Vision system for fast 3-D model reconstruction

Huei-Yung Lin, MEMBER SPIE
Murali Subbarao
State University of New York at Stony Brook
Computer Vision Laboratory
Department of Electrical and Computer Engineering
Stony Brook, New York 11794-2350
E-mail: hylin@ece.sunysb.edu

Abstract. A desktop vision system is presented for complete 3-D model reconstruction. It is fast (3-D reconstruction in under 20 min), low cost (uses a commercially available digital camera and a rotation stage), and accurate (about 1 part in 500 in the working range). Partial 3-D shapes and texture information are acquired from multiple viewing directions using rotational stereo and shape-from-focus (SFF). The resulting range images are registered to a common coordinate system, and a surface representation is created for each range image. The resulting surfaces are integrated using an algorithm named region of construction. Unlike previous approaches, the region of construction algorithm directly exploits the structure of the raw range images. The algorithm determines regions in the range images corresponding to nonredundant surfaces that can be stitched along the boundaries to construct a complete 3-D surface model. The algorithm is computationally efficient and less sensitive to registration error. It also has the ability to construct complete 3-D models of complex objects with holes. A textured 3-D model is obtained by mapping texture information onto the complete surface model representing the 3-D shape. Experimental results for several real objects are presented.
[DOI: 10.1117/1.1758731]

Subject terms: 3-D model reconstruction; multiple view registration; multiple view integration; texture mapping; shape from focus; rotational stereo; range image.

Paper 030240 received May 20, 2003; revised manuscript received Oct. 21, 2003; accepted for publication Feb. 12, 2004.

1 Introduction

Reconstruction of 3-D computer models from existing objects has applications in many areas such as reverse engineering (reverse CAD), pattern recognition, and industrial inspection. In the past, 3-D models of real objects were often created manually by a time consuming and expensive process. Recently, the availability of fast and inexpensive graphics hardware, and technologies such as VRML-ready Internet browsers, have made the automatic reconstruction of 3-D models an important research topic in both computer vision and computer graphics areas. In particular, techniques that use low-cost equipment have attracted the attention of many researchers.

As shown in Fig. 1, there are four major steps in the reconstruction of a complete 3-D model[1,2]:

- data acquisition
- registration
- surface integration
- texture mapping.

The data acquisition stage consists of acquiring the partial 3-D shapes and the corresponding texture information of an object. Datasets from multiple viewpoints are needed to completely describe an object. The acquired range images are registered to a common coordinate system based on their acquisition viewing directions. The registered range images are then integrated into a single surface representation. Finally, the texture information is mapped onto the surface to create a textured 3-D model.

We present a vision system that includes these four stages to reconstruct complete 3-D models of real objects. The object is placed on a rotation stage in front of a stationary camera. In the data acquisition stage, input image sequences from different viewpoints are acquired by rotating the object with known rotation angles. Range and the corresponding focused images are recovered using *rotational stereo*[3] and *shape from focus* (SFF).[4,5] The range images are then registered to the camera coordinate system according to the rotation axis and their acquisition viewpoints. A fast and robust registration algorithm is developed to refine the rotation axis derived from system calibration. It updates the translation vector iteratively and converges very fast, even under poor initial estimations. In the surface integration stage, a new algorithm named *region of construction* is used to *stitch* the partial 3-D models from different viewpoints. Unlike the previous approaches,[6–8] the raw data of range images is directly accessed to create nonoverlapping regions for integration. It takes advantage of the known topology of each region of construction to perform fast triangulation. Each range image corresponding to a region of construction is used in integration without any modification (such as weighted summation of overlapping range images from adjacent views). Therefore the registration errors are always limited to the boundaries of region of construction. This algorithm is also extended to handle complex objects with holes and multiple-object scene. The integration method is computationally efficient in the sense that no searching is required for mesh triangulation. Finally, the focused images recovered by SFF are mapped onto the

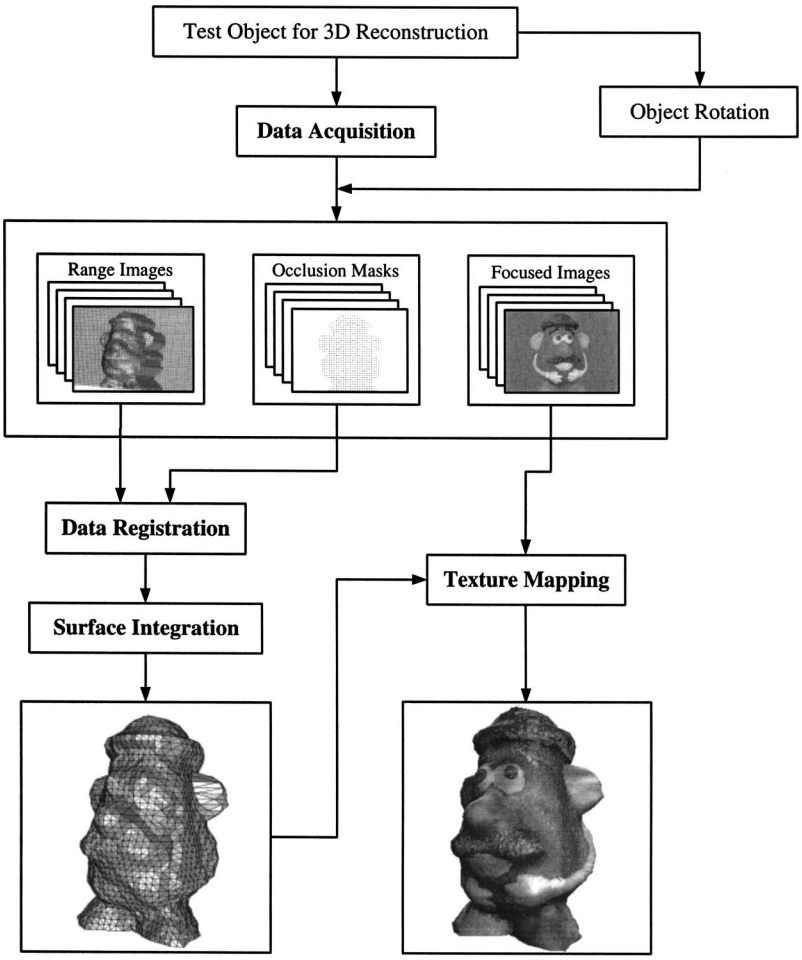

Fig. 1 System flow chart.

reconstructed wireframe model to create a textured 3-D model.

2 Previous Work

Most previous works on complete 3-D model reconstruction from multiple views assume that the range data is given. They focus on algorithms for data registration and integration steps. But the data acquisition step, which includes range image recovery and texture maps from intensity images, is a core problem in computer vision research. Some popular techniques are shape from stereo, shape from motion, structured light analysis, etc.[9] However, we limit our literature review to registration, integration, and complete systems for 3-D model reconstruction.

2.1 *Registration*

A popular method for refining a given registration is the iterative closest point (ICP) technique, first introduced by Besl and McKay.[10] It uses a nonlinear optimization procedure to further align the datasets from initial registration. Given two datasets and a rough initial transformation, potential correspondences are developed as follows. For each point in the first dataset, we find its closest point in the second dataset under the current transformation. Let P be the set of m points from the first dataset $\{p_1, p_2, \ldots, p_m\}$ and Q be the set of n points from the second dataset $\{q_1, q_2, \ldots, q_n\}$. An incremental transformation T_i is defined such that $Q_{i+1} = T_i Q_i$ for each iteration i, where $Q_0 = Q$. T_i reduces the registration error between P and the new point set Q_{i+1} by moving the points Q_i closer to P. Then Q_i is assigned to Q to initialize the system and update the next transformation T_{i+1}. The transformation usually involves a translation vector and a rotation matrix. The ICP algorithm repeatedly computes the closest points between datasets and computes the transformation to register the data, until a minimum tolerance on a mean square distance metric between the surfaces is obtained. They showed that the ICP algorithm will monotonically converge to a minimum, but it is not necessarily the global or the best minimum. To converge to the global minimum, the initial parameter estimate must be reasonably close to the true value to avoid converging to a nonoptimal solution.

Chen and Medioni[1] also use an iterative refinement of initial registrations between views to perform fine registration. A good initial estimate of the registration is assumed to be available. Instead of minimizing the distance between the closest points in the two datasets (such as general ICP algorithms), orientation information is used for minimization. Their technique refines a nominal transformation by iteratively computing incremental transformations that

minimize the distance between the transformed points from the first view and the points from the second view. The minimized functional is expressed in terms of the distance between each *control point* in the first view, and the tangent plane at an intersected point in the second view. This will minimize the distance between the two surfaces as well as match their local shape characteristics. Because of the expense of computing intersections between lines and the discrete surface (generated from the tangent planes), they subsample the original data to obtain a set of control points. Since the surface normals can be computed more accurately in smooth areas, the control points are selected from this subsampling of the surface (regularized grid) that lies in smooth regions on the surface.

