

3D Modeling of Optically Challenging Objects

Johnny Park, *Member, IEEE*, and Avinash C. Kak

Abstract—We present a system for constructing 3D models of real-world objects with optically challenging surfaces. The system utilizes a new range imaging concept called multippeak range imaging, which stores multiple candidates of range measurements for each point on the object surface. The multiple measurements include the erroneous range data caused by various surface properties that are not ideal for structured-light range sensing. False measurements generated by spurious reflections are eliminated by applying a series of constraint tests. The constraint tests based on local surface and local sensor visibility are applied first to individual range images. The constraint tests based on global consistency of coordinates and visibility are then applied to all range images acquired from different viewpoints. We show the effectiveness of our method by constructing 3D models of five different optically challenging objects. To evaluate the performance of the constraint tests and to examine the effects of the parameters used in the constraint tests, we acquired the ground-truth data by painting those objects to suppress the surface-related properties that cause difficulties in range sensing. Experimental results indicate that our method significantly improves upon the traditional methods for constructing reliable 3D models of optically challenging objects.

Index Terms—Range data, geometric modeling, image analysis, virtual reality, feature representation.

1 INTRODUCTION

IN the last few decades, constructing accurate 3D models of real-world objects has drawn much attention from many industrial and research groups. Earlier, the 3D models were used primarily in robotics and computer vision applications such as bin picking and object recognition. The models for such applications only require salient geometric features of the objects so that the objects can be recognized, and the pose can be determined. Therefore, it is unnecessary in these applications for the models to faithfully capture every detail on the object surface. More recently, however, there has been considerable interest in the construction of 3D models for applications where the focus is more on visualization of the object by humans [1], [2], [3], [4], [5], [6], [7], [8], [9]. This interest is fueled by the recent technological advances in range sensors and the rapid increase of computing power that now enables a computer to represent an object surface by millions of polygons or points and render such representations in real time. Obviously, in order to take advantage of these technological advances, the 3D models constructed must capture to the maximum extent possible the shape and surface-texture information of real-world objects.

The classical approach for constructing 3D models of real-world objects involves a range sensor that acquires 3D measurements (that is, range image) of an object. Optical-triangulation-based structured-light scanners are commonly used due to their superior accuracy and relatively simple hardware requirement. A sketch of a typical structured-light scanner is shown in Fig. 1. A plane of light, typically a laser, parallel to the yz -plane of the world frame is projected onto the object of interest. The intersection of the light and the object creates a stripe of illuminated points on

the object surface. A camera captures the scene, and the center of illuminated points in each camera scan line is extracted. Given the image coordinates of the extracted illuminated points, the 3D coordinates with respect to the world frame can be computed by the equations based on triangulation. In order to scan the whole scene, the object is either translated along the x -axis or rotated about the z -axis through the plane of light, as the camera takes a sequence of images. A single scan produces a range image that captures only part of the object. Therefore, multiple range images are acquired from different viewpoints in order to capture the entire surface of the object.

Researchers have shown that the state-of-the-art techniques can now construct detailed 3D models of objects ranging from small figurines to large statues. Although they have established the feasibility of constructing accurate 3D models, there still remain several challenging issues. One of these challenging issues arises from the fact that many real-world objects have surface properties that are not ideal for range sensors. For example, consider the images shown in Fig. 2. The images in the top row show the photographs of objects with different surface materials, and the images in the bottom show the camera images of those objects when a laser stripe is projected onto the surface. The camera image for the first object (painted bust) corresponds to an ideal case, where the reflection of the projected laser stripe can be clearly detected for each camera scan line (that is, horizontal line) of the image. For the second object (metal bowl), however, multiple illuminated points in the same camera scan line exist due to the mutual reflections that occurred on the object's specular surface. Mutual reflections give rise to ghosts in the measured structured-light data, and depending on the extent of specularity, the presence of these ghosts can make it difficult to localize a data point that corresponds to the object point that was actually illuminated. Choosing the peak with the highest intensity value in a camera scan line does not guarantee that it corresponds to the object surface that was illuminated directly by the light source. For the third object (gorilla), the surface absorbed most of the projected light, causing the reflections captured in the camera image to have relatively low intensity values, which makes it difficult to disambiguate the primary

- The authors are with the School of Electrical and Computer Engineering, Purdue University, Electrical Engineering Building, 465 Northwestern Ave., West Lafayette, IN 47907-2035. E-mail: {jpark, kak}@purdue.edu.

Manuscript received 30 May 2006; revised 13 Dec. 2006; accepted 27 Mar. 2007; published online 29 May 2007.

Recommended for acceptance by H.-P. Seidel.

For information on obtaining reprints of this article, please send e-mail to: tcvg@computer.org, and reference IEEECS Log Number TVCG-0074-0506. Digital Object Identifier no. 10.1109/TVCG.2007.1069.

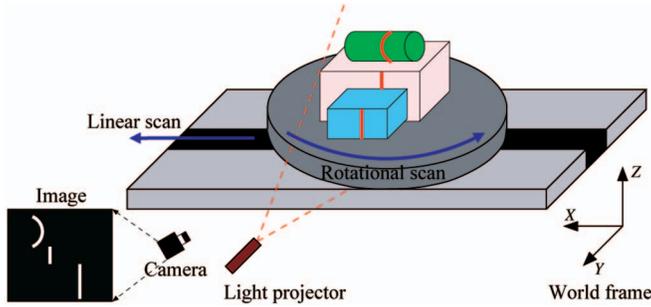


Fig. 1. A typical optical-triangulation-based structured-light scanner.

reflections from other spurious reflections in the camera image. Some researchers have tried to simply do away with such surface-related problems by painting or coating the object with removable powder to ensure that the surfaces reflect the laser in a diffuse manner. Obviously, this approach would not be desirable and may not even be feasible for many applications outside the laboratory.

In this paper, we present a system for constructing 3D models of real-world objects that is capable of handling optically challenging surfaces. The system utilizes a new range imaging concept called multippeak range imaging, which stores multiple candidates of range measurements for each point on the object surface. The multiple measurements include the erroneous structured-light data caused by various surface properties that are not ideal for structured-light range sensing. False measurements generated by spurious structured-light reflections are iteratively eliminated by applying a series of constraint tests. After convergence of the constraint tests, the remaining data are integrated to form a final geometric 3D model.

The remainder of this paper is organized as follows: We will first review some of the previous work in two related areas: construction of 3D models by using range data and range data acquisition of optically challenging objects. In Section 3, we will outline our 3D modeling system, and in Section 4, we will describe the concept of multippeak range imaging. Section 5 includes detailed descriptions of each step involved in the 3D modeling process, in particular all the constraint tests for eliminating false range measurements. We will then show our experimental results in Section 6 and, finally, the conclusion and future work in Section 7.

2 PREVIOUS WORK

2.1 Construction of 3D Models Using Range Data

Much work has been devoted to the area of constructing 3D models of real-world objects using range data [1], [2], [3], [4], [6], [7], [8], [10], [11]. In general, there are three main steps involved in the process. The first step is to acquire the geometric shape of the exterior of an object. Among many existing techniques for acquiring 3D data, optical triangulation sensors [12] are most commonly used. Typically, multiple range images viewed from different directions are required—reaching up to hundreds of range images, depending on the shape complexity and the size of object—so that all of the surface detail is captured.

The second step is the registration of multiple range images into a common coordinate system. The Iterative Closest Point (ICP) algorithm [13] has now become a dominant technique for registering data sets. The algorithm selects the closest

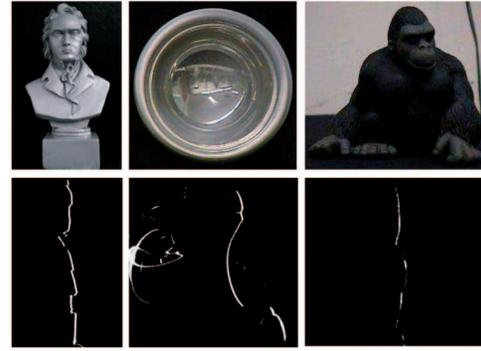


Fig. 2. The images in the bottom row show the camera images of the objects in the top row when a laser stripe is projected onto the surface.

points between two data sets as the corresponding points and computes a rigid transformation that minimizes the distances between the corresponding points. The data set is updated by applying the transformation, and the process is continued until the error between the corresponding points falls below a threshold. Since the introduction of ICP, many variants of ICP [14], [15] have been proposed to improve its performance. Since the ICP algorithm only deals with pairwise registration, Bergevin et al. [16] developed a multiview registration technique that minimizes the registration errors between all pairs of range images. Benjema and Schmitt [17] accelerated the multiview registration method of Bergevin et al., and Pulli [2] extended it so that it can be applied to large data sets.

All of the registration methods mentioned above require the initial data sets to be approximately aligned, which traditionally has been provided by either specialized hardware that records the sensor position or simply by a manual intervention. In the effort to automate the entire registration process without any special hardware, Huber and Hebert [8] proposed a method that automatically registers a set of range images. The method constructs a globally consistent solution from a list of pairwise registration results based on visibility consistency. Stamos and Leordeanu [18] utilize range segmentation and line features for automatic pairwise registrations and use a topological graph to form a globally consistent coordinate frame. Recently, Gelfand et al. [19] proposed an algorithm for automatic pairwise registration using a point descriptor based on the local surface geometry.

The registered range images taken from adjacent view-points will typically contain overlapping surfaces with common features in the areas of overlap. The third step consists of integrating the registered range images into a single connected surface model to eliminate the redundancies. The volumetric method [1], [20], which is one of the more widely used integration methods, computes an implicit function $d(x)$ that represents the closest distance from an arbitrary 3D point x to the surface that we want to reconstruct. Then, the isosurface, meaning the surface defined by $d(x) = 0$, is extracted by triangulating the zero-crossing points of $d(x)$ by using the marching cubes algorithm [21]. Other integration methods include the mesh stitching method [22], [23] and the ball-pivoting algorithm (BPA) [24].

2.2 Range Data Acquisition of Optically Challenging Objects

In contrast to the work reported on the construction of 3D models using range images, relatively little work is seen

in the literature on the range acquisition of optically challenging objects. As mentioned earlier, it is a common practice to paint or coat objects before acquiring range data when the objects contain optically challenging surfaces.

Curless and Levoy [25] proposed a new range sensing method that is less sensitive to nonuniform light reflections by analyzing the time evolution of the light reflections. The method also improves the range data accuracy on sharp edges or discontinuous surfaces caused by sensor occlusions. However, it requires a large number of images to ensure that the laser stripe passes over every pixel in the image and, more importantly, the existence of spurious reflections is ignored.

Nayar et al. [26] proposed an iterative algorithm that recovers the shape and reflectance properties of surfaces in the presence of mutual reflections. This algorithm is useful for the shape-from-intensity approach to range acquisition. This approach, however, does not produce dense and accurate range maps compared to the optical triangulation methods. Additionally, the proposed algorithm was tested only on Lambertian surfaces of simple geometry.

Clark et al. [27] developed a laser scanning system that uses the polarization analysis to disambiguate the primary reflections from those caused by mutual reflections. Their system was tested on shiny aluminum objects with concavities, and spurious reflections were successfully discriminated. However, the system requires special equipment such as a linear polarizer, and multiple images need to be captured at each position of the laser: in their experiments, three images were acquired at three different angles of the linear polarizer.

Trucco and Fisher [28] proposed a number of consistency tests for acquiring reliable range images of specular objects. Their range sensor consists of two charge-coupled device (CCD) cameras observing a laser stripe from opposite sides. The consistency tests are based on the fact that the range measurements obtained from the two cameras will be consistent only if the measurements correspond to the true illuminated point. Their method was tested on a polished aluminum block with holes. However, their method does not consider the situation where more than one illuminated point is observed in the same camera scan line. The consistency tests are therefore applied only to the measurements corresponding to a single illuminated point observed per camera scan line. In our experiments, we have noticed that the illuminated points caused by mutual reflections occur very frequently in the vicinity of the true illuminated points and, thus, they are seen together along the same camera scan line. Eliminating all points whenever multiple illuminated points are observed in the same camera scan line may result in too few range measurements.

