High-Quality Texture Reconstruction from Multiple Scans

Fausto Bernardini, Ioana M. Martin, *Member*, *IEEE Computer Society*, and Holly Rushmeier

Abstract—The creation of three-dimensional digital content by scanning real objects has become common practice in graphics applications for which visual quality is paramount, such as animation, e-commerce, and virtual museums. While a lot of attention has been devoted recently to the problem of accurately capturing the geometry of scanned objects, the acquisition of high-quality textures is equally important, but not as widely studied. In this paper, we focus on methods to construct accurate digital models of scanned objects by integrating high-quality texture and normal maps with geometric data. These methods are designed for use with inexpensive, electronic camera-based systems in which low-resolution range images and high-resolution intensity images are acquired. The resulting models are well-suited for interactive rendering on the latest-generation graphics hardware with support for bump mapping. Our contributions include new techniques for processing range, reflectance, and surface normal data, for image-based registration of scans, and for reconstructing high-quality textures for the output digital object.

Index Terms—Computer vision, image processing, range images, reflectance and shading models, 3D scanning, texture acquisition.

1 Introduction

Three-dimensional scanners are used increasingly to capture digital models of objects for animation, virtual reality, and e-commerce applications for which the central concerns are efficient representation for interactivity and high visual quality. Most high-end 3D scanners sample the surface of the target object at a very high resolution. Hence, models created from the scanned data are often over-tesselated and require significant simplification before they can be used for visualization or modeling. Texture data is often acquired together with the geometry; however, a typical system merely captures a collection of images containing the particular lighting conditions at the time of scanning. When these images are stitched together, discontinuity artifacts are usually visible. Moreover, it is rather difficult to simulate various lighting conditions realistically or to immerse the model in a new environment. The scanning system used to perform the experiments described in this paper is equipped with a high-resolution digital color camera that acquires intensity images under controlled lighting conditions. Detailed normal and albedo maps of the surface are computed based on these images. By comparison, geometry is captured at lower resolution, typically enough to resolve only the major shape features. The benefits of such a system are twofold. First, it allows for the use of relatively inexpensive hardware by eliminating the need for dense geometric sampling and by taking advantage of digital color cameras that are quickly gaining in resolution while dropping in price. Second, the generated models are more readily usable in a visualization or modeling environment which exploits the hardware-assisted bump mapping feature increasingly available in commercial-grade 3D accelerators.

In general, the issue of acquiring and reconstructing high-quality texture maps has received less attention than the issue of capturing high-quality geometry. In this work, we build upon existing techniques developed for texture acquisition, reconstruction, and image registration to generate maps of high visual quality for the scanned objects. Particularly because the noise and inaccuracies of our scanner are greater than those of high-end systems, we wish to exploit in full all the geometric and image information acquired to improve the visual quality of the final representation.

In this paper, we introduce a novel texture reconstruction framework that uses illumination-invariant albedo and normal maps derived from the calibration-registered range and intensity images. The albedo maps are used in a unique way to refine a geometry-only registration of the individual range images. After the range data is integrated into a single mesh, the resulting object is partitioned into a set of patches. New textures are reconstructed by projecting the maps onto the patches and combining the best data available at each point using weights that reflect the level of confidence in the data. The weighted averaging lowers noise present in the images, while the fine registration reduces blurring, ghosting, and loss of fine texture details, as shown by the results in Fig. 1.

2 Related Work

A variety of techniques are used to capture digital models of physical objects, including CAT scans [1] and structure from motion applied to video sequences [2]. We restrict our discussion to methods involving instruments that capture range images (in which each pixel value represents depth) and intensity images (in which each pixel is proportional to

• *The authors are with the IBM T.J. Watson Research Center, PO Box 704, Yorktown Heights, NY 10598.*
E-mail: {fausto, holly}@watson.ibm.com, ioana@us.ibm.com.

Manuscript received 2 May 2000; revised 19 Oct. 2000; accepted 5 Dec. 2000. For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number 112032.

Fig. 1. The visual quality of textures for the model of a statue obtained from multiple overlapping scans is enhanced with our image-based fine-registration and weighted averaging techniques. The top image shows the final texture reconstructed from 20 overlapping scans. The two small images illustrate a detail from a highly carved area of the hair before (left) and after (right) image-based registration. The fine chisel marks become clearly visible after the registration.

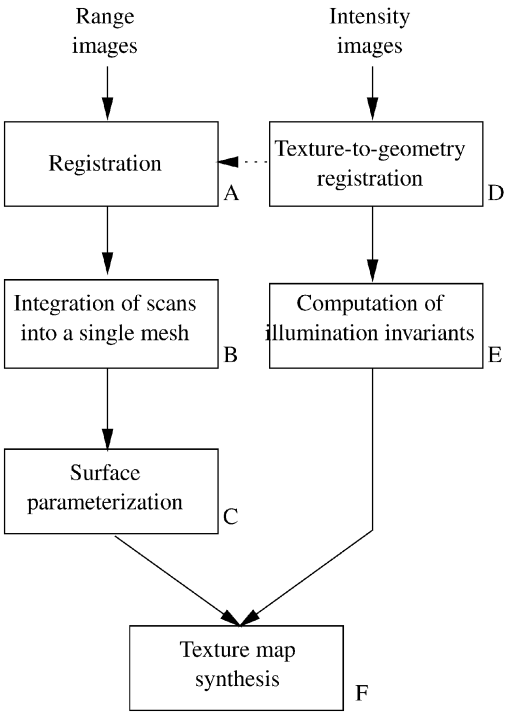


Fig. 2. The sequence of steps required for the reconstruction of a model from multiple overlapping scans.

the incident light). A detailed summary of such methods can be found in [3].