2.2 *Integration*

Surface reconstruction from range data has been extensively investigated over the past several decades. The methods in literature that are able to create seamless meshes from 3-D range datasets can be divided into two groups, the *volumetric approach* and the *surface approach*. The volumetric approaches[7,11–14] first store the 3-D data points in a volumetric data structure, either a voxel grid or an octree. Each voxel contains eight vertices, and by evaluating a field function at those vertices, it is possible to extract a level surface of the field function. The field function is often chosen as a signed distance function. The triangular mesh is then created using an isosurface extraction algorithm such as the *marching cubes* algorithm.[15] The surface approaches[16–18] create an initial set of triangulated meshes from the original 3-D range images. These meshes are then merged together to create the final complete 3-D model.

Volumetric approaches work on both structured and unstructured input data. Hoppe et al.[7] introduce an algorithm to construct 3-D surface models from a cloud of unorganized points without spatial connectivity. They determine an approximate tangent plane at each data point using least squares on k nearest neighbors, and then take the signed distance to the nearest point's tangent plane as the distance function in 3-D space. The distance function is then interpolated and polygonalized by the marching cubes algorithm. In two subsequent steps,[19,20] the constructed mesh is optimized (i.e., the number of triangles is reduced while the distance of the mesh from the data points is kept small), and then a piecewise smooth subdivision surface is built on it.

Curless and Levoy[12] propose an approach similar to Hoppe's algorithm with a few differences. They integrate distance estimates at each voxel instead of searching for the closest point from a voxel's center to determine the signed distance. The range images are taken separately and scanned along the line of sight to each of them. The integration is done on the signed distance to the point for each voxel the line passes through. The final signed distance estimate is a weighted average of all the estimates from different range images. The marching cube algorithm is then used to approximate the zero set of the distance function by a set of connected triangles using the values of the distance function at the voxel vertices.

Pulli et al.[2] propose a simpler and more attractive method for a volumetric approach. They classify the voxels in the volumetric grid for each range image as either outside, inside, or on the surface of the object, by using geometric properties about the viewing angle. Their algorithm recursively subdivides each voxel to be classified on the surface into eight smaller voxels, and the process is stored in an octree data structure. By some simple rules, the classifications of the voxels from different range images are combined into one common classification. Those rules are also used to remove the outliers. When a predefined level of refinement is reached, a triangular mesh is then constructed from those voxels.

Turk and Levoy[16] propose a polygonal method that fits a triangular mesh to each range image. They employ an incremental algorithm that updates a reconstruction by eroding redundant data, followed by ''zippering'' along the remaining boundaries. The final consensus step reintroduces the original geometry to establish final vertex positions. Their method utilizes the structure in each range image, but could have bad behavior in areas of high curvature.

Soucy and Laurendeau[18,21] describe a method that builds surface descriptions from multiple registered range images using Venn diagrams. They decompose range images into canonical views, which are areas common to a unique subset of range images. A common reference plane is defined for each canonical view, and the data points are projected onto it. Sets of common points are found using neighborhood and visibility tests. A Delaunay triangulation is made on each reference plane, and they are combined to form the complete model by reparameterization.

2.3 *Complete 3-D Acquisition System*

Pulli et al.[22] present a complete system for scanning the range and color information of a 3-D object and for displaying realistic images of the object from arbitrary viewpoints. They build a range and color image scanner with four digital cameras and a slide projector placed on a computer-controlled turntable. A vertical stripe of white light is emitted from the slide projector into the working volume to produce several views of dense range data. The data are registered, and a surface that approximates the data is constructed. One major advantage of their approach is that the surface estimate obtained from range scanner can be fairly coarse. The appearance of fine detail is recreated by view-dependent texturing of the surface using color images.

Reed and Allen[8] describe a system that builds a 3-D CAD model of an object incrementally from multiple range images. They built a robotic system that consists of a laser rangefinder attached to a robot arm to acquire range images of the object. A volume-based method using a constructive solid geometry (CSG) technique to form a solid model of the object from multiple registered range views is developed. They create a solid for each view by sweeping the range data from the object away from the scanner filling in the space self-occluded by the object. The surfaces of the solid are labeled as being either visible to the sensor or occluded by the object view. This labeling technique is used for view planning, since it provides information about which portions of the current viewspace are not covered by the previous views. Once all the visible portions of the object have been covered by one or more views, the view solid models are intersected to form a complete model.

Niem[23] presents a method for the automatic reconstruction of 3-D objects from multiple camera views using a mobile camera. He uses a simple measurement environment that consists of a new calibration pattern placed below the object to increase the flexibility of the system. With the simultaneous acquisition of object and pattern, each view can be calibrated individually. From those calibrated camera views, a textured 3-D model is estimated using a shape-from-silhouette approach and texture mapping of the original camera views. The major drawback of this system is that some concavities may not be fully recovered because of the shape-from-silhouette technique.

Albamont and Goshtasby[24] developed a scanner system that can capture surround images of an object and reconstruct its 3-D model. The scanner has four synchronous camera heads, each equipped with a camera and a laser line generator. The four camera heads move together during the scanning process. The scanner uses an imaginary (virtual) laser rather than a real one for scanning. This makes it possible to scan an object with detailed color and under bright lighting. The laser images from the four cameras are processed to obtain the 3-D structure of the object.

3 Data Acquisition

To acquire the range and intensity images, the object is placed on a rotation stage. The images from different acquisition viewpoints are taken by rotating the object. For each viewpoint, range and focused images are obtained using *rotational stereo* and *shape from focus* (SFF). Two sequences of images with different focus positions are taken with a small rotation angle to obtain stereo image pairs. Each sequence of images is used to construct the focused image and a rough depth map using SFF. A more accurate 3-D shape is then obtained using rotational stereo on the focused image pair with the rough depth map information.

3.1 *Shape from Focus*

In SFF, a large sequence of image frames of a 3-D scene is recorded with different camera parameters (e.g., focal length or/and lens to image detector distance). In each image frame, different objects in the scene will be blurred by different degrees, depending on their distance from the camera lens. Each object will be in best focus in only one image frame in the image sequence. The entire image sequence is processed to find the best focused image of each object in the 3-D scene. The distance of each object in the scene is then found from the camera parameters that correspond to the image frame that contains the best focused image of the object. The SFF methods are based on the fact that for an aberration-free convex lens, 1. the radiance at a point in the scene is proportional to the irradiance at its *focused image*[25] (photometric constraint), and 2. the position of the point in the scene and the position of its focused image are related by the *lens formula* (geometric constraint)

$$\frac{1}{f}=\frac{1}{u}+\frac{1}{v}, \tag{1}$$

where f is the focal length, u is the distance of the object from the lens plane, and v is the distance of the focused image from the lens plane (see Fig. 2).

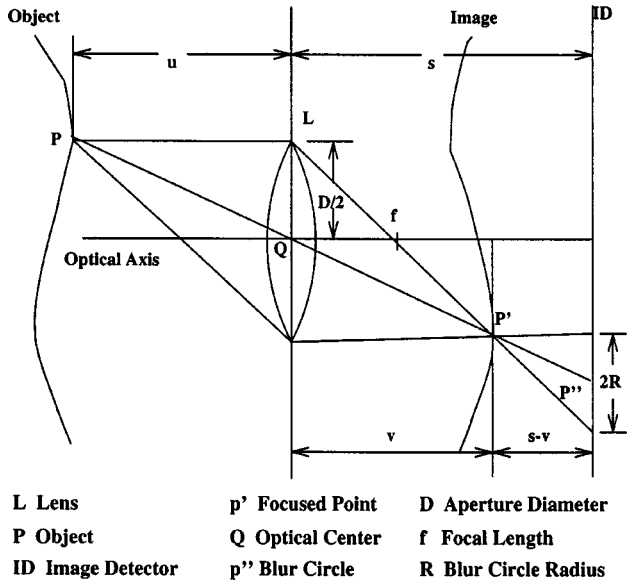

Fig. 2 Image formation in a convex lens.

Given the irradiance and the position of the focused image of a point, its radiance and position in the scene are uniquely determined. In a sense, the positions of a point object and its image are *interchangeable*, i.e., the image of the image is the object itself. Now, if we think of an object surface in front of the lens to be comprised of a set of points, then the focused images of these points define another surface behind the lens (see Fig. 2). This surface is defined to be the *focused image surface* (FIS) and the image irradiance on this surface to be the *focused image*. There is a *one-to-one correspondence* between the FIS and the object surface. The geometry (i.e., the 3-D shape information) and the radiance distribution (i.e., the photometric information) of the object surface are uniquely determined by the FIS and the focused image.

In traditional SFF methods (e.g., Refs. 4, 5, and 26), a sequence of images is obtained by continuously varying the distance s between the lens and the image detector or/and the focal length f (see Fig. 3). For each image in the sequence, a focus measure is computed at each pixel (i.e., each direction of view) in a small (about 15×15) image neighborhood around the pixel. At each pixel, that image frame among the image sequence that has the maximum focus measure is found by a search procedure. The gray level (which is proportional to image irradiance) of the pixel in the image frame thus found gives the gray level of the focused image for that pixel. The values of s and f for this image frame are used to compute the distance of the object point corresponding to the pixel. An example of a focus measure is the gray-level variance. SFF methods involve a search for the values of s or/and f that result in a maximum focus measure, and these methods require the acquisition and processing of a large number of images.