Unlike the previous work mentioned here, our system takes into account the existence of spurious reflections in the structured-light data, including multiple illuminated points in the same camera scan line. Also, the discrimination of spurious reflections is solely based on the algorithmic constraint tests; thus, no special hardware is required.

3 SYSTEM OVERVIEW

Fig. 3 shows the flowchart of our 3D modeling process. First, several scans from different viewpoints are performed so that the entire surface of the object can be captured. For each scan, the multippeak range imaging is carried out to account for the erroneous structured-light data. For each

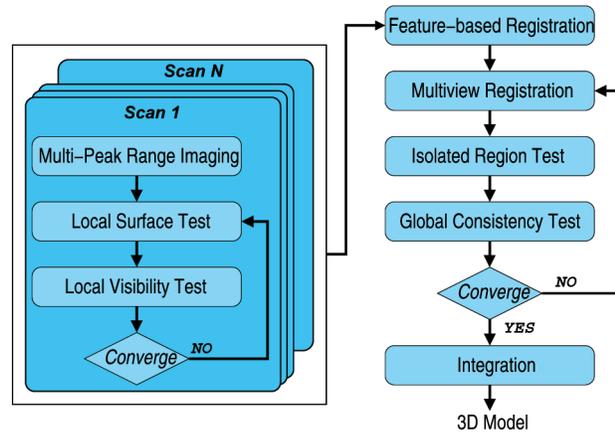


Fig. 3. Flowchart of our 3D modeling process.

range image produced, two constraint tests, namely, the local surface test and the local visibility test, are applied. The application of these two tests is repeated until the number of eliminated points during the tests becomes very small. For our experiment, we define the convergence of the local surface and the local visibility tests as the time when the number of eliminated points during the tests is less than 0.1 percent of the remaining points.

After executing the steps described above for each scan, the registration of all the range images acquired from different viewpoints is carried out. First, the initial registration process using a set of salient geometric features is performed. After obtaining an approximate registration, the multiview registration step fine-tunes the registration of all range images.

After all range images have been registered into a common coordinate system, two constraint tests, namely, the isolated region test and the global consistency test, are applied to all range images taken together. After applying these two constraint tests, the multiview registration step is carried out again on the new data set. Since we have eliminated some spurious measurements during the application of the constraint tests, we can expect the multiview registration applied on the new data set to yield a more precise registration. After the multiview registration, the two constraint tests are applied again, and this iteration is continued until it converges. Again, we define the convergence as the time when the number of eliminated points in a single iteration is less than 0.1 percent of the remaining points. Finally, the integration is performed on the resulting output by using the method proposed in [1].

4 MULTYPEAK RANGE IMAGING

In order to improve the conventional range imaging in the presence of erroneous structured-light data caused by various object surface related problems, we developed the *multipeak range imaging* technique, which is a new concept in range data collection for a more reliable extraction of range images [29], [30]. The basic idea is to store multiple candidates of range measurements when multiple illuminated points in the same camera scan line exist.¹ Since

1. The idea of storing multiple range measurements is similar to a Layered Depth Image [31]. The difference is that our multipeak range image stores at most one "true" range measurement that belongs to the object surface.

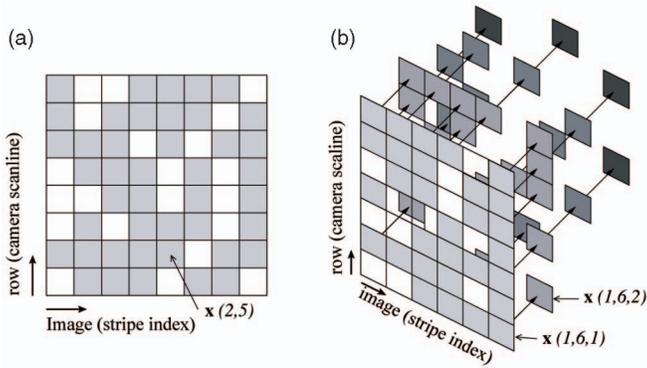


Fig. 4. Data structure of a conventional range image (a) and a multipeak range image (b).

choosing the highest intensity illuminated point does not guarantee the correctness of its corresponding range measurement and since a single camera scan line, or even the entire image, does not provide sufficient information to determine whether an illuminated point corresponds to a spurious reflection, our multipeak range imaging initially stores all detected illuminated points and then iteratively eliminates the false measurements generated by spurious reflections by applying a series of constraint tests. These constraint tests will be the topic of the next section.

Fig. 4 depicts the data structure used in conventional range imaging (Fig. 4a) and what we used in multipeak range imaging (Fig. 4b). In a conventional range image, the range measurements are arranged in an $m \times n$ grid, where typically, m is the number of scan lines of a camera image, and n is the number of captured images (that is, number of light stripes) in the scan. This grid can be represented by $x(i, j)$, where x represents the 3D coordinates of a range measurement at the i th camera scan line for the j th light stripe. To generalize this representation to multipeak range imaging, a range image element is now a linked list, where each node of the list contains 3D coordinates corresponding to one of the multiple peaks detected in the same scan line. Symbolically, the measured range information can now be represented by $x(i, j, k)$, where x is the 3D coordinates for the k th detected peak at the i th camera scan line for the j th light stripe.

Fig. 5a shows a multipeak range image acquired for a shiny bowl, where the intensity of the image represents the number of measurements, not the depth of a measurement that is typically used for visualizing a conventional range image. One can observe that a higher number of measurements are stored in the region corresponding to the inner parts of the bowl. This agrees with the fact that mutual reflections occur frequently at concave regions. The shape of the bowl in the multipeak range image is an oval shape because the bowl was captured at a tilted angle with respect to the perspective of the laser projector. Fig. 5b shows the computed 3D points visualized from the viewpoint of the camera, and Fig. 5c visualized from top to bottom. It is evident that the data shown in Fig. 5 contain many false measurements. Again, the objective of multipeak range imaging is to store all possible range measurements that are potentially true object surface points. It is the constraint tests that are responsible for eliminating the false measurements.

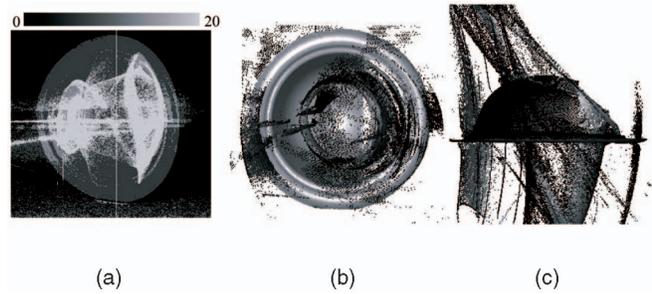


Fig. 5. Typical data acquired by multipeak range imaging. (a) A multipeak range image, where the intensity represents the number of measurements. (b) A 3D point cloud visualized from the camera viewpoint. (c) A 3D point cloud visualized from top to bottom.

5 FALSE MEASUREMENT ELIMINATION

The false measurement elimination process can be broken down into two stages. The first stage consists of the local surface test and the local visibility test and is performed on each range image separately. The second stage, which consists of the isolated region test and the global consistency test, is performed on all range images as a whole. From now on, we will refer to the local surface and the local visibility tests together as the *local tests* and to the isolated region and the global consistency tests together as the *global tests*.

Let us first define some terms and notations. First we use the term *rigel* to refer to a range image element (every cell in the matrix in Fig. 4a is a rigel). Note that a rigel in a multipeak range image may contain multiple range measurements. Let p_i denote the i th range measurement. For simplicity, we will assume that by having the index i , the measurement's row and column indices in the range image and the node index of its linked list are known. Recall that the linked list must account for multiple measurements in a rigel. When there is a superscript, for example, p_i^j , it represents the i th measurement in the range image acquired from the j th viewpoint. Let $x(p_i)$ and $\mathbf{n}(p_i)$ denote respectively the 3D coordinates and the unit surface normal vector² at p_i . The 3D coordinates and the surface normal vector are all with respect to the world frame, that is, the common coordinate system to which all range images are registered. Let $D(a, b)$ denote the closest euclidean distance between two elements a and b , where the elements can be 3D coordinates or line vectors. Finally, the angle between unit vectors $\mathbf{n}(p_i)$ and $\mathbf{n}(p_j)$ is denoted by $\Theta(\mathbf{n}(p_i), \mathbf{n}(p_j))$.

5.1 Local Surface Test

It is legitimate to assume that a range measurement lies on one of the three surface types: smooth surface,³ near crease edge, and near jump edge. A crease edge is where surface normals suddenly change directions, and a jump edge is where a spatial discontinuity between adjacent range measurements occurs. A common approach to estimate the local surface property of a range measurement is to fit a planar patch to the neighboring points, where the neighboring points are typically those within a small window (for example, 3×3 or 5×5) centered at the point in question. It has been shown in [33] that the planar patch can be reliably

2. We will assume that every range measurement belongs to a local surface patch, where the surface is estimated by fitting the best plane on the neighboring range measurements (see Section 5.1 for more details).

3. Additionally, smooth surfaces may be categorized into eight different types based on the surface curvature sign [32].

computed not only for those on smooth surfaces, but also for those located near edges by appropriately selecting neighbors. What we are trying to convey here is that given a true range measurement, whether it lies on a smooth surface or near an edge, we can find a planar patch that fits reasonably well to the carefully selected neighboring points that will all be on one or the other side of a crease boundary or a range discontinuity. If no suitable planar patch can be found, then it is likely that the measurement is spurious and should be eliminated. The elimination of such spurious measurements is the object of the local surface test.

Suppose a range measurement p_i is being tested. We first need to find appropriate neighbors to compute the planar patch that would best estimate the local surface at p_i . We define a measurement p_j to be a valid element of the fitting window for p_i if

$$D(\mathbf{x}(p_i), \mathbf{x}(p_j)) < \rho B(p_i, p_j), \quad (1)$$

where ρ represents the maximum distance allowed between two immediate neighbors, and $B(p_i, p_j)$ is the city-block distance [34] between p_i and p_j . In case multiple valid elements exist in the same rigel (due to multiple detected peaks), the one with the smallest distance is selected. The parameter ρ should be a function of the range sensor resolution and should be chosen properly so that the adjacent points between a jump edge would not become valid elements for each other's fitting window. We empirically set ρ to be four times the range sensor resolution.

Let $m(p_i)$ denote the number of valid elements in a fitting window, including the point in question p_i , that satisfy (1). Let us denote the valid elements as p_e , $e = 1, \dots, m(p_i)$, and the center of mass of the elements as \mathbf{x}_c . Then, the covariance matrix $\mathbf{C}(p_i)$ is computed by

$$\mathbf{C}(p_i) = \sum_{e=1}^{m(p_i)} [\mathbf{x}(p_e) - \mathbf{x}_c][\mathbf{x}(p_e) - \mathbf{x}_c]^T.$$

The eigenvector corresponding to the smallest eigenvalue of $\mathbf{C}(p_i)$ represents the normal vector of the best fitting plane. Thus, this eigenvector is used as the estimate of the surface normal at p_i , denoted as $\mathbf{n}(p_i)$. The euclidean distance between an element p_e and the best fitting plane is simply the scalar projection of the vector $\mathbf{x}(p_e) - \mathbf{x}_c$ onto the plane's surface normal $\mathbf{n}(p_i)$. Thus, the fitting error of the elements to the best fitting plane, denoted as $\varepsilon(p_i)$, is given by

$$\varepsilon(p_i) = \frac{1}{m(p_i)} \sum_{e=1}^{m(p_i)} [\mathbf{x}(p_e) - \mathbf{x}_c] \bullet \mathbf{n}(p_i).$$

There are two constraints in the local surface test. All range measurements that do not satisfy either constraint are eliminated. The first constraint requires that each range measurement p_i has the number of valid elements for its fitting window greater than a threshold τ_m :

$$m(p_i) > \tau_m. \quad (2)$$

The second constraint in the local surface test requires that the fitting error of the best fitting plane be less than τ_ε :

$$\varepsilon(p_i) < \tau_\varepsilon. \quad (3)$$

In general, the threshold τ_m must be high enough so that the best fitting plane can be reliably computed, but low enough so that the points near a jump edge will not be eliminated.