The basic operations necessary to create a digital model from a series of captured images are illustrated in Fig. 2. After range images are processed to remove outliers, they are in the form of individual height-field meshes (*scans*). Step A is to align these meshes into a single global coordinate system. In high-end systems, registration may be performed by accurate tracking. For instance, the scanner may be attached to a coordinate measurement machine that tracks its position and orientation with a high degree of accuracy. In less expensive systems, an initial registration is found by scanning on a turntable, manual alignment, or approximate feature matching [4], [5]. The alignment is then refined automatically using techniques such as the Iterative Closest Point (ICP) algorithm of Besl and McKay [6] or the technique of Chen and Medioni [7]. Recent multiview registration algorithms include [8], [9].

In Step B, the scans are integrated into a single mesh. The integration may be performed by zippering/stitching [10], [11], isosurface extraction from volumes [12], [13], or interpolating mesh algorithms [14] applied to points which have been corrected for line-of-sight errors (geometric errors along the line from the scanner to the acquired point) in the individual scans.

To relate a texture map to the integrated mesh, in Step C, the surface is parameterized with respect to a 2D coordinate system and texture coordinates are interpolated between mesh vertices. A simple parameterization is to treat each triangle separately [13], [15] and to pack all of the individual texture maps into a larger texture image. However, the use of mip-mapping in this case is limited since adjacent pixels in the texture may not correspond to adjacent points on the geometry. Another approach is to find patches of geometry which are height fields that can be parameterized by projecting the patch onto a plane. Stitching methods [1] use this approach by simply considering sections of the scanned height fields as patches. Marschner [16] subdivides the surface into individual patches by starting with seed triangles distributed over the object and growing regions around each seed. Using the technique described in [17], harmonic maps are found to establish a 2D coordinate system for each patch, so individual patches need not be height fields. Other methods could be built on tiling methods developed for multiresolution analysis [18] or interactive texture mapping [19].

Parallel to acquiring geometry, intensity images are captured to obtain information about the albedo of the surface. Such images may be recorded with electronic or traditional cameras or by using polychromatic laser technology [15]. In Step D, these images are aligned to the corresponding geometry. In some cases, the image acquisition is decoupled from the geometry acquisition [16], [20], [1]. The camera intrinsic and extrinsic parameters for the images are estimated by manual or automatic feature

matching. The advantage is that acquisition modalities that cannot capture surface reflectance can be used for capturing geometry.

In most cases, however, the alignment is performed by calibration. Geometry and intensity are captured simultaneously from scanners with a measured transformation between sensing devices. The resolution of the intensity image may be the same as that of the range image or even higher. When the resolution is the same, texture mapping is unnecessary since a color can be assigned to each vertex. Nevertheless, such a representation is inefficient and geometric simplification is typically performed before the surface parameterization step.

The main benefit of obtaining intensity and range images simultaneously is that the intensity information can be used in the registration process in Step A. Various approaches have been developed to use intensity images in registration. Johnson and Kang [13] use color as an additional coordinate in the ICP optimization. This avoids local minima in the solution in areas that have no geometric features, but have significant variations in the intensity. For models with pronounced geometric and intensity features, the method has proven to be very effective. A drawback is having to combine position and color data with different ranges and error characteristics. For subtle feature variations, these can cause one type of data to erroneously overwhelm the other. Schütz et al. [21] introduce empirically computed normalization factors to define a coupling distance that accounts for position, normal, and color differences.

A number of researchers have used information in intensity images to avoid the spatial search required by ICP. These methods use scans that have been aligned approximately, generally by manual means. Weik [22] uses intensity and intensity gradient images from the approximately aligned scans after they are transformed into a common camera view. Locations of corresponding points on overlapping scans are inferred based on the difference between intensity values at a given pixel and the gradient at that pixel. Pulli [23] performs a full image registration to find corresponding points. Similar to Weik's method, one intensity image is projected into the camera view of the second, using the approximate alignment. A planar perspective warping [24] of the first image is then computed to match the rendered image of the second scan. For each corresponding pixel of the two images, under the computed transformation, a pair of points from the two scans is generated. A least-squares optimization is then performed to compute a registration matrix. Both Weik's and Pulli's methods require operations on the full high-resolution intensity images. A high degree of overlap is required and scan-to-scan variability in illumination introduces error. Fine scale geometry is matched only if these details are revealed by lighting in the images. Both methods are effective if there are substantial albedo variations in the scans that dominate illumination variations. Gagnon et al. [25] avoid full image operations in an alternative non-ICP method for using intensity images to refine an initial manual alignment. Pairs of range images are aligned manually by marking three points on overlapping intensity images. The locations of the matching points are refined by searching their immediate neighborhoods using image cross-correlation [26]. A least-squares optimization follows to determine a general 3D transformation that minimizes the distances between the point pairs.

After intensity images are aligned to geometry, illumination invariant maps are computed to estimate the surface albedo (Step E). The number of scans versus the number of intensity images, as well as the resolution of the scans compared to the resolution of the images, are issues that have to be considered. For a small number of scans and a large number of intensity images obtained under calibrated lighting conditions, the full Bidirectional Reflectance Distribution Function (BRDF) can be estimated [27]. If many scans are required to represent an object and only a few high-resolution intensity images are captured per scan, photometric stereo techniques [28] can be used to estimate Lambertian reflectance [29]. Alternatively, if the range and intensity images have the same resolution, the geometry can be used to compute reflectance from a single image [30].