3.2 *Rotational Stereo*

The rotational stereo model used in data acquisition is shown in Fig. 4. The rotation axis is described by the unit vector $\vec{\mathbf{n}}=(n_1,n_2,n_3)^T$ and the translation vector $\vec{\mathbf{d}}$

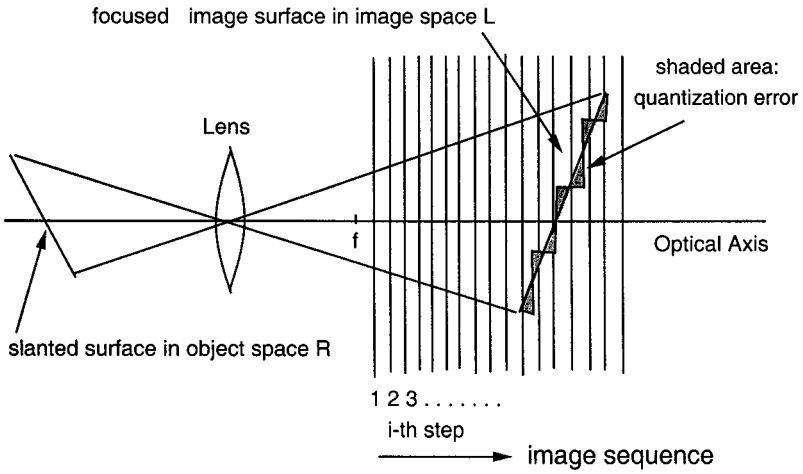

Fig. 3 Focus image surface.

$=(d_1,d_2,d_3)^T$ in the camera coordinate system. The image pair used for stereo matching is obtained by rotating the object an angle θ with respect to the rotation axis. Let (x_1,y_1,z_1), (x_2,y_2,z_2) be the same object point before and after rotation, respectively; and $(\hat{x}_1,\hat{y}_1)$, $(\hat{x}_2,\hat{y}_2)$ denote the corresponding image point. The images taken before and after rotation are referred to as the first and second image. For each point $(\hat{x}_1,\hat{y}_1)$ in the first image, the corresponding *epipolar line* in the second image is calculated and used for stereo matching.

Let $\vec{\mathbf{n}}$ be a unit vector along the selected rotation axis and θ be the specified rotation angle about this axis. As shown in Ref. 9, the rotation matrix can be written as

$$\mathbf{M}_R(\theta)=(1-\cos\theta)\begin{bmatrix} n_1^2 & n_1n_2 & n_1n_3 \\ n_2n_1 & n_2^2 & n_2n_3 \\ n_3n_1 & n_3n_2 & n_3^2 \end{bmatrix} + \sin\theta\begin{bmatrix} 0 & -n_3 & n_2 \\ n_3 & 0 & -n_1 \\ -n_2 & n_1 & 0 \end{bmatrix} + I\cos\theta.$$

The rotation matrix for any rotation axis with a translation vector $\vec{\mathbf{d}}$ is then

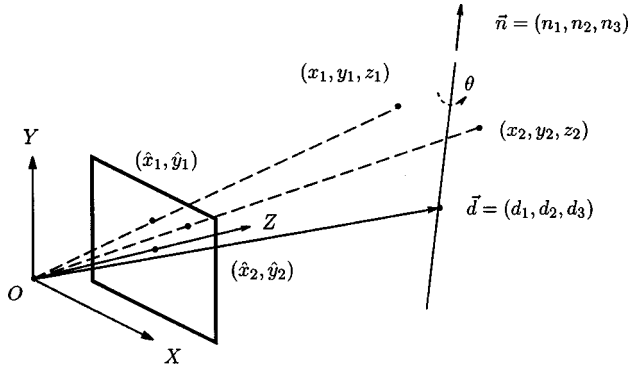

Fig. 4 Rotational stereo model.

$$\mathbf{R}(\theta)=\mathbf{T}^{-1}\cdot\begin{bmatrix}\mathbf{M}_R(\theta) & 0\\ 0 & 1\end{bmatrix}\mathbf{T}, \tag{2}$$

where

$$\mathbf{T}=\begin{bmatrix}1&0&0&d_1\\0&1&0&d_2\\0&0&1&d_3\\0&0&0&1\end{bmatrix}.$$

By perspective projection, the relationship between an object point (x_i,y_i,z_i) and an image point $(\hat{x}_i,\hat{y}_i)$ is given by

$$x_i=\hat{x}_it_i,\quad y_i=\hat{y}_it_i,\quad \text{and}\quad z_i=ft_i,$$

where t_i's are unknown parameters. Thus, we have the equations

$$\begin{pmatrix}\hat{x}_2t_2\\ \hat{y}_2t_2\\ ft_2\\ 1\end{pmatrix}=\mathbf{R}(\theta)\cdot\begin{pmatrix}\hat{x}_1t_1\\ \hat{y}_1t_1\\ ft_1\\ 1\end{pmatrix}, \tag{3}$$

for the same object point projected on different images. Solving Eqs. (2) and (3), the epipolar line on the second image for any fixed $(\hat{x}_1,\hat{y}_1)$ is given by

$$\hat{y}_2=m\hat{x}_2+c, \tag{4}$$

where m and c are functions of $\hat{x}_1$ and $\hat{y}_1$. Details of the calculation can be found in Ref. 27.

In our rotational stereo model, the stereo matching is done along the epipolar line derived before at 16×16 image block intervals for only the foreground pixels, as determined by SFF. Therefore image rectification is avoided.

3.3 *Rotation Axis Calibration*

A simple method is used to calibrate the rotation axis. It takes two images of a planar object before and after rotating the object. Two 3-D point pairs with known coordinates

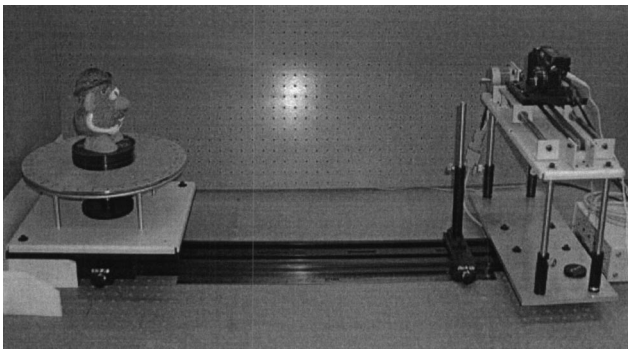

Fig. 5 SVIS-2 camera system.

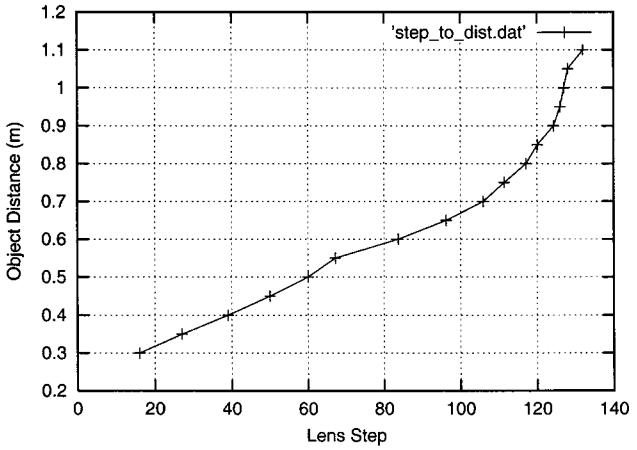

Fig. 6 Lens step versus best focused distance used in SFF.

are used to compute the rotation axis. Let P_1, Q_1 be the 3-D points before rotation, and P_2, Q_2 be the corresponding 3-D points after rotation. Let Π_P be the plane passing through the middle point of P_1 and P_2, and perpendicular to $\overline{P_1P_2}$; and Π_Q be the plane passing through the middle point of Q_1 and Q_2, and perpendicular to $\overline{Q_1Q_2}$. Then the rotation axis is uniquely determined by the intersection of Π_P and Π_Q. Another calibration method using one known 3-D point with two rotation angles can be found in Ref. 28.

Although only four 3-D points are required for determining the rotation axis, one can increase the accuracy of the calibration by taking more points and using a least squared fit. In the experiments, the rotation axis is found to be (0.218,0,831.774) for the translation vector and (4.123×10^{-3}, 0.9997, -2.364×10^{-2}) for the unit vector. The result indicates that the rotation axis is almost perpendicular to the image scanlines and camera optical axis.

3.4 *Camera System and Implementation*

The vision system used for image acquisition in our experiment is called Stonybrook Vision System 2 (SVIS-2). The SVIS-2 camera system consists of a high-resolution digital camera (Olympus C-3030), a rotation stage with a stepper motor, and a PC (see Fig. 5). The object is mounted on the rotation stage and the camera is placed in front of it, so that the optical axis is close to the rotation axis. The calibration method described in the previous section is used for rotation axis estimation. The calibrated rotation axis is used for stereo matching in rotational stereo, and is further refined during registration. The parameters of SVIS-2 are adjusted for objects that fit inside a $250\times250\times250$-mm cube placed about 750 mm from the camera. The camera focal length is set to 19.35 mm and the f-number to 1.8. Different focus settings are obtained by moving the camera's motorized lens controlled by a PC. The image resolution is set to 1280×960 pixels.