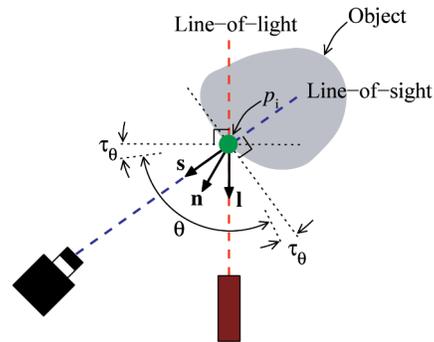


Fig. 6. Local visibility constraint.

The threshold τ_ε must be set in such a way that the points on crease edges will not be eliminated. Section 6.2 discusses in detail how the two thresholds τ_m and τ_ε can be set.

5.2 Local Visibility Test

The local visibility test checks whether the direction of the estimated surface normal meets the constraints imposed by the sensor visibility. Based on the Lambertian reflectance model [35], it has been commonly accepted to assign a measurement's confidence value as the dot product between the surface normal and the projector's line-of-light. Therefore, we can impose a constraint that the angle between the surface normal and the laser projector's line-of-light be less than 90 degrees. Using a similar rationale, we can impose another constraint that the angle between the surface normal and the camera's line-of-sight be less than 90 degrees, since otherwise, the illuminated points on the object surface could not have been seen by the camera. Fig. 6 illustrates the local visibility constraints. Let the unit vector \mathbf{l} represent the laser projector's line-of-light, the unit vector \mathbf{s} the camera's line-of-sight, and the unit vector \mathbf{n} the estimated surface normal at p_i . Writing the two constraints mentioned above into equations, we have

$$\mathbf{n}(p_i) \bullet \mathbf{l}(p_i) < \cos\left(\frac{\pi}{2} - \tau_\theta\right), \quad (4)$$

$$\mathbf{n}(p_i) \bullet \mathbf{s}(p_i) < \cos\left(\frac{\pi}{2} - \tau_\theta\right), \quad (5)$$

where τ_θ is a small angle (we shall discuss about the threshold τ_θ shortly). Pictorially, a range measurement is valid if the direction of its surface normal lies within the range of angles indicated by θ in Fig. 6. All measurements that do not satisfy the local visibility constraints given by (4) and (5) are eliminated.

By having the threshold τ_θ , we are eliminating the measurements whose surface normals are near orthogonal to either the projector's line-of-light or the camera's line-of-sight. This way, the measurements with very low confidence can be eliminated without having to go through the later constraint tests that are more computationally expensive. Apart from this obvious reason for having the threshold τ_θ , another reason is that a significantly large number of false measurements have surface normals near orthogonal to the corresponding line-of-sight vectors. These false measurements are usually generated by the spurious reflections that managed to stay at the same position with respect to the world frame over consecutive camera images. Let us elaborate with the illustration shown in Fig. 7. The positions of the laser projector and the camera at three

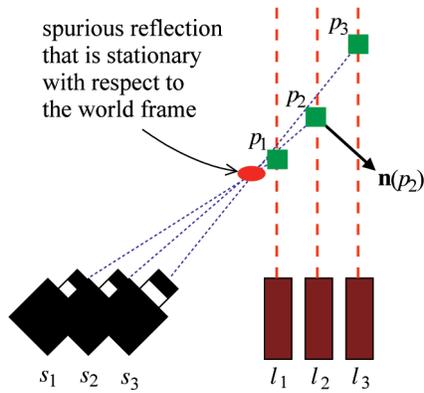


Fig. 7. Illustration of a phenomenon that occurs when spurious reflections that are stationary with respect to world frame are observed in consecutive camera images.

consecutive sampling positions during a scan are shown. Suppose that the camera observes a spurious reflection, depicted as the red dot, that is stationary with respect to the world frame at all three sampling positions. In practice, such a spurious reflection can be a secondary reflection on the object surface or simply a bright spot in the background of the scene. The false illuminated point will result in 3D points p_1 , p_2 , and p_3 at the respective sampling positions. As one can observe, these points line up in the direction near the camera's line-of-sight. Consequently, their estimated surface normals will be close to orthogonal to the line-of-sight vectors. For example, estimating the surface normal $\mathbf{n}(p_2)$ using the neighboring points p_1 and p_3 , one can observe that $\mathbf{n}(p_2)$ would be near orthogonal to the line-of-sight vector of p_2 .

Fig. 8a shows 3D points of a multippeak range image visualized from top to bottom, where the false measurements generated by the phenomenon described above are seen along the several diagonal lines that extend from the bottom left to the top right of the image. Fig. 8b shows the data after applying a single iteration of only the local surface test. Notice that false measurements on the diagonal lines are still prevalent. Figs. 8c and 8d show the data after applying a single iteration of both the local surface test and the local visibility test, with the threshold τ_θ set as 5 degrees and 10 degrees, respectively. Many of the false measurements that the local surface test alone could not eliminate were successfully eliminated with the local visibility

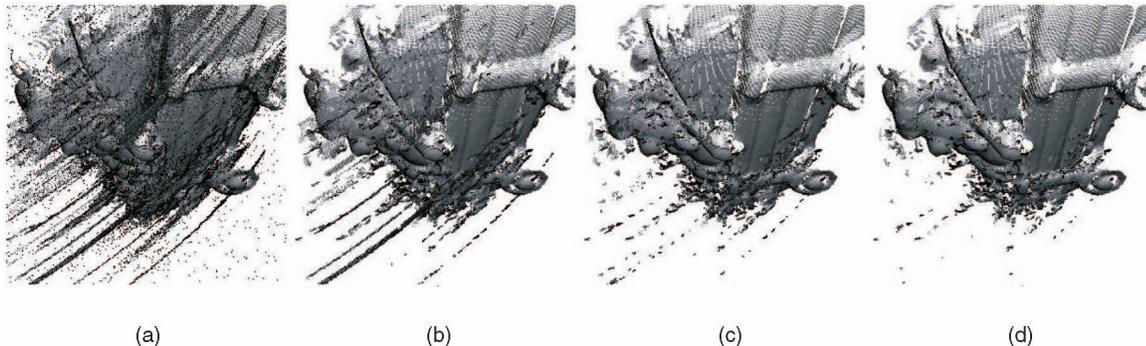


Fig. 8. Illustration of the effect of the local visibility test. (a) Original data. (b) After applying only the local surface test. (c) After applying the local surface test and the local visibility test, with τ_θ set as 5 degrees. (d) After applying the local surface test and the local visibility test, with τ_θ set as 10 degrees.

constraints. We will discuss about setting the threshold τ_θ in Section 6.2.

When the local surface and the local visibility tests converge (see Section 3 for the definition of convergence), a weight w (or confidence value) is assigned to each range measurement by incorporating all four constraints applied in the local tests (that is, (2), (3), (4), and (5)). We assign equal weight to all four constraint tests, and by normalizing the weight contribution from each constraint test to 1, w is also normalized to 1.

5.3 Registration

5.3.1 Geometric Feature-Based Registration

The feature-based registration serves as the initial step that provides an approximate registration for the next step, the multiview registration, that fine-tunes the registration. First, the feature-based registration extracts a set of salient geometric features from the range data. Then, the correspondences between the features of the anchor data set and the moving data set are established. Finally, the best transformation is computed using the feature correspondences.

To be useful in the registration of the kinds of range data that we are interested in, the features not only need to be invariant under rigid transformations but also insensitive to noise and false measurements. We developed a 3D feature descriptor that is similar to the one described in [19]. The main differences are 1) our descriptor considers a set of neighboring 3D points as opposed to a local surface geometry, 2) our descriptor uses a 2D Gaussian kernel to maintain the isotropic property of the descriptor, and 3) our descriptor is designed to handle multiple candidates of neighboring points. The 3D feature descriptor is computed at each measurement p_i as follows: In a small window centered at p_i , the neighboring points are obtained by the same procedure as described earlier in Section 5.1. Using the same notations as before, let the valid neighboring points be p_e , $e = 1, \dots, m(p_i)$, where $m(p_i)$ is the total number of valid neighboring points. Also, let the distance from p_e to the tangent plane at p_i be denoted as d_e , that is, $d_e = [\mathbf{x}(p_e) - \mathbf{x}(p_i)] \bullet \mathbf{n}(p_i)$. Then, the 3D feature descriptor f is defined as

$$f(p_i) = \frac{1}{m(p_i)} \sum_{e=1}^{m(p_i)} d_e g_e,$$

where g_e is the corresponding 2D Gaussian kernel value at p_e . By applying the 2D Gaussian kernel, the isotropic property of the descriptor is maintained. Moreover, the

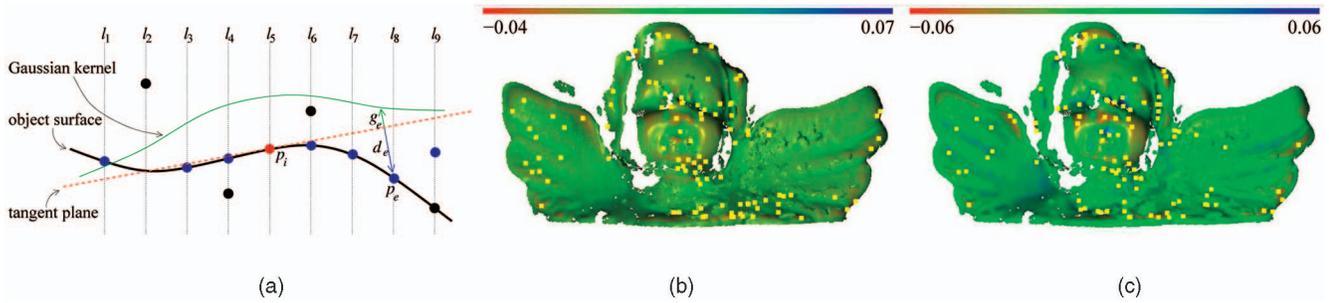


Fig. 9. (a) Illustration of the 3D feature descriptor. (b) Color-coded 3D feature descriptor values and 50 extracted salient feature points using a 15×15 kernel window. (c) Result using a 25×25 kernel window.

Gaussian kernel suppresses the noise, especially around the boundary of the window where erroneous neighboring points are more likely to arise. Fig. 9a illustrates the 3D feature descriptor. The solid line represents the object surface, and the blue points are the selected neighboring points of the measurement p_i in the nearby scan lines. Notice at most one neighboring point per rigel that satisfies (1) is selected, and in case of multiple candidates in the same rigel (for example, along l_4 , l_6 , and l_9), the one with the smallest distance to p_i is selected. Although the spurious measurement along l_9 has been falsely selected as a neighbor, it has little effect on the descriptor value $f(p_i)$, since the corresponding 2D Gaussian kernel value is small.

After computing the 3D feature descriptor values, we then extract a small set of salient points whose descriptor values are distinctive. We use a standard histogram approach to populate the descriptor values and select the salient points starting from the least populated bin until the number of salient points reaches a certain threshold. Since we want to avoid selecting points that are too close to one another, we require that all salient points be separated by at least a certain distance threshold: we empirically choose this distance threshold to be 20 times the range sensor resolution. Since we also want to obtain both small and large-scale salient features, we first extract salient points by using a small kernel window and combine those with the ones extracted using a large kernel window. In our experiments, we use 15×15 and 25×25 windows. Fig. 9b shows the color-coded 3D feature descriptor values computed using a 15×15 kernel window. The yellow squares are 50 selected salient feature points. Fig. 9c shows the result of using a 25×25 kernel window that produces larger-scale feature points.