In Step F, the final texture is reconstructed. The illumination invariant images are mapped onto the integrated, parametrized surfaces. The main concerns at this step are that the final texture is as sharp as the best input images, that seams between scans or texture patches are not visible, and that all information available is fully exploited to maximize the signal-to-noise ratio. To maintain sharpness, Rocchini et al. [1] propose a stitching approach that uses a single illumination invariant map at any given surface point. Continuity in sharp features between adjoining maps is maintained by a local texture adjustment at texture boundaries. This requires high-quality input maps that have no visible noise and no scan-to-scan chromatic differences. Map adjustment techniques such as this, as well as deghosting methods for image mosaics [24], decouple texture from geometric variations. This may cause noticeable artifacts when these variations are correlated (e.g., dents and scratches that reveal underlying material with different reflectance properties). To avoid jumps in color appearance and to reduce noise, Johnson and Kang [13] combine information from multiple overlapping scans. Marschner [16] estimates Lambertian reflectance at each texture pixel in the reconstruction step, using every data value from captured images covering the point. Each data value is weighted by the relative angle to the camera view, light source, and specular direction in the captured image. In this case, if texture alignment is imperfect, blurring or ghosting artifacts may be generated. Pulli et al. [23], [31] do not explicitly form texture images associated with the geometry, but propose a dynamic, view-dependent texturing algorithm which determines a subset of the original images taken from view direction close to the current view and synthesizes new color images from the model geometry and input images.

3 Overview

In this section, we present our approach to each of the steps described in the previous section. Our goal is to produce a model of high visual quality, rather than to acquire a dense geometric representation. We propose two new techniques for the registration and reconstruction steps, respectively.

They are outlined in this overview and will be described in more detail in Sections 4 and 5.

3.1 Scan Acquisition and Processing

We use a scanning system in which range and intensity images are obtained simultaneously and are registered via a priori calibration. The scanner is a hybrid multiview photometric system built around the Visual Interface Virtuoso scanner. Range images are captured by a multicamera stereo system which projects a striped pattern onto the target object. The system produces range images that are converted to individual triangle meshes (scans) with an approximate intersample distance of 2 mm and submillimeter depth accuracy.

Intensity images are recorded under five calibrated lighting conditions and have a much higher resolution than the geometric scans (between 0.25 mm and 0.5 mm intersample distance, depending on surface orientation). The intensity images are processed to extract RGB albedo, normal, and weight (confidence) maps for each scan. This photometric processing involves using the approximate knowledge of the underlying geometry to account for imperfections in the measuring system, as well as global compensation of variations in camera response [29], [33].

We refer to the images that contain the RGB albedo and normals generically as *texture maps*. Holes may occur in these textures where photometric processing fails whenever fewer than three light sources are visible at a point. In such regions, the textures are filled using the normals of the underlying mesh and data from just one or two of the associated intensity images. A weight map encodes a degree of confidence in the data at each pixel of a texture map. The weight includes the effects of distance to the camera, angle to the camera view direction, whether a photometric normal was computed, and the distance to the edge of the scan. The boundaries between photometric values and underlying values are smoothed, as are the corresponding weights, so that abrupt changes in the final texture will not be visible.

The acquisition of intensity images aligned with the corresponding range images and the photometric processing just described correspond to Steps D and E in Fig. 2.

3.2 Registration

We begin the scan registration Step A with a pairwise manual alignment to estimate the position of each scan with respect to a global coordinate system. This step is performed at the same time with the removal of outliers in the range data. Next, the initial manual registration is refined using a variation of the multiview ICP algorithm [8]. Our ICP implementation uses geometric information only.

Like Johnson and Kang [13] and Schütz et al. [21], we seek to produce high-quality texture maps by combining information from multiple overlapping scans at each pixel. However, we avoid the mixing of position and color and the use of empirical weights by performing an image-based alignment *after* the geometry-based registration has converged. Similarly to Gagnon et al. [25], we use image matching to refine the alignment. However, our method is different in a number of critical ways. First, since this is a refinement step, we compare images that are reprojected onto the same camera view to account for geometric distortions. Second, instead of manually selecting three points to be matched, we have implemented an automatic selection procedure that identifies a larger number of samples in areas with significant image structure. Finally, we use images that are consistent with each other by processing them according to methods described in [33]. Our registration method has some similarities with the one proposed by Pulli [23]. Pulli also uses images to find matching points pairs, projects them back onto the range scans to find corresponding surface points, and then minimizes a functional of their pairwise distances. However, rather than relying on a full-image warping transformation to find matching points on overlapping scans, we do local searches on small regions based on image correlation. The local searches avoid the need for a hierarchical image registration method and bound the adjustments to a small search range since the initial alignment has already been improved by ICP. Also, rather than considering each pair of corresponding pixels as a valid match, we automatically select a subset of sample pixels in areas rich with detailed image features (derived from the albedo or normal maps) to minimize the occurrence of false matches.

The basic idea of our image-based registration algorithm is to use *detail maps*, by which we mean any image maps that can be computed by processing and/or combining the available texture maps. Detail maps are used to generate highly accurate pairs of corresponding points on the scans by matching their image neighborhoods. The scans are considered for alignment one at a time, in random order, while the others remain fixed, as suggested in [8]. Given a scan to be aligned, sample points are selected automatically on its detail map in regions where interesting features are present. The detail maps corresponding to all overlapping scans are then projected onto the current image plane and a search is performed in the neighborhood of each sample point for a best match in the overlapping areas of the projected maps. The resulting pairs of image point samples are backprojected onto their corresponding geometry and used to compute a rigid transformation that minimizes the sum of squared distances between corresponding points. The transformation thus obtained is applied to the moving scan and the process is repeated until a convergence criterion is satisfied.

3.3 Surface Reconstruction and Parameterization

Having obtained a tightly aligned set of scans, we proceed to integrate them into a seamless triangle mesh (Step B). We use a technique called Ball Pivoting [14], which efficiently generates a triangle mesh by interpolating an unorganized set of points.

Next, we determine a suitable surface parameterization (Step C) to allow for an efficient computation of texture maps that cover the entire object without overlapping. Since we wish to use mip-mapping techniques, we partition the surface into *patches* and assign one texture image to each patch, rather than parameterizing the mesh one triangle at a time.

Our texture reconstruction technique uses all the acquired data at each point as opposed to stitching together pieces of the initial scans. Hence, we do not use

the initial scans as texture patches. Instead, a new partition of the geometry into height-fields is computed by a greedy approach that begins with a number of seed triangles and grows the surface regions around them until a maximum patch size or maximum slope deviation is reached. Each patch is then projected in the direction which maximizes its total projected area, providing a simple local parameterization.