In the experiments, we use objects with fine texture to help stereo matching. The steps for recovering 3-D shape and focused image of each view of an object are described as follows. First, two sequences of four images with different focus settings are recorded before and after rotating the object by a small angle to obtain the stereo image pairs. The stereo rotation angle is set to 6 deg, which gives the equivalent parallel stereo baseline of about 174 mm. Shape from focus is applied on each sequence of images to get the focused image and a rough depth map. 16×16 image blocks are used to obtain a 80×60 rough depth map and a 1280×960 focused image. The depth map is thresholded to segment both the depth map and the focused image into two regions, one corresponding to the background region (points farther than the expected distance of object points) and the other to the object (foreground) region.

Figure 6 shows a plot of the relationship between the lens step number and the best focused distance of the camera used in our experiment. The plot indicates that the lens step number and the best focused distance have an almost linear relationship in several regions (step numbers 20 to 65, 65 to 105, and 105 to 125). The sequence of four images used to construct the focused image and depth map are taken at lens step numbers 108, 115, 122, and 129. They are roughly in the linear region.

Rotational stereo analysis is then carried out using the focused images and the initial depth map estimated by SFF to get an accurate depth map. A sum-of-squared-difference measure[29] on 16×16 image blocks is used for matching in the foreground regions. Fast stereo matching is done by limiting the correspondence search to a small segment on the epipolar line, determined by the rough depth map obtained from SFF. The length of the segment for searching is computed using the maximum expected error of SFF. Finally, the 3-D shape is obtained by an inverse perspective projection of the resulting depth map. This data acquisition procedure is repeated for four viewpoints by rotating the object every 90 deg. The reconstructed focused image and partial 3-D shapes with different resolution settings are shown in Fig. 7.

3.5 *Accuracy Analysis and Quantization Error*

We analyze the accuracy of 3-D shape based on the disparity of stereo matching. The depth difference of one pixel disparity near the rotation axis is used to calculate the accuracy. To simplify the calculation, we assume that the rotation axis is parallel to the image plane and perpendicular to the image scan line. It is well known that the stereo disparity $d=bf/z$, where f is the camera focal length, b is the equivalent stereo baseline, and z is the depth. The disparity difference at two depths z_1 and z_2 is given by

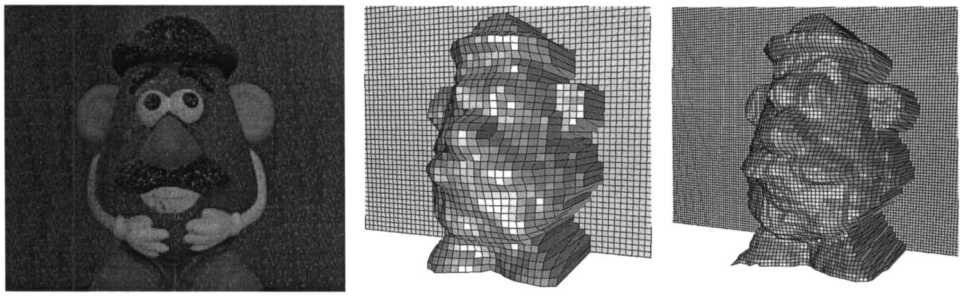

Fig. 7 Focused image and different resolutions of a 3-D shape (left: focused image; middle: 80 $\times$60; right: 160$\times$120).

$$\Delta d = bf\left(\frac{1}{z_1} - \frac{1}{z_2}\right), \tag{5}$$

or

$$z_2 = \frac{bfz_1}{bf - z_1\Delta d}. \tag{6}$$

Thus, the depth difference Δz can be written as

$$\Delta z = z_2 - z_1 = \frac{z_1^2\Delta d}{bf - z_1\Delta d}. \tag{7}$$

The equivalent stereo baseline in this setup is given by $b = 2z_0 \sin\beta/2$, where z_0 is the distance to the rotation axis and β is the rotation angle.

In the experiments, the pixel size is 0.00552 mm, the rotation angle is set to 6 deg, the camera focal length is 19.35 mm, and the distance to rotation axis is 832 mm. The maximum depth error for one pixel disparity is calculated as 1.41 mm at 832 mm from the camera. The rms error for a planar surface with a random dot pattern placed at 800 mm from the camera was found to be 1.21 mm (0.15%).

In addition to the 80$\times$60 depth map, higher resolution 3-D shapes corresponding to 160$\times$120 and 320$\times$240 were obtained with matching block sizes of 8$\times$8 and 4 $\times$4 pixels, respectively. Because of the quantization error introduced by pixel size, there exist *flat regions* of same disparity. This quantization error can be mitigated by increasing the stereo baseline, but it is not a perfect solution to a rotational stereo system. In our implementation, a linear interpolation using the boundaries of a region with the same disparities is applied. It gives a good approximation of the depth map. A median filter is then applied to smooth the resulting 3-D shape. Figure 8 shows the reduction of quantization errors. The left figure is the original 3-D shape of resolution 320$\times$240, and the right figure is the same 3-D shape after reducing the quantization error.

4 Registration

To create a complete description of an object, we need to acquire and register multiple range images to a common coordinate frame. Since our data acquisition system acquires range images from different viewpoints by rotating the object in front of the camera, the registration problem is equivalent to finding the rotation axis of the rotation stage. The registration consists of two steps. First, we find the initial estimated transformation by calibrating the rotation axis described in the previous section. Second, the estimated registration is further improved during surface integration. The overlapping parts of range images are used to refine the rotation axis and increase the accuracy of registration.

Registering range images in an iterative way is popular because of its ability to update information after each iteration. Most of the iterative registration methods try to find the translation and rotation matrices by either matching the closet points from different views[10,30] or minimizing the distance from points in one view to planes in another view.[1,31] We propose a new approach, which directly computes the rotation axis after each iteration without data fitting or minimization. Assuming the rigid body motion of the object is known (i.e., the rotation angle of the stage is accurate in our case), selected control points before and after current transformations are used to find the next transformations. Those control points are chosen from the overlapping part of two range images with certain criteria. Since the unit vector of the rotation axis is found to be (4.123×10^{-3}, 0.9997, -2.364×10^{-2}), which is very close to (0,1,0), we consider only the 2-D case (finding the translation vector).

The 2-D registration problem is to find the rotation center of the rotation transformation. The goal is to partly match the curves on the same plane from two different views. Consider the registration model shown in Fig. 9. Two curves are observed from different viewpoints with a 90-deg rotation angle (left and middle figures). The curve from the second viewpoint is transformed back to the world coordinate system according to the initial or estimated ro-

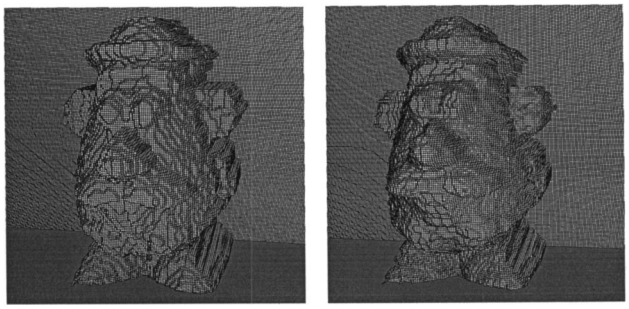

Fig. 8 Reduction of quantization error.

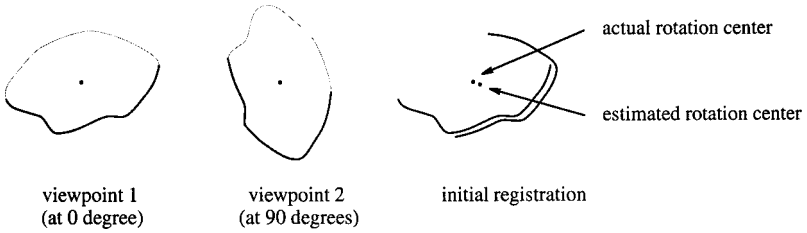

Fig. 9 Registration of two curves from different viewpoints with 90-deg interval.

tation center, as shown in the right figure. Then the overlapping part of these two curves is used to update the rotation center for the next rotation transformation. This process is repeated until the rotation center converges.

At each iteration, the next rotation center is found in three steps: 1. finding the overlapping part of two curves, 2. finding the *matching point* on the second curve, and 3. using the matching point and the corresponding point on the first curve to compute the rotation center. To find the overlapping part of the registered curves, we first convert the data points to the polar coordinate system centered at the estimated rotation center (see Fig. 10). The angles (θ values) are used to detect the overlap of these two curves. Let (r_1,θ_1) be the last point on the first curve and (r_2,θ_2) be the first point on the second curve. Then the overlap can be specified by all the points (r,θ) with $\theta_2 \leqslant \theta \leqslant \theta_1$. Since the points from the second view will be used to update the rotation center, we only consider the overlapping part on the second curve. Here we assume that there exists a one-to-one mapping between r and θ on all data points used for registration. The rotation axis calibration generally gives us the required accuracy on the rotation center to find the overlapping region.