After extracting a set of salient feature points from the anchor data set and the moving data set, we next establish the feature correspondences between the two data sets. Since we are dealing with very noisy range images, it is not easy to compute an optimal threshold value that will minimize false correspondences. Instead, we assign a “loose” threshold for correspondences that will generate a large number of false correspondences and rely on the Random Sample Consensus (RANSAC) algorithm [36] to find the correct correspondences.⁴ Even when the RANSAC

4. Over the years, the RANSAC method has emerged as a highly effective approach to the elimination of outliers in the candidate matches between two data sets (two range maps in our case). Given a set of candidate matches between two range maps, the RANSAC approach consists of using the least possible number of matches to construct a pose transform between the two range maps and then ascertaining the extent to which the estimated pose is supported by the rest of the matches. If the consensus set is large enough, then the pose estimate is accepted (and, in some cases, further refined using all the accepted matches).

algorithm results in incorrect correspondences, it is easy to verify the result by applying a pairwise ICP algorithm. If the initial correspondences given by the RANSAC algorithm are incorrect, then the registration error at the convergence of ICP will be large, and vice versa. In the case of an incorrect solution, we can apply the RANSAC algorithm again until the solution is verified by the ICP algorithm. Fig. 10 illustrates the process of feature-based registration.

This newly proposed feature-based registration has been tested on all of the data presented in our experimental section and has proved to be reliable and robust to noisy data. However, the current implementation assumes that the two data sets to be registered have sufficient overlapping area. Since the RANSAC algorithm continues to search for the correct correspondences until they are verified by the ICP algorithm, we provided an ordered list of range images to the algorithm so that any two successive data sets for each registration will have some common surface areas.

5.3.2 Multiview Registration

Our approach for multiview registration is similar to the one proposed by Bergevin et al. [16]. Adapting the correspondence criteria presented in [37], our method selects the corresponding points between two data sets as the closest points with the angle between surface normals less than a threshold. The thresholds for selecting the corresponding points are set dynamically in each iteration of the ICP by using an approach similar to the one proposed by Zhang [14]. Since our correspondence criteria also include the angle between surface normals, the angle threshold is also computed dynamically using the same approach as the distance. Our multiview registration randomizes the processing order of range images to reduce any bias toward a particular range image [17]. Also, similar to the method in [23], the weights of the range measurements are taken into account when computing the transformation that minimizes the distances between the corresponding points.

Even after multiview registration, we must anticipate some registration errors, which generally depend on the accuracy of the previous registration and the number of remaining false measurements in the data. The registration errors play an important role in the global tests because the global tests utilize the information from all range images, which is greatly influenced by how well the range images are registered with one another. Note that computing the registration error is not trivial mainly for two reasons. First, we do not know which parts between the data sets are

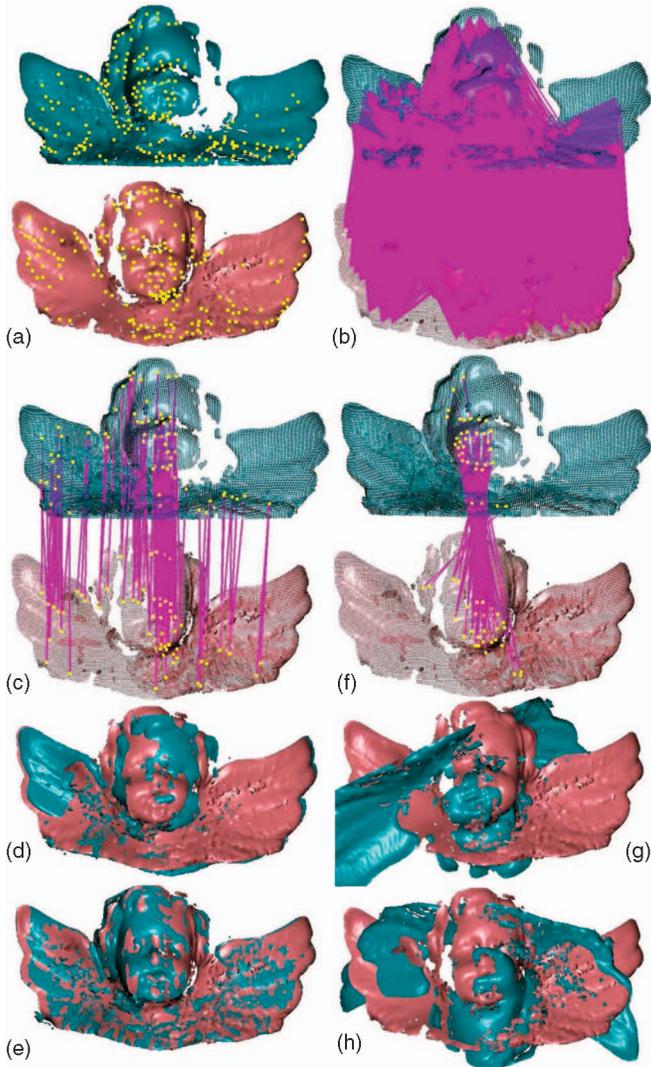


Fig. 10. Feature-based registration. (a) Two data sets to be registered with extracted salient features. (b) Initial feature correspondences. (c) A set of correct correspondences obtained via RANSAC. (d) After the transformation using the correspondences shown in (c). (e) After applying ICP. The registration is verified to be correct. (f) An example of incorrect correspondences. (g) After the transformation using the correspondences shown in (f). (h) After applying ICP. The registration is incorrect.

overlapping and which parts are distinctive. Second, we do not know which measurements correspond to true surface points and which ones are spurious. Nevertheless, it is commonly accepted in the literature on registration algorithms that the mean distance between the corresponding points or the distance threshold for the correspondence search at the termination of the ICP algorithm be the estimate of the registration error. We adopted the latter approach, where we use the distance and the angle thresholds at the termination of the ICP algorithm as the estimates of the registration error. We will denote the registration error of the i th range image with respect to the distance as λ_D^i and with respect to the angle as λ_θ^i .

5.4 Isolated Region Test

The main purpose of the isolated region test is to eliminate all points that are far separated from the true object points. The test involves constructing a 3D volumetric grid that

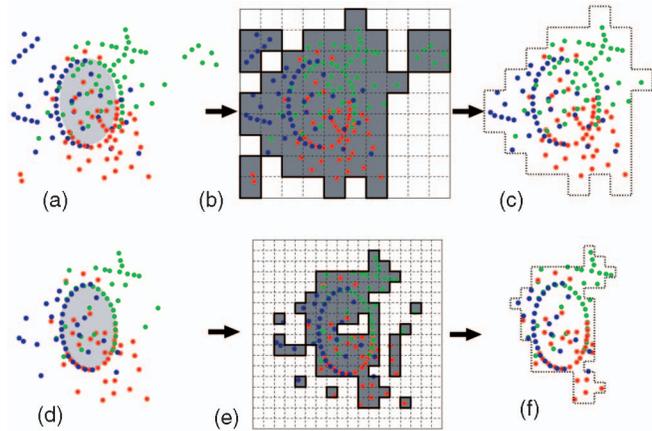


Fig. 11. Illustration of the isolated region test. (a) Range measurements from three range images indicated by red, blue, and green points. The three range images sampled the oval-shaped object from different directions. (b) The volumetric grid (shown in 2D), where the gray color represents the binary value of 1, and the rest the binary value of 0. The regions of connected voxels are shown as the regions that are enclosed by thick lines. (c) The isolated region test eliminated all points except the ones that belong to the largest connected region. (d) The same data as in (c), but with a better registration among the data. (e) The volumetric grid containing the data in (d). The grid resolution is smaller than the one in (b) because of better registration among the data. (f) After another isolated region test.

contains the entire data, with each voxel having a binary value of 1 if any point exists inside the voxel, and 0 otherwise. A voxel segmentation based on region growing [34] is then performed to group the connected voxels. In order to ensure that all true measurements belong to a common connected voxel region, the resolution of the volumetric grid is set to $\max(\max_i(\lambda_D^i), \rho)$. Recall that ρ is the maximum distance allowed between two immediate neighbors defined in Section 5.1. The isolated region test eliminates all measurements except the ones that belong to the region with the largest size. By the largest size we mean the largest number of connected voxels in a region (see Fig. 11 for a pictorial illustration of the isolated region test).

5.5 Global Consistency Test

The global consistency test is based on two criteria: the *coordinate consistency* and the *visibility consistency*. Here, the word “global” implies that the consistency is tested among all possible range images acquired from different viewpoints.

5.5.1 Coordinate Consistency

The coordinate consistency states that the 3D coordinates of true measurements are always consistent among the registered range images that capture the same object surface. On the other hand, the 3D coordinates of false measurements generated by noise or mutual reflections are likely to be inconsistent, since the locations where the spurious reflections occur vary from viewpoint to viewpoint. For example, consider the scene depicted in Fig. 12. On the left, a specular object is being sampled. Due to the mutual reflections, three range measurements a , b , and c are generated. On the right, the same object point is now being sampled from a different viewpoint. As the surface normal relative to the direction of the light source changes, the bouncing light direction also changes. In this case, two illuminated points are detected by the camera, resulting in two range measurements d and e . Observe that only the

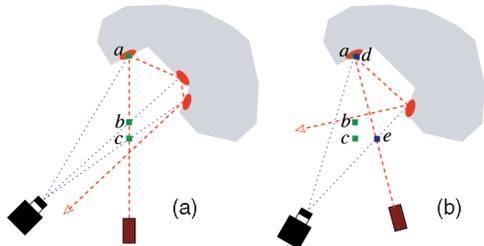


Fig. 12. Illustration of the coordinate consistency.

coordinates of the true range measurements a and d are consistent, whereas the coordinates of the false measurements are inconsistent with one another.

Assuming that we have range data from N different viewpoints and that there are a total of M^j measurements in the j th range image, the coordinate consistency value, denoted as \mathcal{C} , is computed as

$$\mathcal{C}(p_i^j) = w(p_i^j) + \sum_{v=1, v \neq j}^N \max_{u=1, \dots, M^v} \left\{ \delta_{\mathcal{C}}(p_i^j, p_u^v) \cdot w(p_u^v) \right\}. \quad (6)$$

Here, the test function $\delta_{\mathcal{C}}$ is given by

$$\delta_{\mathcal{C}}(p_i^j, p_u^v) = \begin{cases} 1 & \text{if } D(\mathbf{x}(p_i^j), \mathbf{x}(p_u^v)) < \max(\lambda_D^j, \lambda_D^v) \text{ and} \\ & \Theta(\mathbf{n}(p_i^j), \mathbf{n}(p_u^v)) < \max(\lambda_{\theta}^j, \lambda_{\theta}^v) \\ 0 & \text{otherwise.} \end{cases}$$

Recall that D is the closest euclidean distance between two elements, Θ is the angle between surface normals, and w represents the weight. Notice in the test function $\delta_{\mathcal{C}}$ that the corresponding registration errors for each range measurement are incorporated. Notice also in (6) that by taking the maximum value, we are limiting the coordinate consistency value to be contributed only once per range image in case multiple valid points in the same range image exist. In other words, for each measurement p_i^j , the maximum of $N-1$ measurements can contribute to its coordinate consistency value. Since the weight w is normalized to 1 and since the coordinate consistency includes the weight value of its own, the upper bound of the coordinate consistency value is N . This allows us to obtain a more balanced distribution of the coordinate consistency values throughout the data, and more importantly, it allows us to combine the coordinate consistency and the visibility consistency, as we will explain in the next section.

5.5.2 Visibility Consistency

The second criterion of the global consistency test, which we call *visibility consistency*, is based on the facts that a line space between a sensor and a true measurement is empty and that a line space beyond a true measurement is invisible to the sensor. Note that these basic concepts are also used in the space carving approaches [38], but in the context of reconstructing 3D objects from a set of calibrated images. Some other researchers have used a similar concept to remove small outliers [1], [39].

Although we can apply the visibility consistency concept for both the projector's line-of-light and the camera's line-of-sight, only the former is considered in our work: obtaining the visibility consistency for the camera's line-of-sight is computationally much more expensive than that for the projector's line-of-light, as we will explain in more detail in Section 5.5.4.

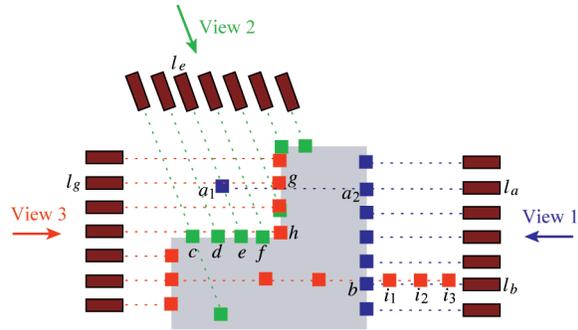


Fig. 13. Illustration of the visibility consistency.