More sophisticated techniques for partitioning a mesh into patches suitable for texture computation have been proposed, see, for example, [16], [17], [18], [19]. Any method that provides a mapping from the texture image assigned to the patch onto the patch geometry could be used in our approach and it is reasonable to expect that any of the methods cited above would generate a smaller number of patches and a more even allocation of texture pixels per surface area than what we obtain with our simple greedy strategy. However, we found that, for the datas ets used in our experiments, our technique provided patches of sufficient quality.

3.4 Texture Reconstruction

Once the model is partitioned into height-field patches, albedo and normal maps are reconstructed for each patch by combining the information in all overlapping textures (Step F). Information from the various textures is combined using weights representing the confidence in each pixel value. Various proposals for weights for texture combination have been proposed by Pulli et al. [23] [31], Ofek et al. [33], Pighin et al. [34], and Neugebauer and Klein [20]. Weights are combined in the techniques described in these papers from factors such as the orientation of the surface element in the direction to the scanner, the distance to the edge of the scan, and the variation of pixel values for the same surface element appearing in multiple scans.

In our approach, we begin by identifying all scans which contain the albedo and normal data for the current output pixel being computed. The corresponding values are combined using a weighting scheme using geometric factors described in previous methods and a weight indicating whether the albedo and normal were computed from the photometric data or are a background, low resolution value. Using weights computed per scan avoids discontinuities at patch-to-patch and scan-to-scan transitions (see also Fig. 9). Occlusions must also be handled correctly. In practice, since the amount of data involved in this phase may be quite large, it is important to organize the required computations efficiently.

4 Image-Based Registration

The goal of image-based registration is to improve the geometry-based alignment of scans that make up a 3D object by taking into account additional information contained in the high-resolution detail maps computed for each scan.

Detail maps are generated from albedo and normals maps. Depending on the application, they may be single or multichannel images. In our experiments we use gray-scale albedo maps and geometry-invariant maps derived from the normals. The geometry-invariant, that is, coordinate system independent, maps are computed as the dot product of each surface normal with the average of the normals in a small surface region around it. The proposed image-based registration algorithm makes use of image matching to identify pairs of corresponding points on the high-resolution detail maps. Subsequently, these pairs are backprojected onto the scans and used to derive a rigid transformation that minimizes the distance between corresponding points in a least-squares sense [6].

Fig. 3 illustrates our method using a simple example consisting of three scans S_i, S_j, and S_m, with their respective detail maps, D_i, D_j, and D_m, and cameras C_i, C_j, and C_m. For the purposes of this example, S_i and S_j are fixed, while S_m is being moved to achieve better alignment with respect to S_i and S_j. The detail maps D_i and D_j are rendered onto the image plane of camera C_m, generating projected detail maps $\tilde{D}_i$ and $\tilde{D}_j$, respectively. Ideally, in the absence of noise and errors in the initial alignment, the images D_m, $\tilde{D}_i$, and $\tilde{D}_j$ would be identical in the areas of overlap. A sample point selected in such an area in one image would identify precisely the same feature in all overlapping images. In practice, corresponding features may appear shifted due to misalignment. Let t_m^1 be a sample point on D_m with pixel coordinates (u, v) that are inside the overlap region of D_m with $\tilde{D}_i$ and let t_i^1 be the point on $\tilde{D}_i$ with the same coordinates: $t_i^1 = (u, v)$. A correlation-based search is conducted in a neighborhood of t_i^1 to identify a point b_i^1 for which the image area around it best matches the area around t_m^1. The points t_m^1 and b_i^1 are backprojected onto $p_m^1 \in S_m$ and $p_i^1 \in S_i$, respectively, and the pair (p_m^1, p_i^1) is saved. The process is repeated for other sample points on D_m where there is overlap with either $\tilde{D}_i$ or $\tilde{D}_j$. Finally, after all samples are processed, the pairs $(p_m^k, p_{h=i,j}^k)$ are used to compute a rigid transformation that improves the position of scan S_m with respect to S_i and S_j in the least-squares sense [6].

Our image-based registration algorithm is described in pseudocode in Fig. 4. The input consists of the scans to be registered with their initial registration matrices, as well as their corresponding detail maps, depth maps, bounding boxes, and lists of sample points. In addition, the calibration parameters of the intensity-capture camera are considered known. These parameters include the position and orientation of the camera in the local frame of each scan, its field-of-view angle, and the pixel size of the output image. Knowledge of these parameters allows us to define a camera that matches the view of the capture camera. With the exception of the bounding boxes which are stored in memory for the entire duration of the alignment, all other data is loaded on demand. The output is a set of registration matrices (one per scan) that defines the new, more accurate alignment. The main steps of the algorithm are described next.

4.1 Selection of Sample Points

Prior to the actual alignment, a list of sample points L_m is computed for each scan S_m. We use image-processing techniques to identify points $t_m^k, k = 1, \ldots, n_m$ in the detail map D_m. The goal is to select points in areas of rich content