The second step is to choose proper points for computing the new rotation center (Fig. 11). Since we assume that the rotation angle is fixed (e.g., $\pi/2$ in this case), the rotation center can be found if the points are known before and after rotation transformation. Let the overlapping data points on the second curve be $P_1,P_2,\ldots,P_n$, the current rotation center be denoted as C, and $Q_1,Q_2,\ldots,Q_n$ be the projection of $P_1,P_2,\ldots,P_n$ on the first curve along the lines $\overline{CP_1},\overline{CP_2},\ldots,\overline{CP_n}$. For this discretely sampled data, the projected point is given by the intersection of the projection line and the line segment of the data points on the first view. The *matching point* is defined as the projected point Q_i, which satisfies

$$\overline{P_iQ_i} = \max_{1\leqslant j\leqslant n} \overline{P_iQ_j}. \tag{8}$$

That is, the matching point is a point that satisfies the following two conditions: 1. it is one of the intersections of the projection lines and the first curve, and 2. its distance to the second curve is the maximum among those intersections. For a given data point on the first curve, the line segment (connected by two consecutive data points on the second curve) that is to be intersected is found by considering all of the overlapping line segments on the second curve. That line segment is then used to find the distance in Eq. (8).

The matching point Q on the second curve and the corresponding data point P on the first curve before transformation are used to compute the new rotation center. Let (x_1,y_1) be the point before transformation and (x_2,y_2) be the matching point. Then the rotation center (x_0,y_0) is given by

$$x_0=(x_1-y_2+y_1+x_2)/2,$$

$$y_0=(y_1+y_2+x_2-x_1)/2,$$

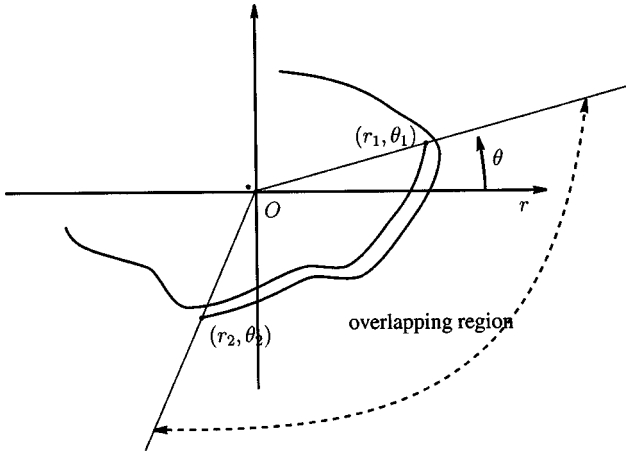

Fig. 10 Overlapping region of two registered curves.

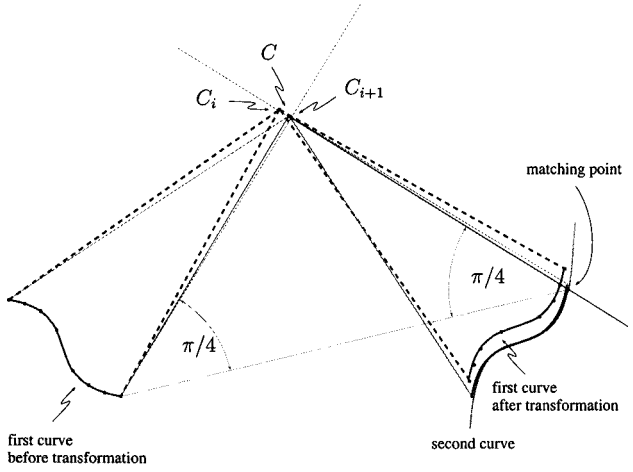

Fig. 11 Calculation of new rotation center.

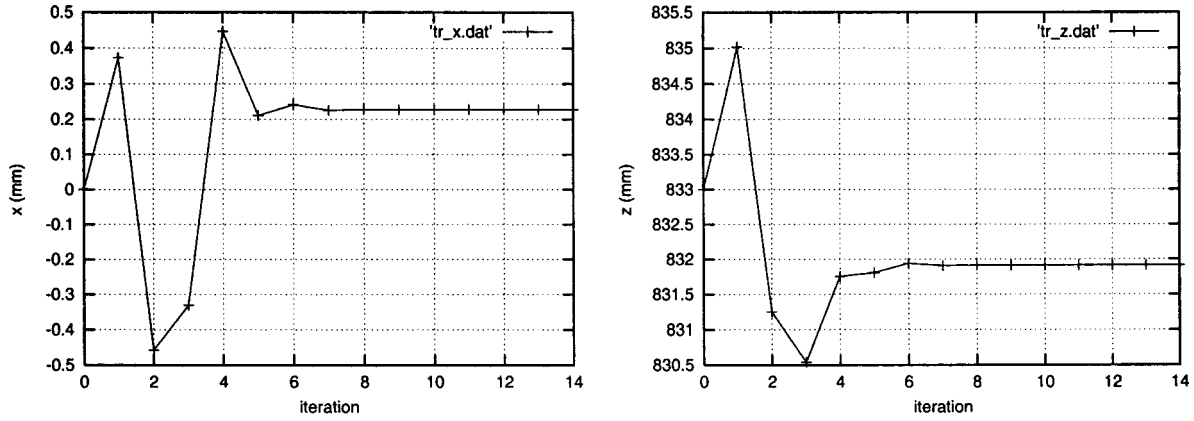

Fig. 12 Convergence of the translation vector.

or

$$x_0=(x_1+y_2-y_1+x_2)/2,$$

$$y_0=(y_1+y_2-x_2+x_1)/2,$$

for a fixed rotation angle of $\pi/2$. There are two solutions located at either side of $\overline{PQ}$. The ambiguity is solved by choosing the one on the same side as the previous rotation center.

In the experiments, cross sections of an object are used to find the rotation center. For any two range images from consecutive viewpoints, the rotation center is updated by taking the average of the rotation centers computed from the cross sections. The rotation center provided by rotation axis calibration is (0.0,833.0). After registration, the rotation center is moved to (0.23,831.91). Figure 12 shows the actual values of the translation vector converging with iterations. The experiments show that convergence of the translation vector is very fast (usually less than 20 iterations).

5 Surface Integration

Given a set of registered views, an integration algorithm should combine the partly overlapping datasets into a complete nonredundant 3-D dataset without any loss of detail in the original raw data. In the existing literature, Hoppe et al.[7] and Amenta, Bern, and Kamvysselis[6] construct a 3-D surface from unorganized points. Hilton et al.[13] and Soucy and Laurendeau[18] use structured data to combine multiple range images. All of them assume that accurate range data is available and they are perfectly registered. It is difficult to obtain such data in a practical data acquisition system. Therefore, an integration algorithm that is robust under the presence of noise and registration error is an important goal in designing a practical system.

5.1 *Region-of-construction Algorithm*

Our input depth-map data are given on a regular grid of points for each partial 3-D model. Therefore we develop a specialized triangulation method depending on the viewpoints. The basic idea is to *stitch* the *regions of construction* of different viewpoints at their boundaries to create a complete 3-D model. This algorithm takes advantage of the known topology of each range image and does not involve any spatial search of the data points. The mesh triangulation is done using indices of data points on a grid network. As a result, it is much faster than the general algorithms mentioned earlier.

For two range images from consecutive viewpoints, the overlapping part is divided into two regions and each region is assigned to one of the range images for construction. We first create the *region of construction* for each range image, and then use them to create a nonoverlapping dataset for surface integration. Region of construction is defined in Fig. 13. In the figure, the range image corresponding to the shaded region is the region of construction for view 0. The left (right) *line of sight* is defined as the line determined by the viewpoint and the leftmost (rightmost) data point. The angular bisector of right (left) line of sight of $i-1$ $(i+1)$ range image and left (right) line of sight of i range image is defined as the left (right) *line of division* for range image i. The 2-D region of construction for each cross section is then bounded by the left and right line of division. For each viewpoint, the region of construction includes all the 2-D cross sections. For each region of construction, the triangular mesh is created as follows.

We start at the upper left corner of the range image, find the points belonging to the region of construction, mark them as *valid points*, and establish the connection. For each valid point there exist five possible tessellations for triangles or quadrilaterals (see Fig. 14). For the first and second cases, the column index of the first valid point in the i'th row is smaller or larger than the column index of the first valid point in the j'th row, where $|i-j|=1$. Without loss of generality, assume row i contains the smaller column index. Let p be the smaller and q be the larger column index, then we make triangles by connecting indices $(j,q)-(i,p)-(i,p+1)$, $(j,q)-(i,p+1)-(i,p+2),\ldots,$ until $p=q-1$. For the third and fourth cases, the column index of the last valid point in the i'th row is smaller or larger than the column index of the last valid point in the j'th row, where $|i-j|=1$. Without loss of generality, assume row i contains the smaller column index. Let p be the smaller and q be the larger column index, then one or more triangles are created by connecting indices $(i,p)-(j,p)-(j,p+1)$, $(i,p)-(j,p+1)-(j,p+2),\ldots,$ $(i,p)-(j,q$

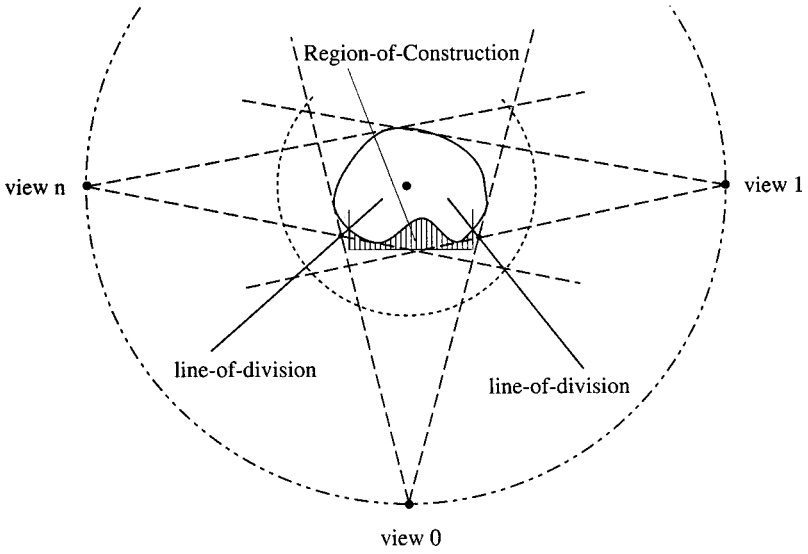

Fig. 13 Region of construction on a cross section.