Consider the example depicted in Fig. 13, where an object was scanned from three different viewpoints. The dotted lines represent the projector's lines of light at the respective sampling positions during the scan. Suppose in the first view, among other detected measurements, points a_1 and a_2 were detected in the same rigel. If a_1 is a true measurement (we do not know yet which one is true), then the space at a_2 should have been empty. On the other hand, if a_2 is true, then a_1 should not have been visible (or illuminated) by the projector l_a . Thus, the measurements a_1 and a_2 are inconsistent with each other with respect to the projector l_a . In fact, the measurements in the same rigel are always inconsistent with each other, for there cannot be more than one true measurement in the same rigel. The visibility concept applies also for the measurements obtained from different viewpoints. For example, a_1 acquired from the first view and e from the second view are inconsistent with respect to the projector l_e . Similarly, a_1 and g are inconsistent with respect to the projector l_a and also with respect to l_g .

In order to test the visibility consistency of a range measurement p_i^j , we first need to check for additional measurements in the same rigel, since multiple measurements in the same rigel are always inconsistent with one another. We then need to check for measurements from other range images that lie on the line-of-light, where p_i^j was sampled. Let the light projector that sampled p_i^j be labeled as l_i^j and the 3D coordinates of the light projector with respect to the world frame be $\mathbf{x}(l_i^j)$. Then, the ray equation of the line-of-light that sampled p_i^j , denoted as $\vec{\mathbf{l}}(p_i^j)$, is given by

$$\vec{\mathbf{l}}(p_i^j) = \mathbf{x}(l_i^j) + \alpha(\mathbf{x}(p_i^j) - \mathbf{x}(l_i^j)), \quad \alpha = (0, \infty).$$

Here, the ray symbol $\vec{\cdot}$ is used to distinguish from the unit vector of line-of-light \mathbf{l} described in Section 5.2. The closest distance between $\vec{\mathbf{l}}(p_i^j)$ and a range measurement, say, p_u^v , can be computed by

$$D(\mathbf{x}(p_u^v), \vec{\mathbf{l}}(p_i^j)) = \frac{\|(\mathbf{x}(p_i^j) - \mathbf{x}(l_i^j)) \otimes (\mathbf{x}(p_u^v) - \mathbf{x}(l_i^j))\|}{\|\mathbf{x}(p_i^j) - \mathbf{x}(l_i^j)\|}. \quad (7)$$

Here, \otimes is a cross product, and $\|\cdot\|$ is a vector magnitude. Obviously, few range measurements will lie exactly on a ray; therefore, we use the registration error for determining whether a point lies on a line-of-light. That is, we define that a range measurement p_u^v lies on the line-of-light $\vec{\mathbf{l}}(p_i^j)$ if

$$D(\mathbf{x}(p_u^v), \vec{\mathbf{l}}(p_i^j)) < \max(\lambda_D^j, \lambda_D^v). \quad (8)$$

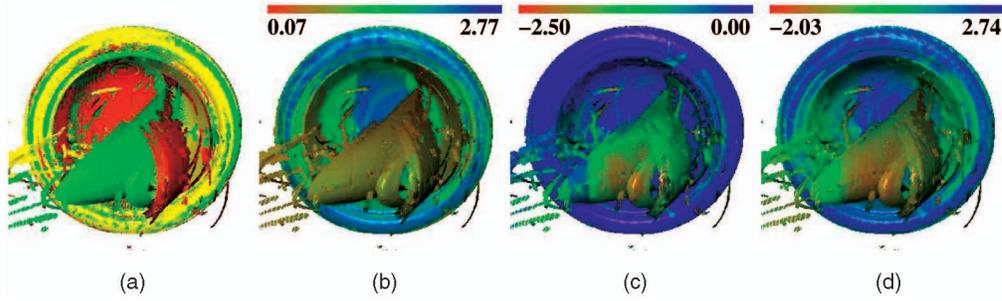


Fig. 14. (a) Three-dimensional points from three multipeak range images acquired for a bowl. Different colors represent individual range image. (b) The same data as in (a), but the color represents the coordinate consistency value \mathcal{C} , where the range of the values is indicated by the legend bar on the top. (c) Color-coded visibility consistency value \mathcal{V} . (d) Color-coded global consistency value \mathcal{G} .

Once we find the measurements that lie on the line-of-light $\bar{\mathbf{l}}(p_i^j)$, the distance between each of those found measurements and $\mathbf{x}(l_i^j)$ (the 3D coordinates of the light projector that sampled p_i^j) needs to be computed. If that distance is less than the distance between $\mathbf{x}(p_i^j)$ and $\mathbf{x}(l_i^j)$, then there is a visibility inconsistency. For example, in Fig. 13, the points a_1 and e are inconsistent because a_1 lies on the line-of-light of e , and $D(\mathbf{x}(e), \mathbf{x}(l_e)) > D(\mathbf{x}(a_1), \mathbf{x}(l_e))$. On the other hand, although a_2 and g lie on each other's line-of-light, they are consistent because $D(\mathbf{x}(g), \mathbf{x}(l_g)) < D(\mathbf{x}(a_2), \mathbf{x}(l_g))$ and $D(\mathbf{x}(a_2), \mathbf{x}(l_a)) < D(\mathbf{x}(g), \mathbf{x}(l_a))$. Again, we need to take into account the registration error when comparing the distances to the projector. Therefore, we define that there is a visibility inconsistency between p_i^j and p_u^v with respect to the light projector l_i^j if (8) is satisfied, and

$$D(\mathbf{x}(p_i^j), \mathbf{x}(l_i^j)) > D(\mathbf{x}(p_u^v), \mathbf{x}(l_i^j)) + \max(\lambda_D^j, \lambda_D^v). \quad (9)$$

A straightforward implementation of the visibility consistency can result in situations where true measurements may be determined to be highly inconsistent. For example, the point h in Fig. 13 is inconsistent with four measurements c , d , e , and f from the second view, even though all of them are true measurements. This kind of situation may occur between two orthogonal surfaces, where each side of the surfaces is captured by a different range image. For another example, the point b is inconsistent with three false measurements i_1 , i_2 , and i_3 that happened to be along the line-of-light of b . Similar situations can occur more frequently as the noise increases in the scene, and the object shape becomes more complex.

In order to consider all the discussions above, we compute the visibility consistency value for each measurement as follows: Using the same notations as (6), the visibility consistency value of p_i^j , denoted as $\mathcal{V}(p_i^j)$, is defined as

$$\mathcal{V}(p_i^j) = \sum_{v=1}^N \min_{u=1, \dots, M^v} \left\{ \delta_{\mathcal{V}}(p_i^j, p_u^v) \cdot w(p_u^v) \cdot |\mathbf{n}(p_i^j) \bullet \mathbf{n}(p_u^v)| \right\}. \quad (10)$$

Here, the test function $\delta_{\mathcal{V}}$ is given by

$$\delta_{\mathcal{V}}(p_i^j, p_u^v) = \begin{cases} -1 & \text{if } p_i^j \text{ and } p_u^v \text{ are in the same rigel or} \\ & \text{if (8) and (9) both satisfy} \\ 0 & \text{otherwise.} \end{cases}$$

There are several things that need to be mentioned. First, notice that $\mathcal{V}(p_i^j)$ can only have a zero or a negative value, since the test function $\delta_{\mathcal{V}}$ checks only for inconsistency. Second, the absolute value of the dot product between the surface normal vectors is multiplied so that any two measurements whose surface normals are close to orthogonal (such as points c and h in Fig. 13) have little effect on each other. Also, only the minimum consistency value (that is, highest inconsistency) computed for each range image is added to $\mathcal{V}(p_i^j)$ in order to prevent the situation described earlier with the example of the points b , i_1 , i_2 , and i_3 in Fig. 13. By adding only the smallest consistency value for each range image, the lower bound of the visibility consistency value is $-N$, which enables us to obtain more evenly distributed values of \mathcal{V} throughout the data.

5.5.3 Constraints for the Global Consistency Test

The total global consistency value, denoted as \mathcal{G} , is simply the sum of the coordinate consistency and the visibility consistency:

$$\mathcal{G}(p_i^j) = \mathcal{C}(p_i^j) + \mathcal{V}(p_i^j). \quad (11)$$

The main reason for considering the global consistency value \mathcal{G} is that we discovered in our experiment that it is easier to distinguish between the true and the false measurements by using \mathcal{G} as opposed to applying \mathcal{C} or \mathcal{V} separately. We are able to add \mathcal{C} and \mathcal{V} , since the scales of both values are normalized to the number of range images. By simply adding \mathcal{C} and \mathcal{V} without any coefficients, we are assuming that they have equal weights.

Fig. 14a shows the 3D points from three multipeak range images acquired for a bowl, where different colors represent each multipeak range image. Fig. 14b shows the same data where the color now represents the coordinate consistency value \mathcal{C} indicated by the legend bar on the top. Clearly, the surface areas where overlaps between range images occur have higher consistency values whereas distinctive surface areas, both the false measurements and some of the true measurements that were captured only once from a single viewpoint, have low consistency values. Fig. 14c shows the visibility consistency value \mathcal{V} for the same data. As expected, most of the false measurements have negative values of consistency, meaning they are highly inconsistent, and most of the true measurements have zero values, meaning no visibility inconsistency was detected. Finally, Fig. 14d shows the global consistency value \mathcal{G} .

Given the values of \mathcal{G} , the global consistency test employs a simple threshold technique in order to eliminate

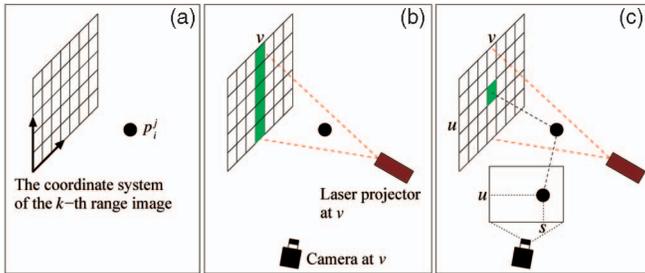


Fig. 15. Projecting a point onto a range image coordinate system.

the measurements with low consistency values. There are two constraints in the global consistency test. The first constraint requires that the global consistency value of a measurement be greater than a threshold τ_G :

$$\mathcal{G}(p_i^j) > \tau_G \quad (12)$$

with the threshold set as

$$\tau_G = \min(\mu_G - t\sigma_G, 0), \quad (13)$$

where μ_G and σ_G are the mean and standard deviation of \mathcal{G} , and t is a positive constant. We will evaluate the effect of varying t in Section 6.3. Note that by forcing $\tau_G \leq 0$, we are preserving all measurements that do not have any neighbors from other range images, as long as there is no visibility inconsistency with other measurements. This property lifts an otherwise restrictive requirement that every part of the object surface must be sampled at least twice from different viewpoints.

The second constraint of the global consistency test also eliminates the measurements with small \mathcal{G} values, but the difference from the first constraint is that it now considers only the range measurements in the same rigel. We define a measurement in a rigel to have a small \mathcal{G} if the value is smaller than the maximum \mathcal{G} in that rigel minus $t\sigma_G$, where t is the same constant as used in (13). Formally, let $\mathcal{G}_{max}(p_i^j)$ denote the maximum global consistency value among all the range measurements in the rigel to which p_i^j belongs. Then, the second constraint requires that

$$\mathcal{G}(p_i^j) > \mathcal{G}_{max}(p_i^j) - t\sigma_G. \quad (14)$$

Note that the second constraint only applies to those with more than one measurement in the same rigel. All measurements that do not satisfy either constraints (12) or (14) are eliminated in the global consistency test.

