size employed during integration was approximately that of the smallest pixel. We have not experimented with the effect of varying grid location or voxel size; we anticipate that raising this size will result in over-smoothing of the surface, thus losing details of surface discontinuities, and lowering the voxel size will result in the retention of too many noisy surface points. We have tested our registration algorithm using simulated noise with standard deviation ranging from 0.001 to 0.005 inch, which realistically models the error in z introduced by a Technical Arts White scanner that was employed to obtain the range data for our experiments. The performance of the algorithm was satisfactory under these conditions.

6 Summary

We have proposed and implemented a complete prototype system for registering and integrating multiple 3D object views from range data which can then be used to construct geometric models of the objects. We have employed an improved registration technique to compute the view transformation relating a pair of views. Our experimental results illustrate that by combining preprocessing techniques such as removal of isolated pixels and median filtering, and by enhancing the performance of the registration algorithm with the verification of control point correspondences, the accuracy of view registration results can be improved noticeably. A weighted integration algorithm was proposed to result in smoothly merged surfaces. We have demonstrated that it is feasible to integrate surface data of any complex shaped 3D object from its multiple views accurately.

Acknowledgments

We thank the reviewers for their thoughtful suggestions for improvement.

References

[1] C.R. Mercer and G. Beheim, "Fiber-Optic Projected-Fringe Digital Interferometry," *Hologram Interferometry and Speckle Metrology Proc.*, pp. 210-216, Bethel, Conn., Society for Experimental Mechanics, 1990.

[2] C. Dorai, J. Weng, and A. Jain, "Optimal Registration of Multiple Range Views," Tech. Rep. CPS94-36, Dept. of Computer Science, Michigan State Univ., July 1994 (to appear *IEEE Trans. Pattern Analysis and Machine Intelligence*).

[3] B. Bhanu, "Representation and Shape Matching of 3-D Objects," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6, no. 3, pp. 340-351, May 1984.

[4] N. Ahuja and J. Veenstra, "Generating Octrees From Object Silhouettes in Orthographic Views," *IEEE Trans. Pattern Analysis and Machine Intelligence.*, vol. 11, no. 2, pp. 137-149, Feb. 1989.

[5] B.C. Vemuri and J.K. Aggarwal, "3-D Model Construction From Multiple Views Using Range and Intensity Data," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 435-437, Miami Beach, Fla., 1986.

[6] M. Potmesil, "Generating Models for Solid Objects by Matching 3D Surface Segments," *Proc. Int'l Joint Conf. Artificial Intelligence*, pp. 1,089-1,093, Karlsruhe, Germany, 1983.

[7] Y. Chen and G. Medioni, "Object Modelling by Registration of Multiple Range Images," *Image and Vision Computing*, vol. 10, pp. 145-155, Apr. 1992.

[8] P.J. Besl and N.D. McKay, "A Method for Registration of 3-D Shapes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239-256, Feb. 1992.

[9] R. Szeliski, "Estimating Motion From Sparse Range Data Without Correspondence," *Proc. Second Int'l Conf. Computer Vision*, pp. 207-216, Tampa, Fla., Dec. 1988.

[10] M. Soucy and D. Laurendeau, "A General Surface Approach to the Integration of a Set of Range Views," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 4, pp. 344-358, Apr. 1995.

[11] M. Rutishauser, M. Stricker, and M. Trobina, "Merging Range Images of Arbitrarily Shaped Objects," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 573-580, Seattle, Wash., June 1994.

[12] G. Blais and M.D. Levine, "Registering Multiview Range Data to Create 3D Computer Graphics," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 820-824, Aug. 1995.

[13] R. Bergevin, D. Laurendeau, and D. Poussart, "Registering Range Views of Multipart Objects," *Computer Vision and Image Understanding*, vol. 61, pp. 1-16, Jan. 1995.

[14] R. Bergevin, M. Soucy, H. Gagnon, and D. Laurendeau, "Towards a General Multi-View Registration Technique," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, pp. 540-547, May 1996.

[15] H.-Y. Shum, K. Ikeuchi, and R. Reddy, "Principal Component Analysis With Missing Data and its Application to Object Modeling," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 560-565, Seattle, Wash., June 1994.

[16] G. Turk and M. Levoy, "Zippered Polygon Meshes From Range Images," *SIGGRAPH 94*, pp. 311-318, 1994.

[17] W.B. Thompson, J.C. Owen, and H.J. de St. Germain, "Feature-Based Reverse Engineering of Mechanical Parts," Tech. Rep. UUCS-95-010, Dept. of Computer Science, Univ. of Utah, Salt Lake City, Utah, 1995.