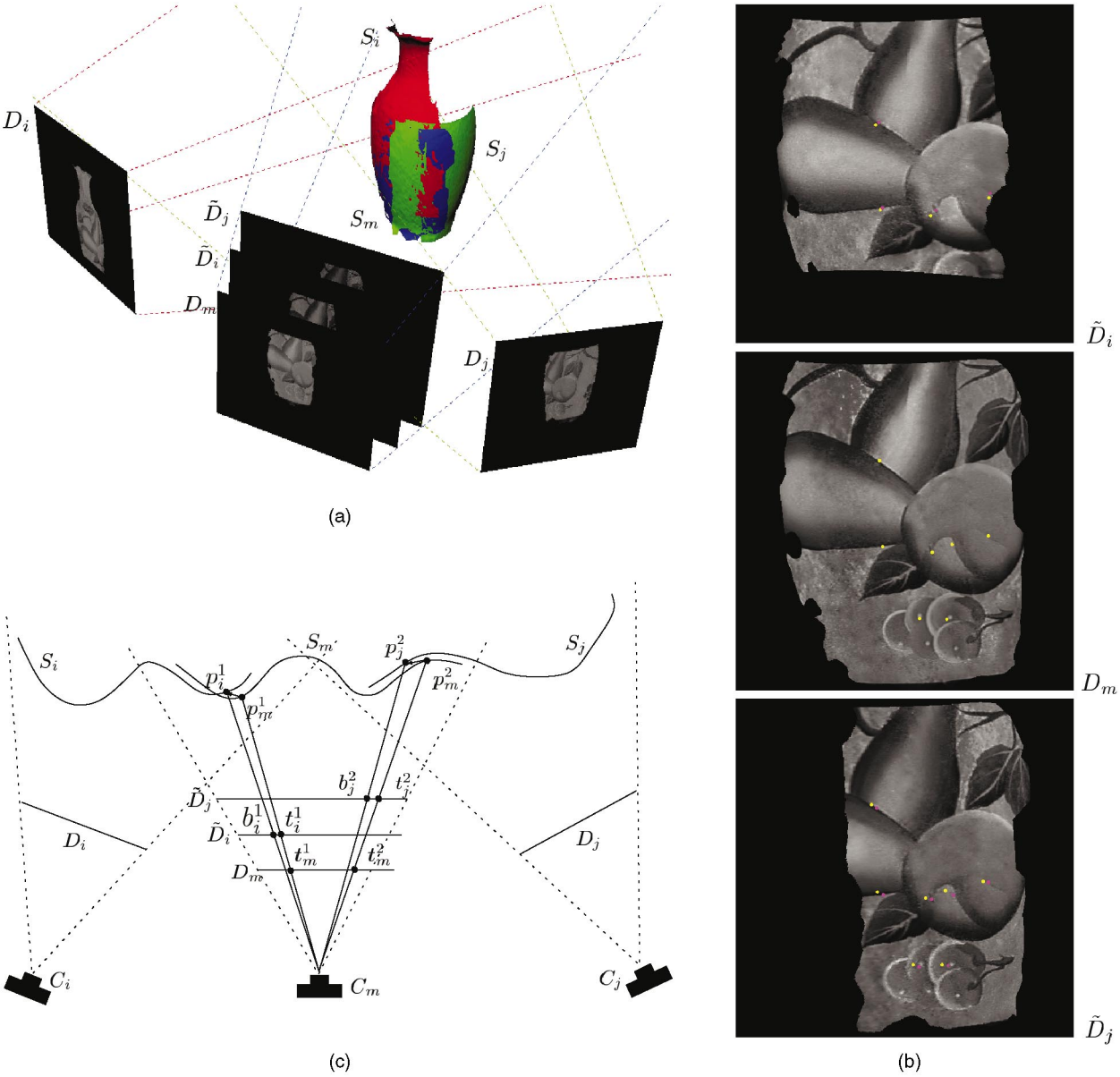


Fig. 3. (a), (b) Example and (c) two-dimensional diagram of the image-based registration algorithm. Three scans are involved: Scan S_m (blue), with corresponding detail map D_m, is being registered to the two overlapping scans S_i (red) and S_j (green). Detail maps D_i and D_j are mapped onto their respective geometry and rendered onto the image plane of camera C_m, generating reprojected images $\tilde{D}_i$ and $\tilde{D}_j$ (the three images D_m, $\tilde{D}_i$, and $\tilde{D}_j$ are shown cropped in (b). $\tilde{D}_i$ and $\tilde{D}_j$ are drawn with a little offset in (a) and (c) for illustrative purposes). In the ideal case of perfect scans and no registration error, these images would be, pixel-by-pixel, the same as D_m in the areas of overlap. Assuming some registration error, we consider a set of sample points on detail map D_m, distributed in areas where interesting image features are present. Starting with a sample point t_m^1 on D_m, corresponding to geometric point p_m^1 on S_m, we search in the neighborhood of the corresponding point t_i^1 on $\tilde{D}_i$ for a point b_i^1 that maximizes image correlation (initial and final points are shown with yellow and magenta dots, respectively, on the images in (b)). The point b_i^1 is then backprojected onto the scan S_i into p_i^1. The process is repeated to generate a set of pairs of corresponding points $(p_m^k, p_h^k), h = i, j$. A rigid transformation is computed that minimizes the sum of squared distances between corresponding points.

that pertain to matching by cross-correlation, rather than to identify precise image features. In addition, for each t_m^k, we also compute the corresponding point $p_m^k \in S_m$ under the perspective projection defined by the capture camera.

We have experimented with a variety of techniques, including selection based on moment invariants and several edge-detection methods. For the test cases described in this paper, we used an edge-detection technique which is illustrated in Fig. 5. Fig. 5a shows a portion of the original detail image. First, the detail image is slightly blurred, using a Gaussian filter to reduce the noise. Next, horizontal and vertical Sobel operators are applied separately to the blurred image. The two resulting edge images are combined into one that contains at each pixel the largest of the corresponding values at that pixel (see Fig. 5b). A thresholding operation is applied next to filter out regions with low gradient variation. The threshold value is selected automatically, based on the image histogram. The result is a bitmap that contains values of one where the gradient variation is above the threshold and zero elsewhere (see Fig. 5c). This bitmap is used to select sample points in regions where edges have been identified. Before selecting

Input: scans to be registered: $S_1, \ldots, S_N$
initial registration matrices: $M_1, \ldots, M_N$
detail maps: $D_1, \ldots, D_N$
depth maps: $Z_1, \ldots, Z_N$
bounding boxes: $B_1, \ldots, B_N$
lists of sample points: $L_1, \ldots, L_N$
where $L_m = \{(t_m, p_m) | t_m \in D_m, p_m \in S_m\}$
camera parameters: C