5.5.4 Implementation of the Global Consistency Test

Since the range data used in 3D modeling typically consists of several millions of 3D points, an efficient algorithm for the global consistency test is necessary. For example, a straightforward implementation of the visibility consistency test would require that (7) and (8) be carried out for every point in every range image against every point in all the rest of range images. In this section, we describe an efficient implementation of the global consistency test. Consider the images shown in Fig. 15. Suppose we want to compute the visibility consistency of a range measurement p_i^j against all range measurements in the k th range image. Recall that the global

consistency test is applied after the multiview registration process, which provides the 3D coordinates of p_i^j with respect to the world frame and the coordinate system of the k th range image also with respect to the world frame. Thus, we can easily compute the position of the laser projector that would have illuminated p_i^j during the acquisition of the k th range image (see Fig. 15b). Given the position of the laser projector, we immediately obtain the corresponding column index v of the k th range image and the position of the camera that would have captured an image at that time instance. Then, as shown in Fig. 15c, by projecting the 3D coordinates of p_i^j onto the image plane of the camera, we can obtain the projected image coordinates (u, s) that represent the center of illuminated points if the light reflection on the object surface at p_i^j were observed by the camera. At this point, we know that p_i^j would have belonged to rigel (u, v) in the k th range image. This implies that p_i^j lies on the line-of-light of the measurements at rigel (u, v) of the k th range image and, therefore, we do not need to consider any other measurements in the k th range image.⁵

We said earlier in Section 5.5.2 that only the visibility consistency with respect to the laser projector's line-of-light is considered in our work. As we just illustrated, by simply projecting a 3D point onto a range image coordinate system, we are able to dramatically reduce the number of measurements that need to be tested for the visibility consistency with respect to the laser projector's line-of-light. For the camera's line-of-sight, however, we cannot use a similar projection approach, since the camera view at each sampling point is not a plane, but rather a four-sided pyramid extending from the center of projection of the camera. Therefore, a more sophisticated computation is required for the line-of-sight visibility.

Here, we describe the outline of our implementation of the global consistency test. For each range image k , all points in the rest of the range images are projected onto the k th range image coordinate system. For each projected point, if its projection coordinates (u, v) are inside the boundary of the k th range image, then the coordinate consistency is tested between the projected point and the points in rigel (u, v) of the k th range image. Since the projection coordinates will most likely be in floating points and since the registration error needs to be taken into consideration, all points in a small window centered at rigel (u, v) are tested, where the size of the window is controlled by the registration error. The visibility consistency is tested only if the coordinate consistency test fails (that is, $\delta_c = 0$).

5.6 Computational Complexity

In this section, the computational complexity of each of the constraint tests is analyzed. For simplicity, we will assume that the number of range measurements in each range image is the same and denoted by M . As before, the number of range images is denoted by N .

The computation of the local surface test consists of two main steps: 1) determining valid elements in a fitting window and 2) computing the best fitting plane for the

5. For this, we are assuming that the laser projection is orthographic, which is a reasonable assumption, given that the laser projector is sufficiently distanced from the object.



Fig. 16. Objects used in the experiment. (a) A stainless steel bowl. (b) and (c) Ceramic glazed seashell and angel figurines. (d) A tray figurine made of multiple materials, including specular, Lambertian, and highly absorptive surfaces. (e) A gorilla figurine made of black plastic. All objects have surface materials that are optically challenging.

valid elements. Both steps require $O(M)$ time for a single range image. Therefore, the total of $O(NM)$ is required for the local surface test for all range images.

The local visibility test simply examines the angles between the estimated surface normals and the corresponding lines of light and lines of sight. Therefore, the time complexity for the local visibility test is also $O(NM)$.

The isolated region test first constructs a binary volumetric grid, which requires $O(NM)$ time, and then performs a voxel segmentation. The complexity of a voxel segmentation depends on the number of voxels in the volumetric grid, where the number of voxels depends on the resolution and the size of the grid. Here, we will assume that the number of voxels grows only in proportion to the size of the object being reconstructed. Then, denoting the number of voxels as v , the time complexity of a voxel segmentation is $O(v)$. Altogether, the isolated region test requires $O(NM + v)$ time.

The global consistency test consists of computing the coordinate consistency and the visibility consistency. The exhaustive approach for computing the coordinate consistency of each point would require $O(NM)$ time and, thus, $O(N^2M^2)$ would be required for all points. A more efficient alternative is to use binning such as k-d tree [40] or Elias [41] methods. For example, it has been shown in [42] that the time complexity of traversing a k-d tree containing M samples is proportional to $\log M$. Utilizing one of the binning methods, therefore, we can expect the time complexity of the coordinate consistency test to be $O(NM \log(NM))$.

The computation of the visibility consistency would require $O(N^2M^2)$ time when implemented, as described in Section 5.5.2. We cannot simply use the k-d tree or the Elias method here, since the computation of the closest distance between a point and a line vector in 3D space is involved in the visibility consistency test. On the other hand, when the visibility consistency is implemented, as described in Section 5.5.4, the time complexity reduces to $O(NM)$, since only the measurements in the projected rigel need to be considered in each range image. Combining the coordinate consistency and the visibility consistency, we have $O(NM \log(NM))$ as the computational complexity of the global consistency test.

6 EXPERIMENTS

We now report the experimental results on five optically challenging objects shown in Fig. 16. The first object is a stainless steel bowl, the second and the third are ceramic glazed seashell and angel figurines, the fourth is a tray figurine made of multiple surface materials, including specular, Lambertian, and highly absorptive surfaces, and the fifth is a gorilla figurine made of black plastic, which is highly absorptive. We will refer each object as bowl, seashell, angel, tray, and gorilla, respectively.

6.1 Data Acquisition

We acquired 3, 27, 18, 40, and 34 range images, respectively, for the objects shown in Fig. 16. Then, we painted all the objects so that the surfaces of the objects are ideal for range sensing. Range data of the painted objects were acquired and registered. Let P_{ref} be the set of measurements in all the registered range images of a painted object and P_i be a set of measurements from the i th range image acquired for the original object. Note that none of the elimination tests have been applied to P_i yet. Each measurement in P_i is labeled as either *true* or *false* by the following procedures. First, P_i is transformed into the coordinate system of P_{ref} , where the transformation matrix is computed beforehand by registering P_i to P_{ref} . In order to obtain an accurate registration, we used the final result of P_i after the convergence of all the constraint tests. Once P_i is transformed, each measurement in P_i is labeled as true if there is a point in P_{ref} that has the distance less than the range resolution and the angle between surface normals less than 30 degrees. Otherwise, it is labeled as false. Having all the original data labeled as true or false, we can simply keep record of which of the true or false measurements are eliminated during the constraint tests.

6.2 Evaluation of the Local Tests

Fig. 17 shows the distributions of true measurements (blue line with \circ markers) and false measurements (red line with \times markers) for all five objects over four different parameters used in the local tests. Fig. 17a shows the distribution of the number of valid elements in the respective 5×5 fitting window (that is, the parameter m in (2)). Notice that the majority of true measurements has all 25 elements, which is the maximum number for a 5×5 window, and very few of them have less than 20 elements. On the other hand, there is a good number of false measurements, with m ranging from 10 to 20. Thus, a good choice for the threshold τ_m would be from

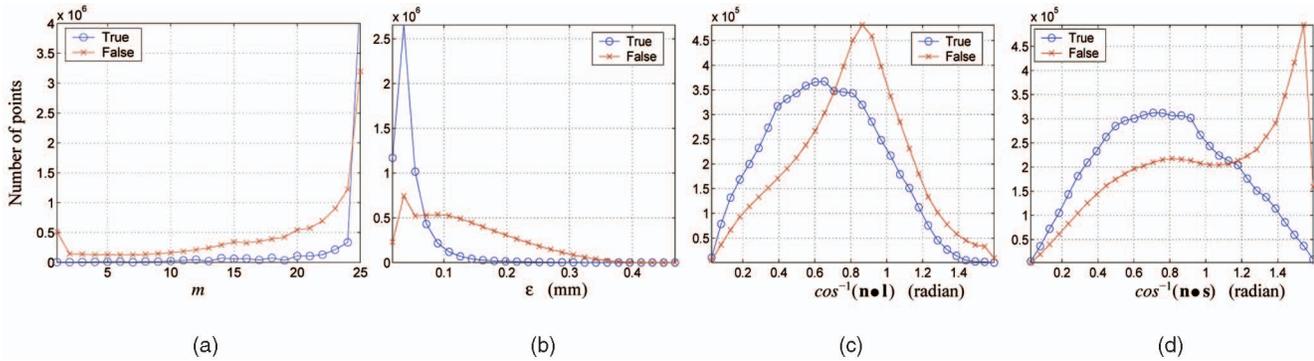


Fig. 17. (a) Distributions of the true measurements (blue line with \circ markers) and the false measurements (red line with \times markers) for all five objects over m , which is the number of valid elements in the fitting window. (b) Distribution over ε , that is, the fitting error of the best fitting plane. (c) Distribution over $\cos^{-1}(\mathbf{n} \bullet \mathbf{l})$, where \mathbf{n} is the unit surface normal vector, and \mathbf{l} is the unit line-of-light vector. (d) Distribution over $\cos^{-1}(\mathbf{n} \bullet \mathbf{s})$, where \mathbf{s} is the unit line-of-sight vector.

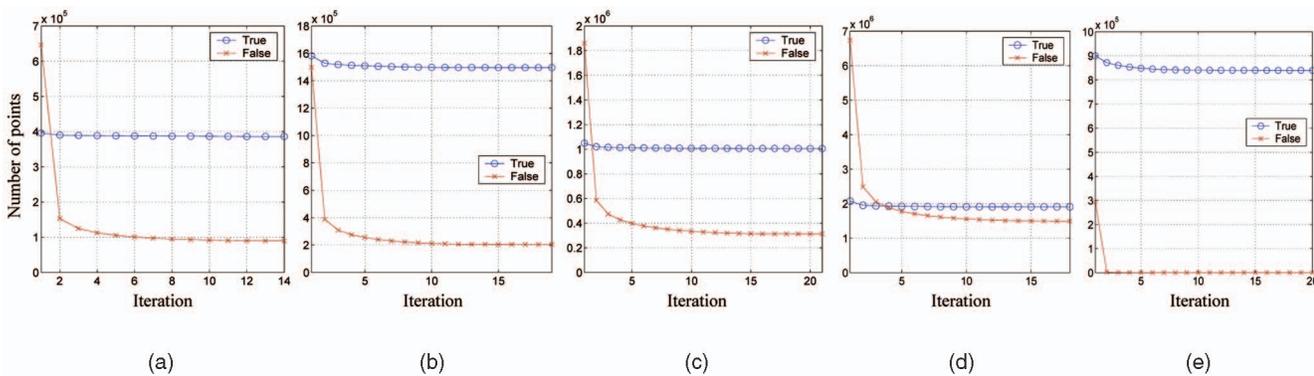


Fig. 18. Number of true measurements and false measurements during the local tests, with the parameters set as $\tau_m = 13$, $\tau_\varepsilon = 0.2$, and $\tau_\theta = 0.1$ (see Sections 5.1 and 5.2 for the descriptions of the parameters). Each iteration indicates both the smoothness test and the local visibility test. (a) Bowl. (b) Seashell. (c) Angel. (d) Tray. (e) Gorilla.

10 to 15 (that is, 40 percent to 60 percent of maximum number). Fig. 17b shows the distribution over the fitting error of the best fitting plane for the elements in the window (that is, the parameter ε in (3)). Most of the true measurements have the fitting error less than 0.1 mm, whereas the false measurements are more evenly distributed. The graph suggests that a good choice for the threshold τ_ε would be from 0.15 mm to 0.3 mm (that is, about 50 percent to 100 percent of sensor resolution). Fig. 17c shows the distribution over the angle between the surface normal and the line-of-light vectors, and Fig. 17d shows the distribution over the angle between the surface normal and the line-of-sight vectors. Observe in Fig. 17d that a large number of false measurements have the surface normals close to orthogonal to the line-of-sight vectors. These measurements correspond to those that we explained in Section 5.2. Since the local visibility test requires both angles $\cos^{-1}(\mathbf{n} \bullet \mathbf{l})$ and $\cos^{-1}(\mathbf{n} \bullet \mathbf{s})$ to be less than $\frac{\pi}{2} - \tau_\theta$, a good choice for the threshold τ_θ would be from 0.05 to 0.15 radians.