$-1)-(j,q)$. In these four cases, one or more triangles are created depending on the column index difference between any two consecutive rows. For the last case, the column index of the valid point in the i'th row is the same as the column index of the valid point in the $(i+1)$ row. In this case, lots of quadrilaterals in the central part of the range image are produced, and the number of quadrilaterals depends on the difference between the last and first column index. Those quadrilaterals are further divided into triangles by connecting the shorter diagonals.

After the meshes for regions of construction are created for each view, the partial 3-D models are *stitched* together by connecting the last valid point of the current range image to the first valid point of the next range image in the same row (see Fig. 15). The resulting quadrilaterals on the boundaries of two range images are also broken into triangles with shorter diagonals.

The results of surface integration are shown in Fig. 16. The wireframe models of a toy, a head, a cylinder, and a bottle are presented. The toy object (upper left) contains approximately 9000 vertices and 18,000 polygons, and the head object (upper right) contains more than 27,000 vertices and 55,000 polygons. Different colors in the cylinder object indicate different regions of construction from different acquisition viewpoints.

5.2 *Complex Object with Holes*

Since the region of construction algorithm provides only nonoverlapping surfaces, complex objects with holes can be reconstructed by selecting proper viewpoints that include the holes. We assume that if a hole can be observed from one viewpoint, it can also be observed from the *opposite* viewpoint. Generally, this holds for most real objects. Under this assumption, a 3-D model is first constructed using the previous algorithm without considering its hole. Then the boundary points of the hole that belong to two opposite viewpoints are connected using the following algorithm.

First, we connect the top and bottom rows of two range images, respectively. The row index can be different. The column indices of the top (bottom) rows are used to create a triangle/quadrilateral mesh similar to the previous section. For a side boundary of the hole, another triangle/quadrilateral mesh is created with variable row index and fixed column index (one belongs to object points and next to a background point). Figure 17 illustrates this *hole-creating* process. One of the regions of construction is shown in wireframe. The quadrilaterals are further broken into triangles with short diagonals.

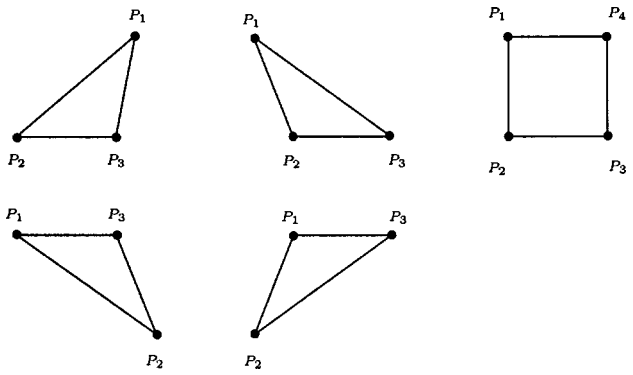

Fig. 14 Possible tessellations within a region of construction.

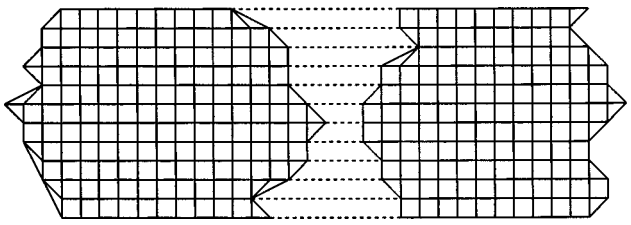

Fig. 15 Stitching between region of construction.

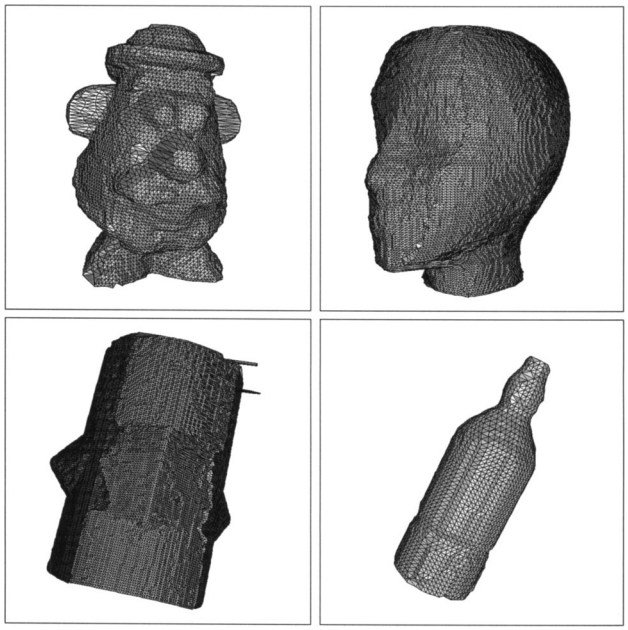

Fig. 16 Wireframe models of the test objects.

In some cases the observed hole does not appear in any region of construction defined in the previous section, but it gives depth discontinuities in one of the range images (see Fig. 18). We define the *principal view* as the viewpoint containing the hole. The lines of division of principal views are shifted such that the new regions of construction can completely cover the hole.

5.3 *Multiple Object Scenes*

To create the 3-D model of multiple objects in a scene, we assume that the objects are separable in the vertical direction. The range images are segmented for each object in the scene to create range image datasets for each object. The 3-D models for each object are first constructed using the integration algorithm described earlier and are then combined together. For each range image, the segmentation masks for each object are obtained by considering the rotation geometry of the scene. A depth threshold is used to separate different objects in the scene.

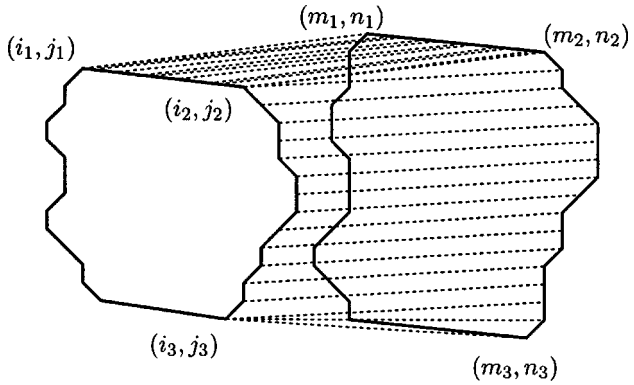

Fig. 17 Mesh for creating a hole. The index differences between (m,n) and (i,j) are used to generate quadrilaterals or triangles.

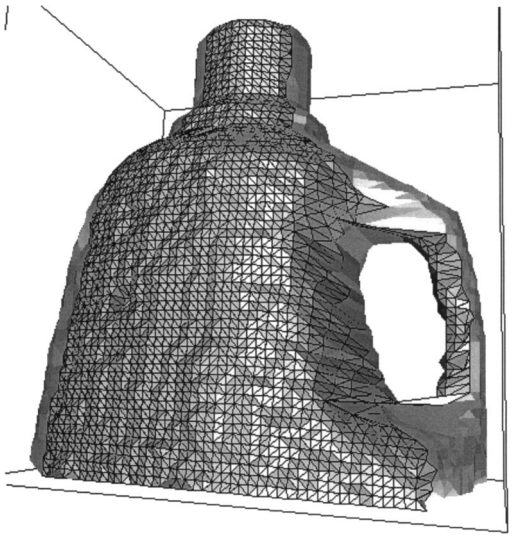

Fig. 18 Complex object with a hole.

Due to perspective projection of the camera, the constructed 3-D model looks distorted if the depth difference of the scene is too large. It shows that there are some missing parts on the 3-D models. However, the data points on the constructed model are exactly the same as the ones from range images. The result of wireframe models that contain multiple objects is shown in Fig. 19.

5.4 *Limitation*

Unlike computationally intensive algorithms such as α shapes or volumetric methods, the region of construction algorithm is not designed to reconstruct objects with arbitrary shapes. The shapes of objects should result in continuous regions and connected range image surfaces for stitching. Here, a continuous region means that there are no self-

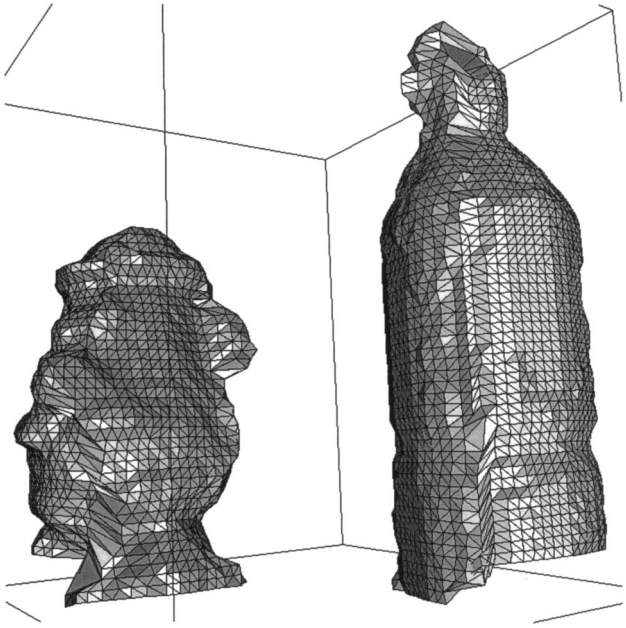

Fig. 19 Wireframe models of multiple object scenes.