Output: accurate registration matrices: $M_1, \ldots, M_N$

Algorithm *Image-Based Registration*

1. read $B_1, \ldots, B_N, M_1, \ldots, M_N, C$
2. while (*convergence criterion not satisfied*)
3. $(m_1, \ldots, m_N) =$ *random_permutation* $(1, \ldots, N)$
4. for $(m = m_1, \ldots, m_N)$
5. $C_m =$ *camera* (M_m, C)
6. *register_scan* (m, C_m)

Procedure *register_scan* (m, C_m)

1. read D_m, Z_m, L_m
2. for $(i = 1, \ldots, N)$ and $(i \neq m)$
3. read S_i, D_i
4. if (B_m *intersects* B_i)
5. $\tilde{D}_i =$ *project_scan* (S_i, D_i, C_m)
6. $O_{mi} =$ *compute_image_overlap* $(D_m, \tilde{D}_i)$
7. if $(|O_{mi}| >$ *min_overlap*$)$
8. for $((t_m, p_m) \in L_m)$ and $(t_m \in O_{mi})$
9. $t_i = t_m$
10. $b_i =$ *find_best_match* $(t_m, D_m, t_i, \tilde{D}_i)$
11. $p_i =$ *back_project* $(b_i, S_i, C_m) \in S_i$
12. *insert_pair* $(p_m, p_i,$ *pairs_list*$)$
13. if $(|$*pairs_list*$| \geq 3)$
14. $M_m =$ *compute_rigid_transform* (*pairs list*)

Fig. 4. Skeleton of the image-based registration algorithm.

the points, all pixel values are set to zero where the corresponding weights in the weight map are below a given confidence threshold since we do not want any samples in these areas (see Fig. 5d). Thresholding weights is particularly critical in our system in which the weights include whether there was adequate data available for a particular scan to compute detailed results. We take care to eliminate areas with low weight values so that scans with inadequate data in a particular area are not mistaken for scans with no fine detail in that area. For the actual selection, a regular grid is superimposed onto the bitmap to ensure that selected points are distributed over the entire image. The first nonzero pixel encountered in each grid cell in row-wise order is selected as the sample point corresponding to that cell and is inserted into the list of samples. During alignment, points in the list are considered in random order to ensure that all meaningful regions of the image contribute to the registration process. We emphasize that we use the edge bitmap solely for the purpose of selecting samples. The actual image matching is performed by cross-correlation on the detail maps in the neighborhood of each sample point.

4.2 Projection of Detail Maps

For the alignment of a given scan S_m (procedure *register_scan* in Fig. 4), the pairs of points to be matched are selected

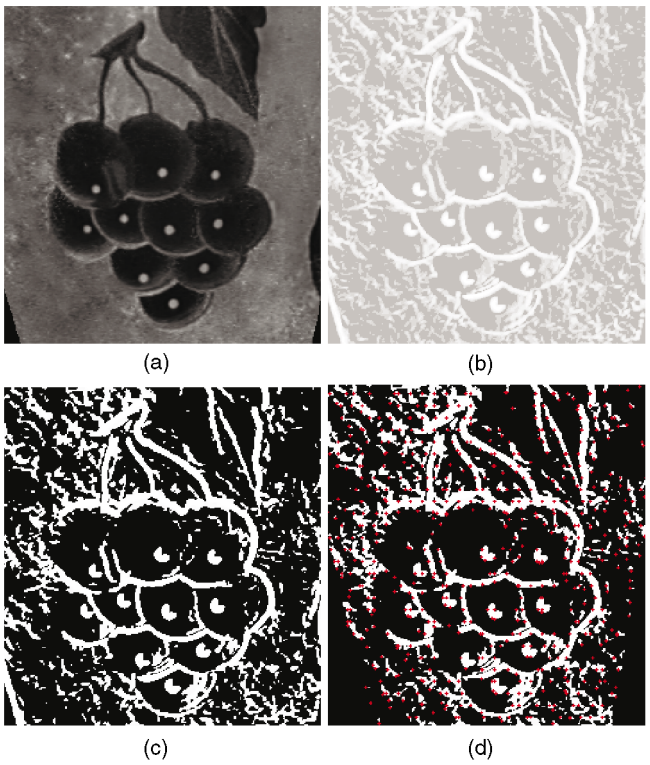


Fig. 5. Selection of sample points from detail maps using edge detection.

based on corresponding points in the detail map D_m and the detail maps of overlapping scans. To compare two detail maps D_m and D_i, they have to be rendered from the same camera position, i.e., D_m is compared to $\tilde{D}_i$, the projection of D_i onto the image plane of the camera C_m (see Fig. 3.)

A simple way to perform such projections is to render scan S_i with the detail image D_i as its texture, using camera C_m. However, this approach produces incorrect results in the presence of scan self-occlusions or occlusions from other scans, as shown in Fig. 6. To avoid this problem, a *depth map*, Z_m, is stored with each scan. This is simply a binary dump of the z-buffer created by rendering S_m using its own camera C_m. Before rendering S_i from camera position C_m, the z-buffer is preloaded with the depth map Z_m. A small offset, ϵ, is added to prevent z-buffer contention and to account for alignment errors between S_m and S_i. The occluded parts of S_i are now correctly discarded, and the corresponding regions in the projected detail map $\tilde{D}_i$ receive no texture. Fig. 6 shows two contrasting situations: Point p_i^1 is occluded by p_m^1 and, thus, the corresponding pixel $t_i^1 \in \tilde{D}_i$ remains black, whereas point p_i^2 is visible from C_m and, thus, t_i^2 receives texture values from D_i.