Fig. 18 shows the number of true and false measurements during the course of the local tests, with the parameters set as $\tau_m = 13$, $\tau_\varepsilon = 0.2$, and $\tau_\theta = 0.1$. Each iteration indicates both the smoothness test and the local visibility test. Observe that a large number of false measurements were eliminated while maintaining almost all of true measurements. For the case of the gorilla figurine, almost all the false measurements were eliminated by the local tests. Table 1 summarizes the results of the local tests for all five objects.

6.3 Evaluation of the Global Tests

Fig. 19 shows the number of true and false measurements over the course of the global tests. The first row shows the number of true measurements for each of the five objects, and the second row shows the number of false measurements. The global tests were evaluated with six different

TABLE 1
Summary of the Local Tests

	<i>bowl</i>	<i>seashell</i>	<i>angel</i>	<i>tray</i>	<i>gorilla</i>
N	3	27	18	40	34
n_{before}^{total}	1,041,533	3,080,140	2,908,048	8,810,257	1,194,217
n_{after}^{total}	475,822	1,699,937	1,318,631	3,401,468	839,966
n_{before}^{true}	395,665	1,581,899	1,049,448	2,073,857	900,081
n_{after}^{true}	386,248	1,496,095	1,005,916	1,908,005	839,572
n_{before}^{false}	645,868	1,498,241	1,858,600	6,736,400	294,136
n_{after}^{false}	89,574	203,842	312,715	1,493,463	394
I_{avg}	13.0	12.8	13.5	12.4	7.7
TN	0.8613	0.8639	0.8317	0.7783	0.9987
FN	0.0238	0.0542	0.0415	0.0800	0.0672
$time$	52.5 sec	168.9 sec	175.8 sec	506.2 sec	49.6 sec

N is the number of range images acquired. n_{before}^{total} and n_{after}^{total} represent the total number of measurements before and after the local tests applied, respectively, n_{before}^{true} and n_{after}^{true} represent the number of true measurements before and after the local tests applied, and analogously, n_{before}^{false} and n_{after}^{false} for the number of false measurements. I_{avg} is the average of number of iterations in the local tests. TN represents the true negative rate, FN represents the false negative rate, and $time$ is the total execution time performed on a Linux machine with 2.8-GHz Pentium 4 CPU and 1.0-Gbyte RAM.

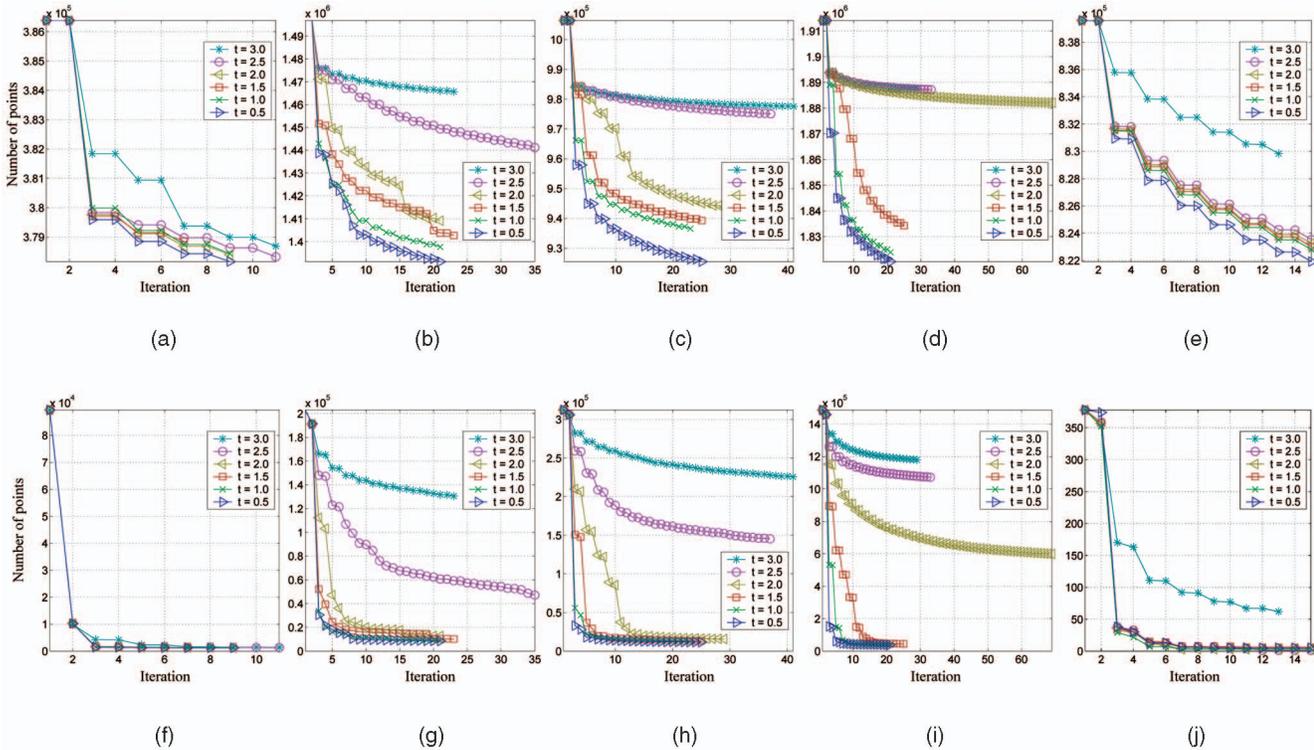


Fig. 19. Number of true and false measurements over the iterations of the global tests evaluated with the varying constant t of (13) and (14). The top row shows the number of true measurements for each of the five objects, and the bottom row shows the number of false measurements. Note that the vertical axis limits for each graph is independent of each other, except that the lower limits for the graphs showing the number of false measurements (second row) are all zeros. (a) Bowl (true). (b) Seashell (true). (c) Angel (true). (d) Tray (true). (e) Gorilla (true). (f) Bowl (false). (g) Seashell (false). (h) Angel (false). (i) Tray (false). (j) Gorilla (false).

values of t , the constant used to determine the thresholds in the global tests in (13) and (14). The odd-numbered iterations represent the isolated region test, and the even-numbered iterations the global consistency test. The initial data was the result after the convergence of the local test. Note that the vertical axis limits for each graph is independent of each other, except that the lower limits for the graphs showing the number of false measurements (second row) are all zeros. As expected, a larger t value maintains more true measurements at the cost of having more false measurements at the convergence. In general, the global tests, regardless of the t value, were able to preserve most of the true measurements, whereas only the tests with the t value less than 2.0 were able to eliminate the false measurements reasonably well.

TABLE 2
Summary of the Global Tests

	<i>bowl</i>	<i>seashell</i>	<i>angel</i>	<i>tray</i>	<i>gorilla</i>
N	3	27	18	40	34
n_{before}^{total}	475,822	1,699,937	1,318,631	3,401,468	839,966
n_{after}^{total}	380,166	1,412,863	952,625	1,878,613	823,003
n_{before}^{true}	386,248	1,496,095	1,005,916	1,908,005	839,572
n_{after}^{true}	378,849	1,402,687	939,227	1,834,272	822,997
n_{before}^{false}	89,574	203,842	312,715	1,493,463	394
n_{after}^{false}	1,317	10,176	13,398	44,341	6
I	6	22	24	24	14
TN	0.9853	0.9501	0.9572	0.9703	0.9848
FN	0.0192	0.0624	0.0663	0.0386	0.0197
<i>time</i>	33.4 sec	1113.5 sec	641.7 sec	3322.6 sec	398.8 sec

Table 2 summarizes the results of the global tests for all five objects, with the t value set as 1.5. The notations in the table are similar to those used in Table 1, except that I is simply the number of iterations in the global tests. Generally, the global tests took more time compared to the local tests, and for all objects, the true negative rates were greater than 0.95 while maintaining the false negative rates under 0.07.

In Fig. 20, the images in the first row show the original data for all five objects, where the light blue points represent the true measurements, and the red points represent the false measurements. The second row shows the data after the convergence of the local tests, and the third row shows the color-coded global consistency value \mathcal{G} after the convergence of the local tests. Similarly, the fourth and the fifth rows show after the convergence of the global tests. Finally, the images in the last two rows show the final integrated 3D models. The final 3D models capture the fine details of the objects, even in the regions of severe concavities, where mutual reflections are highly likely to occur, as well as the regions of highly absorptive surfaces.

7 CONCLUSION AND FUTURE WORK

In this paper, we presented a new method for constructing accurate 3D models of real-world objects that are optically challenging, such as those containing specular and highly absorptive surfaces. The multippeak range imaging is used to capture the erroneous range data. Subsequently, the false measurements are eliminated by applying a series of constraint tests. By obtaining the ground-truth data for

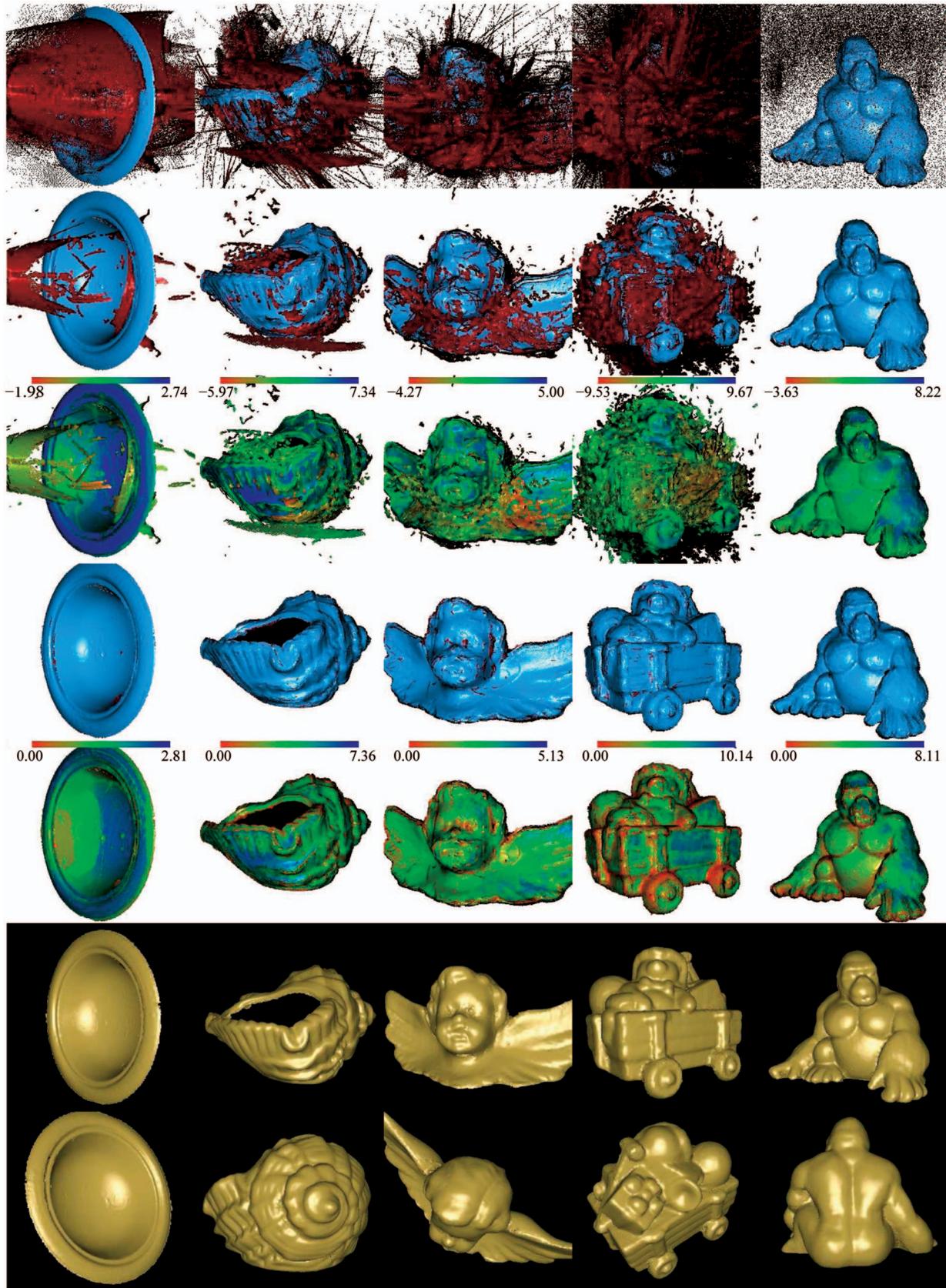


Fig. 20. The images in the first row show the original data for all five objects, where the light blue points represent the true measurements, and the red points represent the false measurements. The second row shows the data after the convergence of the local tests, and the third row shows the color-coded global consistency value \mathcal{G} after the convergence of the local tests. Similarly, the fourth and the fifth rows show the consistency value after the convergence of the global tests. Finally, the images in the last two rows show the final integrated 3D models.

each of the objects used in the experiment, we are able to analyze the performance of each constraint test and also justify the selection of the thresholds used in the constraint tests. The experimental results indicate that our method significantly improves upon the traditional methods for generating accurate 3D models of real-world objects with optically challenging surface properties.