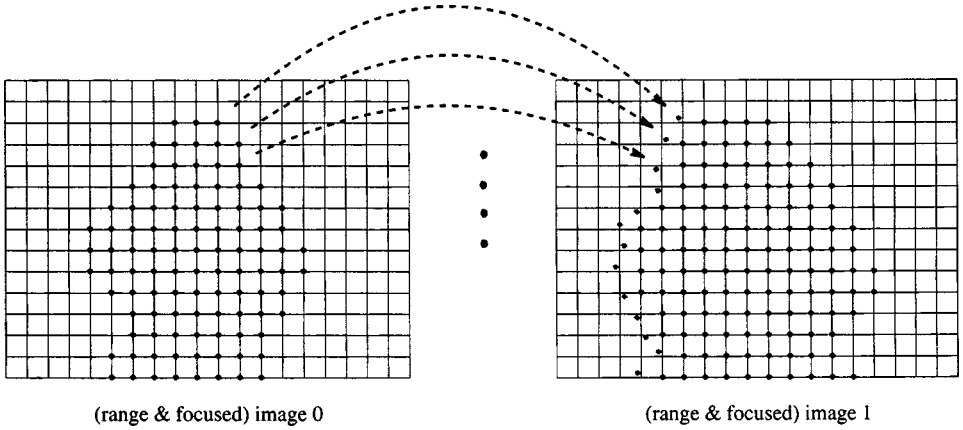

Fig. 20 Texture mapping on the boundary strip.

occlusions or holes. For most real objects, these situations can be possibly avoided during the data acquisition stage by some manual adjustments and handled by the hole-creating process. It should be noted that the object does not have to be convex. Concave regions of range images can also be stitched on the boundaries (see Fig. 13).

In this algorithm we do not use (or average) the overlapping part of the range images, and therefore it may not appear very smooth at the stitching boundaries. The smoothness mainly depends on the registration error and the assumption of a pinhole camera model. One can improve this by providing more accurate registration and using weighted averages of overlapping range data at the boundaries. To reconstruct an object with top and bottom surfaces, defining suitable regions of construction is not an easy task. The adjacencies of range images are not limited to two (it could be four in our implementation), and more sophisticated partitioning methods have to be considered.

6 Texture Mapping

One major advantage of using images to acquire a 3-D model is that the recorded images are used not only for measuring the 3-D shape of objects but also for providing the texture information. The color images used for texture mapping are obtained from shape from focus with four different focus positions. The texture image is focused everywhere and provides more realistic 3-D models.

Having modeled the 3-D shape of an object by a set of vertices and polygons, the image texture of the object is specified by providing a color image for each region of construction. Since the complete 3-D model is created by stitching all regions of construction, the textured 3-D model can be obtained by combining all textured regions of construction of the object. That is, nonoverlapped and texture mapped partial 3-D models are created first, and then stitched together. The texture map on the boundary strip connecting two consecutive regions of construction has to be obtained separately. This is because the boundary strips are not covered by any regions of construction. To obtain the texture map, each boundary strip is associated with (assigned to) one focused image. The vertices of the boundary strip on the adjacent region of construction (range image) are projected onto the focused image to calculate surface UV. The surface UV is then used to extract the texture map on this focused image.[32] As illustrated in Fig. 20, the 3-D points on the right boundary of range image 0 are projected onto range image 1 and the corresponding focused image according to the viewing geometry. The surface UV is calculated on each projected point and used to extract the texture map for the boundary strip between regions of construction 0 and 1 from focused image 1.

Figure 21 shows the focused images constructed by SFF and used for texture mapping of the toy object. To reduce the size of the textured 3-D model, a bounding box enclosing the region of construction is used to extract a subimage for each color image.

The shading information of an object is provided by the normal vector at each data point. The vertex normal is obtained by averaging the surface normals of all polygons sharing that vertex. Instead of using data points of the complete 3-D model, we calculate the surface normals from the range images to keep the original shading information. The resulting vertex normals on the boundaries of two consecutive partial shapes are different from the ones obtained from the complete 3-D model and can be used to check the smoothness of surface integration.

7 Experimental Results

The described algorithms were tested on a number of real objects. The results of a toy, head, cylinder, bottle, and multiple objects scene are presented. Several results of complete 3-D models with texture and shading information are shown in Fig. 22. For each object, it takes approximately 15 min to acquire stereo image pairs with four dif-

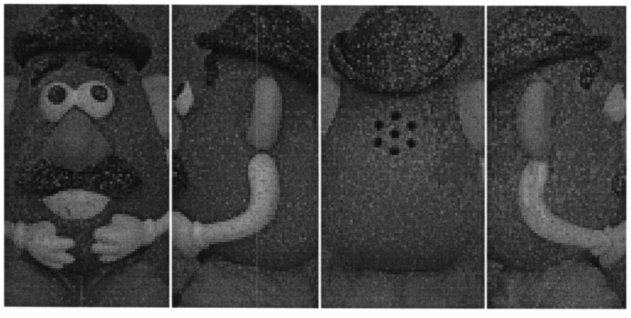

Fig. 21 Extracted subimages using bounding boxes.

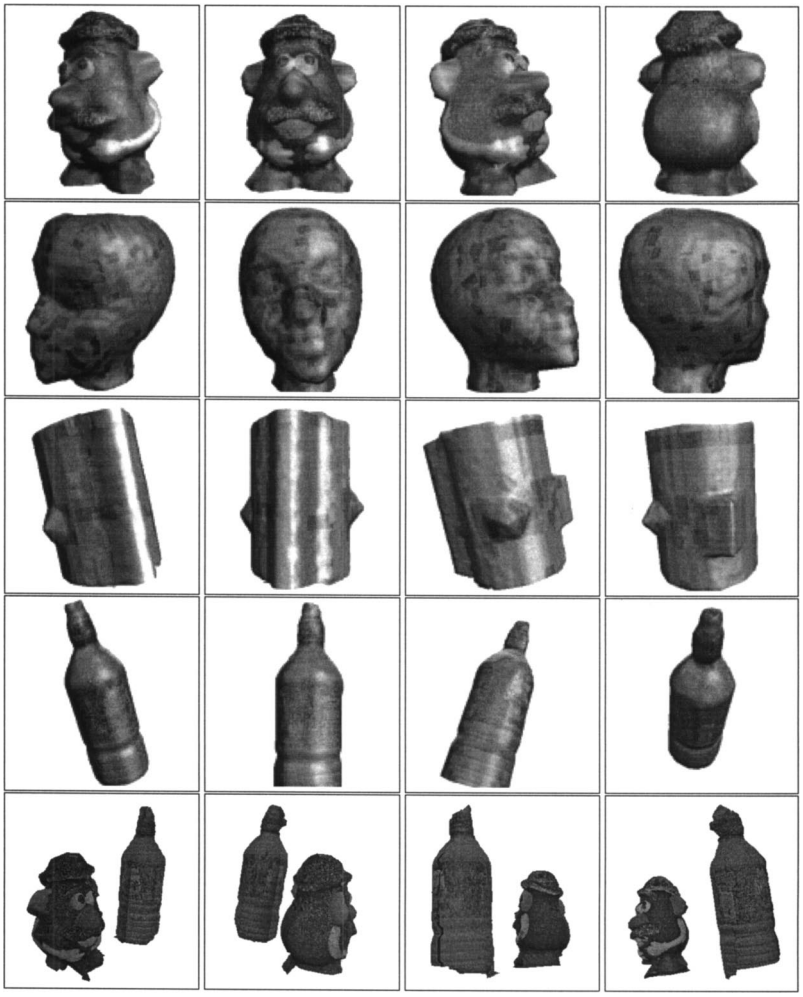

Fig. 22 Several viewing directions of complete 3-D models with texture mapping.

ferent focus positions for four views. It includes the capture of 32 still images and 360-deg movement of the rotation motor. The execution time for creating complete 3-D models—including image acquisition (IA), shape from focus (SFF), stereo matching (SM), surface integration (SI) and texture mapping (TM) are shown in Table 1 for several objects (on a Pentium II 450 MHz PC).

The accuracy of our 3-D model reconstruction depends on the accuracy of the depth map from rotational stereo and rotation axis of the rotation stage. A rudimentary accuracy analysis of our system was done using a cylinder as a test object. The same procedure and algorithms are applied to obtain the complete 3-D model. The acquired 3-D dataset is then fitted to the *perfect* cylinder from our physical measurement to calculate the average error of each data point acquired by our system.