4.3 Identification of Matching Pairs

Given a pair of detail maps D_m and $\tilde{D}_i$, both rendered from the same camera position C_m, the algorithm first computes an overlap bitmap O_{mi}. A pixel in this bitmap is set to one if and only if both D_m and $\tilde{D}_i$ have a weight at that pixel that is larger than the confidence threshold. If there is sufficient overlap, the sample points $(t_m, p_m) \in L_m$ for which t_m is inside the overlap region are processed one-by-one until a

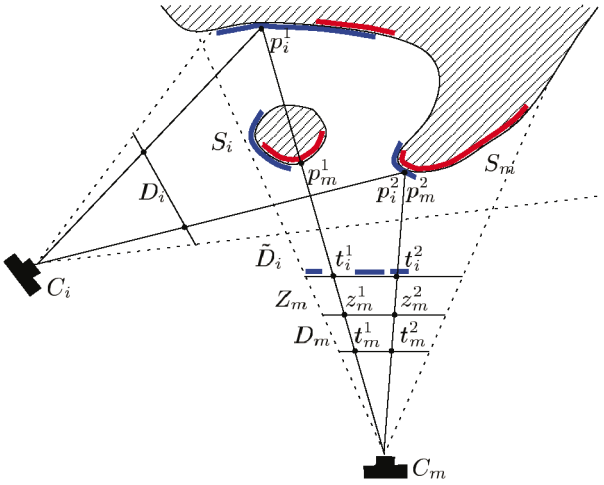


Fig. 6. Occlusions example: Point p_i^1 is visible from camera position C_i, but not from C_m. Since the detail map D_i is projected onto the image plane of D_m, this point must be discarded. The depth map Z_m corresponding to scan S_m is used to mask points of S_i which are not visible from C_m. In contrast, point p_i^2 is visible from C_m and its texture is recorded onto $\tilde{D}_i$. The red and blue curves identify the two scans S_m and S_i, respectively. The areas of $\tilde{D}_i$ that receive texture from scan S_i are also highlighted in blue.

prespecified number of matching pairs is found or until all points have been processed.

For a given sample (t_m, p_m) such that $t_m \in D_m$ has pixel coordinates (u, v), we define a point $t_i \in \tilde{D}_i$ with the same coordinates. Ideally, t_m and t_i mark the same image feature on D_m and $\tilde{D}_i$, i.e., scans S_m and S_i are perfectly aligned. In practice, t_i may be slightly offset with respect to the feature marked by t_m due to errors in the alignment of the scans. Our strategy is to search in the neighborhood of t_i for another point b_i that is a better match for t_m and to use the corresponding point $p_i = back\_projection(b_i, S_i, C_m) \in S_i$ as the match of p_m in the final alignment. The search for b_i is performed using a sliding-window cross-correlation approach which is outlined in Fig. 7.

The radius of the search region for the cross-correlation and the size of the correlation window are defined, taking into account the resolution of the scanning system and the maximum registration error after the geometry-based alignment. In our case, given a linear intersample distance of the scanner of approximately 2 mm and a residual error after the geometry-based alignment of less than 4 mm, we define a conservative radius of 7 mm (i.e., 14 pixels assuming a 0.5 mm resolution in the detail maps) for the search area around each point t_i. A cross-correlation is performed in this area using a window of 15 mm radius (i.e., 30 pixels), also a fairly conservative choice to ensure that significant features around each point are included.

The location within the search area that yields the highest value for the correlation coefficient defines the best match position b_i. Only pairs for which the correlation value is above a certain threshold are used to refine the alignment. We compute the correlation coefficient of n data points (t_m^k, t_i^k) according to Pearson's formula [26], $\rho^2 = s_{mi}^2 / s_{mm} s_{ii}$, where

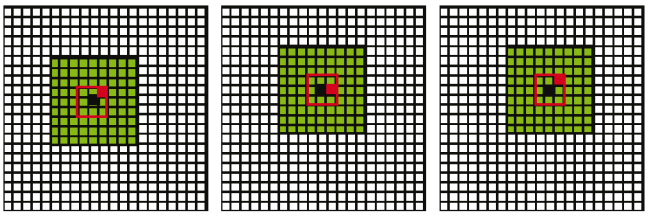


Fig. 7. Sliding-window search for the point maximizing cross-correlation between image neighborhoods. The search starts at point $t_i \in \tilde{D}_i$, marked in black. The green area is cross-correlated to the corresponding area in D_m. The operation is repeated for areas centered in all pixels within the search region outlined in red. The pixel corresponding to the center of the maximum cross-correlation area is marked in red. The search window is moved to the new center and a new iteration starts.

$$s_{mm} = \sum D_m(t_m^k)^2 - \frac{1}{n}\left[\sum D_m(t_m^k)\right]^2,$$
$$s_{ii} = \sum \tilde{D}_i(t_i^k)^2 - \frac{1}{n}\left[\sum \tilde{D}_i(t_i^k)\right]^2,$$
$$s_{mi} = \sum D_m(t_m^k)\tilde{D}_i(t_i^k) - \frac{1}{n}\sum D_m(t_m^k)\sum \tilde{D}_i(t_i^k).$$

Cross-correlation is known to be a time-consuming procedure. For efficiency, we replace an exhaustive search in a 14-pixel radius area with a sliding window approach that is faster to compute. We restrict the radius of the search area to 3 pixels and we allow it to slide in the direction of the best correlation for a predefined number of times. An obvious problem with this method is that the search may terminate by finding a local maximum of the correlation function and, thus, the global maximum is never reached. We compensate for this drawback by identifying a larger number of points than are actually needed. An alternative solution is to allow the search window to slide out of a local maximum with a given probability dependent on the surrounding values.

4.4 Multiple Iterations and Convergence

Our image-based registration method generally performs multiple iterations to allow each scan to adjust its position with respect to its neighbors. The number of iterations is typically determined by some convergence criterion. In our experiments, we use as a measure of convergence the change in the mean square error between consecutive iterations. If this change is small, the algorithm stops. At each iteration, the scans are considered for alignment one-by-one, in random order. The scan currently being aligned moves in search for a better position with respect to all other scans, which are considered temporarily fixed. Culling is used to speed up the procedure by considering only scans with intersecting bounding boxes.