We envision a number of directions for future work. First, a more sophisticated peak detection algorithm needs to be developed to account for various light reflection properties that depend on the material and the shape of objects. One of the most difficult problems is the case when multiple peaks are too close to one another that the borders between the peaks are no longer seen by the intensity change alone. When two peaks are treated as one, the image coordinates of the estimated center will be shifted, as well as, consequently, the 3D coordinates in the world frame. We have found that most of the secondary reflections behave just like the primary reflections in that they stay around the same position or steadily move to a certain direction throughout the image sequence, as opposed to suddenly appearing and disappearing at random positions. By performing a segmentation in each image and by keeping track of the position and the direction of each region throughout the images in the sequence, it may be possible to detect cases where two separate regions are getting close to each other and eventually merge into a single region. If we can detect the merge between separate regions, that is, separate peaks, then it may be possible to compute more accurate centers of the peaks, even when the peaks are very close to each other.

Another goal for future work is to improve the threshold selection in the constraint tests so that the elimination of false measurements is maximized while preserving all true measurements. Although we showed the effects of varying the parameters in the constraint tests and suggested the range of a good choice for each of the parameters, one can certainly compute the best parameters for a given particular data. For example, the constant t of (13) can be set dynamically in each iteration, depending on the current status of the data. Perhaps, instead of using a single t value for the entire data, one can divide the data by using regular 3D blocks and assign different values of t for each block according to the contents of the data inside the block.

Finally, extending our methods to other types of optically challenging surfaces such as translucent or transparent surfaces remains an important area for future work.

REFERENCES

- [1] B. Curless and M. Levoy, "A Volumetric Method for Building Complex Models from Range Images," *Proc. ACM SIGGRAPH '96*, pp. 303-312, 1996.
- [2] K. Pulli, "Multiview Registration for Large Data Sets," *Proc. Second Int'l Conf. 3D Digital Imaging and Modeling (3DIM '99)*, pp. 160-168, 1999.
- [3] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk, "The Digital Michelangelo Project: 3D Scanning of Large Statues," *Proc. ACM SIGGRAPH '00*, pp. 131-144, 2000.
- [4] *Modeling from Reality*, K. Ikeuchi and Y. Sato, eds. Kluwer Academic Publishers, 2001.
- [5] H.P.A. Lensch, J. Kautz, M. Goesele, W. Heidrich, and H.-P. Seidel, "Image-Based Reconstruction of Spatially Varying Materials," *Proc. 12th Eurographics Workshop Rendering*, 2001.
- [6] F. Bernardini, I. Martin, J. Mittleman, H. Rushmeier, and G. Taubin, "Building a Digital Model of Michelangelo's Florentine Pietà," *IEEE Computer Graphics and Applications*, vol. 22, no. 1, pp. 59-67, 2002.
- [7] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy, "Real-Time 3D Model Acquisition," *ACM Trans. Graphics*, vol. 21, no. 3, pp. 438-446, 2002.
- [8] D. Huber and M. Hebert, "Fully Automatic Registration of Multiple 3D Data Sets," *Image and Vision Computing*, vol. 21, no. 7, pp. 637-650, July 2003.
- [9] A. Kalaiah and A. Varshney, "Modeling and Rendering of Points with Local Geometry," *IEEE Trans. Visualization and Computer Graphics*, vol. 9, no. 1, pp. 30-42, 2003.
- [10] Y. Sato, M. Wheeler, and K. Ikeuchi, "Object Shape and Reflectance Modeling from Observation," *Proc. ACM SIGGRAPH '97*, pp. 379-387, 1997.
- [11] T. Masuda, "Generation of Geometric Model by Registration and Integration of Multiple Range Images," *Proc. Third Int'l Conf. 3D Digital Imaging and Modeling (3DIM '01)*, pp. 254-261, 2001.
- [12] P.J. Besl, "Active Optical Range Imaging Sensors," *Advances in Machine Vision*, Springer-Verlag, pp. 1-63, 1989.
- [13] 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, 1992.
- [14] Z. Zhang, "Iterative Point Matching for Registration of Free-Form Curves and Surfaces," *Int'l J. Computer Vision*, vol. 13, no. 2, pp. 119-152, 1994.
- [15] G. Godin, D. Laurendeau, and R. Bergevin, "A Method for the Registration of Attributed Range Images," *Proc. Third Int'l Conf. 3D Digital Imaging and Modeling (3DIM '01)*, pp. 179-186, 2001.
- [16] R. Bergevin, M. Soucy, H. Gagnon, and D. Laurendeau, "Towards a General Multiview Registration Technique," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 5, pp. 540-547, 1996.
- [17] R. Benjemaa and F. Schmitt, "Fast Global Registration of 3D Sampled Surfaces Using a Multi-Z-Buffer Technique," *Proc. Conf. Recent Advances in 3-D Digital Imaging and Modeling (3DIM '97)*, pp. 113-120, 1997.
- [18] I. Stamos and M. Leordeanu, "Automated Feature-Based Range Registration of Urban Scenes of Large Scale," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 555-561, June 2003.
- [19] N. Gelfand, N. Mitra, L. Guibas, and H. Pottmann, "Robust Global Registration," *Proc. Third Symp. Geometry Processing (SGP '05)*, 2005.
- [20] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface Reconstruction from Unorganized Points," *Proc. ACM SIGGRAPH '92*, pp. 71-78, 1992.
- [21] W.E. Lorensen and H.E. Cline, "Marching Cubes: A High Resolution 3D Surface Construction Algorithm," *Proc. ACM SIGGRAPH '87*, pp. 163-169, 1987.
- [22] 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, 1995.
- [23] G. Turk and M. Levoy, "Zippered Polygon Meshes from Range Images," *Proc. ACM SIGGRAPH '94*, pp. 311-318, 1994.
- [24] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin, "The Ball-Pivoting Algorithm for Surface Reconstruction," *IEEE Trans. Visualization and Computer Graphics*, vol. 5, no. 4, pp. 349-359, 1999.
- [25] B. Curless and M. Levoy, "Better Optical Triangulation through Spacetime Analysis," *Proc. Fifth IEEE Int'l Conf. Computer Vision (ICCV '95)*, pp. 987-994, 1995.
- [26] S. Nayar, K. Ikeuchi, and T. Kanade, "Recovering Shape in the Presence of Interreflections," *Proc. IEEE Int'l Conf. Robotics and Automation (ICRA '91)*, vol. 2, pp. 1814-1819, Apr. 1991.
- [27] J. Clark, E. Trucco, and L.B. Wolff, "Using Light Polarization in Laser Scanning," *Image and Vision Computing*, vol. 15, no. 1, pp. 107-117, 1997.
- [28] E. Trucco and R.B. Fisher, "Acquisition of Consistent Range Data Using Local Calibration," *Proc. IEEE Int'l Conf. Robotics and Automation (ICRA '94)*, pp. 3410-3415, 1994.
- [29] J. Park and A. Kak, "Multi-Peak Range Imaging for Accurate 3D Reconstruction of Specular Objects," *Proc. Sixth Asian Conf. Computer Vision (ACCV '04)*, 2004.

- [30] J. Park and A. Kak, "Specularity Elimination in Range Sensing for Accurate 3D Modeling of Specular Objects," *Proc. Second Int'l Symp. 3D Data Processing, Visualization, and Transmission (3DPVT '04)*, 2004.
- [31] J. Shade, S. Gortler, L.-W. He, and R. Szeliski, "Layered Depth Images," *Proc. ACM SIGGRAPH '98*, pp. 231-242, 1998.
- [32] P.J. Besl and R.C. Jain, "Segmentation through Variable-Order Surface Fitting," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 10, no. 2, pp. 167-192, 1988.
- [33] C.H. Chen and A.C. Kak, "3D-POLY: A Robot Vision System for Recognizing Objects in Occluded Environments," Technical Report TR-EE 88-48, Purdue Univ., 1988.
- [34] A. Rosenfeld and A.C. Kak, *Digital Picture Processing*. Academic Press, 1982.
- [35] M. Oren and S.K. Nayar, "Generalization of the Lambertian Model and Implications for Machine Vision," *Int'l J. Computer Vision*, vol. 14, no. 2, pp. 227-251, 1995.
- [36] M.A. Fischler and R.C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Comm. ACM*, vol. 24, no. 6, pp. 381-395, 1981.
- [37] A. Stoddart, S. Lemke, A. Hilton, and T. Penn, "Estimating Pose Uncertainty for Surface Registration," *Proc. British Machine Vision Conf. (BMVC '96)*, pp. 23-32, 1996.
- [38] K.N. Kutulakos and S.M. Seitz, "A Theory of Shape by Space Carving," *Int'l J. Computer Vision*, vol. 38, no. 3, pp. 199-218, 2000.
- [39] K. Pulli, M. Cohen, T. Duchamp, H. Hoppe, L. Shapiro, and W. Stuetzle, "View-Based Rendering: Visualizing Real Objects from Scanned Range and Color Data," *Proc. Eighth Eurographics Workshop Rendering*, pp. 23-34, 1997.
- [40] J.L. Bentley, "Multidimensional Binary Search Trees Used for Associative Searching," *Comm. ACM*, vol. 18, no. 9, pp. 509-517, Sept. 1975.
- [41] J.G. Cleary, "Analysis of an Algorithm for Finding Nearest Neighbors in Euclidean Space," *ACM Trans. Math. Software*, vol. 5, no. 2, pp. 183-192, June 1979.
- [42] J.H. Friedman, J.L. Bentley, and R.A. Finkel, "An Algorithm for Finding Best Matches in Logarithmic Expected Time," *ACM Trans. Math. Software*, vol. 3, no. 3, pp. 209-226, Sept. 1977.



Johnny Park received the BS, MS, and PhD degrees from the School of Electrical and Computer Engineering, Purdue University, in 1998, 2000, and 2004, respectively. Since 2004, he has been a principal research scientist at Purdue University. His current research interests are distributed and collaborative information processing in sensor networks and various topics in computer vision and robotics, including human posture estimation, real-time 3D reconstruction, visual servoing, and 3D object recognition. He is a member of the IEEE.



Avinash C. Kak is a professor of electrical and computer engineering at Purdue University. His research and teaching include sensor networks, computer vision, robotics, and high-level computer languages. He is a coauthor of *Principles of Computerized Tomographic Imaging*, which was republished as a classic in applied mathematics by SIAM, and of *Digital Picture Processing*, which is also considered by many to be a classic in computer vision and image processing. His latest book *Programming with Objects* (John Wiley & Sons, 2003) is used by a number of leading universities as a text on object-oriented programming. His forthcoming book *Scripting with Objects*, also to be published by John Wiley, focuses on object-oriented scripting. These are two of the three books for an "Objects Trilogy" that he is creating. The last, expected to be finished sometime in 2008, will be titled *Designing with Objects*.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.