Let d be the measured radius of the cylinder, $p_1, p_2, \ldots, p_n$ be the sampled data points, and $d_1, d_2, \ldots, d_n$ be the distances from the data points to the line that passes through the center of the cylinder. The average error for the 3-D dataset is then given by $1/n \Sigma |d_i - d|^2$, where $d_i = d(p_i, L_{eq})$. Since the line equation L_{eq} is unknown in the camera coordinate system, it is approximated by minimizing the sum of distances from each data point to the line, i.e., $\min \Sigma d(p_i, L_{eq})$. In other words, we are finding the best fitting cylinder for the acquired dataset. In our experiment, the measured radius is 77.5 mm and the

Table 1 Execution times for several objects.

Object	IA (32 images)	SFF	SM	SI	TM	Total
Toy	12 min. 48 sec.	5.5 sec.	74.2 sec.	1.3 sec.	4.9 sec.	14 min. 16 sec.
Head	13 min. 4 sec.	5.7 sec.	115.2 sec.	1.9 sec.	6.7 sec.	15 min. 14 sec.
Detergent	12 min. 43 sec.	3.0 sec.	37.7 sec.	0.5 sec.	5.4 sec.	13 min. 30 sec.

average error for each data point is 0.44 mm. Thus, the precision is about 1 part in 500 for our working volume of $250\times250\times250$-mm cube.

8 Conclusion

We design and implement a digital vision system for 3-D model reconstruction. The system is comprehensive in that it includes all stages—data acquisition, registration, surface integration, and texture mapping—to create a textured 3-D model. The complete 3-D model is constructed by merging multiple range images, and mapping the texture information acquired by rotational stereo and SFF. A new surface integration algorithm based on region of construction is developed for fast 3-D model reconstruction. It is also capable of constructing complex objects with holes and modeling multiple object scenes. Experimental results are presented for several real objects. Our system can be implemented using low-cost equipment: one commercially available digital camera, one rotation stage, and one personal computer. It is able to construct a complete 3-D model in under 20 min. Most of the acquisition time (about 15 min) is used to capture and transfer image data to a PC. A rudimentary analysis shows that the precision of our system is about 1 part in 500 in the working range. Future research will focus on extending the region of construction algorithm from the viewpoint-based regions to the regions with best viewing directions.

Acknowledgment

The support of this research, in part, by Olympus Optical Corporation is gratefully acknowledged.

References

1. Y. Chen and G. Medioni, ''Object modeling by registration of multiple range images,'' *Image Vis. Comput.* **10**(3), 145–155 (Mar. 1992).
2. K. Pulli, T. Duchamp, H. Hoppe, J. McDonald, L. Shapiro, and W. Stuetzle, ''Robust meshes from multiple range maps,'' *Proc. Intl. Conf. Recent Adv. 3-D Digital Imag. Modeling*, pp. 205–211 (May 1997).
3. H. Y. Lin and M. Subbarao, ''Three-dimensional model acquisition using rotational stereo and image focus analysis,'' *Proc. SPIE* **4189**, 201–210 (Nov. 2000).
4. E. Krotkov, ''Focusing,'' *Int. J. Comput. Vis.* **1**, 223–237 (1987).
5. M. Subbarao and T. S. Choi, ''Accurate recovery of three-dimensional shape from image focus,'' *IEEE Trans. Pattern Anal. Mach. Intell.* **17**, 266–274 (Mar. 1995).
6. N. Amenta, M. Bern, and M. Kamvysselis, ''A new Voronoi-based surface reconstruction algorithm,'' *SIGGRAPH 98*, pp. 415–421 (1998).
7. H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, ''Surface reconstruction from unorganized points,'' *ACM Comput. Surv.* **26**, 71–78 (Jul. 1992).
8. M. Reed and P. K. Allen, ''3D modeling from range imagery,'' *Image Vis. Comput.* **17**(1), 99–111 (Feb. 1999).
9. E. Trucco and A. Verri, *Introductory Techniques for 3D Computer Vision*, Prentice Hall, Englewood Cliffs, NJ (1998).
10. P. J. Besl and N. McKay, ''A method of registration of 3D shape,'' *IEEE Trans. Pattern Anal. Mach. Intell.* **14**(2), 239–256 (Feb. 1992).
11. G. Roth and E. Wibowoo, ''A fast algorithm for making mesh models from multi-view range data,'' *Proc. DND/CSA Robotics Knowledge Based Syst. Workshop*, pp. 339–356 (1995).
12. B. Curless and M. Levoy, ''A volumetric method for building complex models from range images,'' *Proc. SIGGRAPH'96*, pp. 303–312 (Aug. 1996).
13. A. Hilton, J. Stoddart, J. Illingworth, and T. Windeatt, ''Reliable surface reconstruction from multiple range images,'' *Proc. European Conf. Computer Vis.'96*, pp. 117–126 (1996).
14. M. Wheeler, Y. Sato, and K. Ikeuchi, ''Consensus surfaces for modeling 3D objects from multiple range images,'' *Proc. IEEE Intl. Conf. Computer Vision*, pp. 917–924 (Jan. 1998).
15. W. E. Lorenen and H. E. Cline, ''Marching cubes: A high resolution 3D surface reconstruction algorithm,'' *Proc. SIGGRAPH'87*, pp. 163–169 (Jul. 1987).
16. G. Turk and M. Levoy, ''Zippered polygon meshes from range images,'' *Proc. SIGGRAPH'94*, 311–318 (July 1994).
17. M. Rutishauser, M. Stricker, and M. Trobina. ''Merging range images of arbitrarily shaped objects,'' *Proc. 1994 IEEE Computer Soc. Conf. Computer Vision Patt. Recog.*, pp. 573–580 (June 1994).
18. M. Soucy and D. Laurendeau, ''A general surface approach to the integration of a set of range views,'' *IEEE Trans. Pattern Anal. Mach. Intell.* **17**, 344–358 (Apr. 1995).
19. H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, ''Mesh optimization,'' *Proc. SIGGRAPH'93*, pp. 19–26 (Aug. 1993).
20. H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schwitzer, and W. Stuetzle, ''Piecewise smooth surface reconstruction,'' *Proc. SIGGRAPH'94*, pp. 295–302 (1994).
21. M. Soucy and D. Laurendeau, ''Multi-resolution surface modeling from multiple range views,'' *Proc. IEEE Conf. Computer Vision Patt. Recog.*, pp. 348–353 (June 1992).
22. K. Pulli, H. Abi-Rached, T. Duchamp, L. Shapiro, and W. Stuetzle, ''Acquisition and visualization of colored 3D objects,'' *Proc. Intl. Conf. Patt. Recog.*, pp. 11–15, Brisbane, Australia (Aug. 1998).
23. W. Niem, ''Automatic reconstruction of 3d objects using a mobile camera,'' *Image Vis. Comput.* **17**(2), 125–134 (Feb. 1999).
24. J. Albamont and A. Goshtasby, ''A range scanner with a virtual laser,'' *Image Vis. Comput.* **21**(3), 145–155 (Mar. 2003).
25. B. K. P. Horn, *Robot Vision*, McGraw-Hill Book Co., New York (1986).
26. M. Subbarao, T. S. Choi, and A. Nikzad, ''Focusing techniques,'' J. Opt. Eng., **32**(11), 2824–2836 (1993).
27. H. Y. Lin and M. Subbarao, ''Complete 3D model reconstruction from multiple views,'' *Proc. SPIE* **4567**, 29–39 (2001).
28. H. Y. Lin and M. Subbarao, ''A vision for fast 3D model reconstruction,'' *IEEE Computer Soc. Conf. Computer Vis. Patt. Recog.*, Kauai, Hawaii, Dec. 2001
29. M. Okutomi and T. Kanade, ''A multiple-baseline stereo,'' *IEEE Trans. Pattern Anal. Mach. Intell.* **15**(4), 353–363 (Apr. 1993).
30. Z. Zhang, ''Iterative point matching for registration of free-form curves and surfaces,'' Technical Report, INRIA, Sophia-Antipolis (1992).
31. C. Dorai, J. Weng, and A. K. Jain, ''Optimal registration of object views using range data,'' *IEEE Trans. Pattern Anal. Mach. Intell.* **19**(10), 1131–1138 (Oct. 1997).
32. P. S. Heckbert, ''Survey of texture mapping,'' *IEEE Comput. Graphics Appl.* **6**(11), 56–67 (Nov. 1986).

Huei-Yung Lin received his BS in applied mathematics from National Chiao Tung University, Taiwan, and MS and PhD in electrical engineering from State University of New York at Stony Brook. In 2002 he joined the Department of Electrical Engineering, National Chung Cheng University, Taiwan, as an assistant professor. His current research interests include computer vision, digital image processing, and computer graphics. He is a member of the IEEE and SPIE.

Murali Subbarao obtained his BTech in electrical engineering from the Indian Institute of Technology, Madras, and MS and PhD in computer science from the University of Maryland, College Park. He joined the faculty of the Department of Electrical and Computer Engineering, SUNY at Stony Brook, soon after his PhD. He is the Founder and Director of the Computer Vision Laboratory in the department. He is the author of one book, four patents, and has published more than 50 papers. His research and teaching areas include computer vision, digital image processing, software engineering, digital systems design, and web and Internet technology.