This image-based registration algorithm has several advantages. It takes into account high-resolution information in the captured images to fine-tune the geometric alignment of scans. The fine-tuning is performed via image matching involving a simple and efficient cross-correlation procedure that is restricted to small areas around selected sample points. Specific image features need not be identified. Any image processing technique that allows for finding feature-rich regions in the input images is well-suited for the selection of the sample points. The sampling method presented is completely automatic and requires no

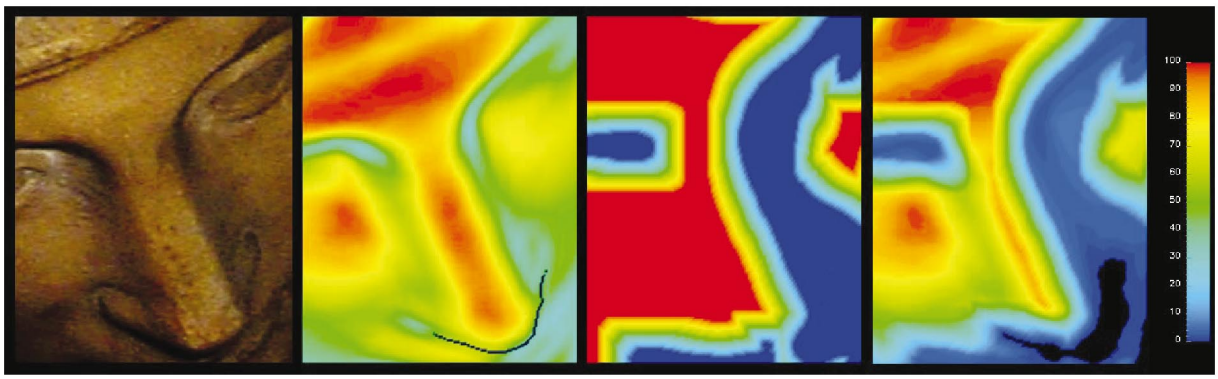


Fig. 8. Calculation of the final weight map for a section of a scan used to reconstruct the model in Fig. 1. (a) Intensity image. (b) Preliminary weight map based on the ratio of projected to true area for each pixel. (c) Preliminary weight map based on the photometric calculations. (d) Final weight image.

manual intervention. Occlusions are handled simply and efficiently by comparing depth values stored in precomputed depth buffers.

There is a potential residual registration error due to the lower resolution and line-of-sight error of the range scans. The effect of these two sources of error is alleviated by two factors. First, these errors are randomly distributed around a mean value so that using a large number of sample points and having multiple overlapping scans at each point tends to average out the error. Second, both errors are nearly perpendicular to the image plane of the albedo and normal maps and contribute minimally to ghosting artifacts in the resulting textures. The line-of-sight error affects all methods that rely on reprojecting matching image points on the corresponding range scans as a step in computing the registration transformation [13], [22], [23], [25], although it has greater impact in our system given our assumption of lower resolution range scans compared to the intensity images. In applications where a greater registration accuracy in the direction normal to the surface must be achieved, the normal maps could be used to locally refine the geometry obtained from the range scans and to estimate with increased resolution the position of points on the surface. Increased accuracy could also be obtained by interleaving steps of registration and line-of-sight error correction. These ideas will be explored in future work.

5 Texture Reconstruction

The goal of texture reconstruction is to generate seamless texture maps covering the model. Our approach for combining textures is similar to those presented by Pulli et al. [23] [31], Ofek et al. [33], Pighin et al. [34], and Neugebauer and Klein [20]. However, we use processed albedo and normals maps, rather than acquired textures, and weights that reflect the confidence in these maps for each scan in addition to geometric confidence measures. We compute view independent maps, rather than the view dependent maps used in some methods. We use all possible data sets for each surface, rather than just a selected two or three. Finally, we discuss here a number of issues important in the efficient implementation of the reconstruction.

To make maximum use of the acquired data, texture maps are recalculated based on the integrated mesh. For a given scan, albedo and normal data may now be obtained in regions where there is no geometry available from that scan.

The input to the texture reconstruction process consists of the collection of height-field patches that form a partition of the model surface and the finely registered scans with the recomputed albedo, normal, and weight maps. A mapping between geometry and corresponding textures is defined as an orthogonal projection in the direction that maximizes the projected patch area (see Fig. 9.)

The pixel size of a texture is determined based on the projected size of the corresponding patch and on a user-specified parameter representing the desired number of pixels per unit of length. All individual texture maps are combined using the weight maps defined in Section 3. Fig. 8 shows the calculation of the final weight map for a section of a scan as the combination of two preliminary weight maps. For illustrative purposes, the relative weights in each weight image are shown with a color map with black as zero and the hues from blue to red corresponding to increasing nonzero weights. Fig. 8a shows a section of an intensity image for one of the scans used to create the model in Fig. 1. Fig. 8b shows the first preliminary weight map for this section of the scan based on the ratio of projected to true area for each pixel. This ratio is computed as the cosine of the surface normal to the camera direction divided by the square of the distance to the camera. Occlusion boundaries, such as the tip of the nose in this image, are detected and a weight of zero is assigned to each occlusion boundary. By setting these pixels to zero, occlusion boundaries are treated in the same manner as the edges of the scan. Fig. 8c shows the second preliminary weight map based on the photometric calculations. If a surface normal can be computed from photometric data for a pixel, it is assigned a high weight. If such a normal cannot be computed, but there is an underlying surface normal value, a much lower, but nonzero value is assigned. A smooth transition (indicated by the transition from red to blue in Fig. 8c) is applied between the photometric and nonphotometric areas so that there will not be a sudden spatial break in the contribution of this scan to the final result. The values of Fig. 8b and

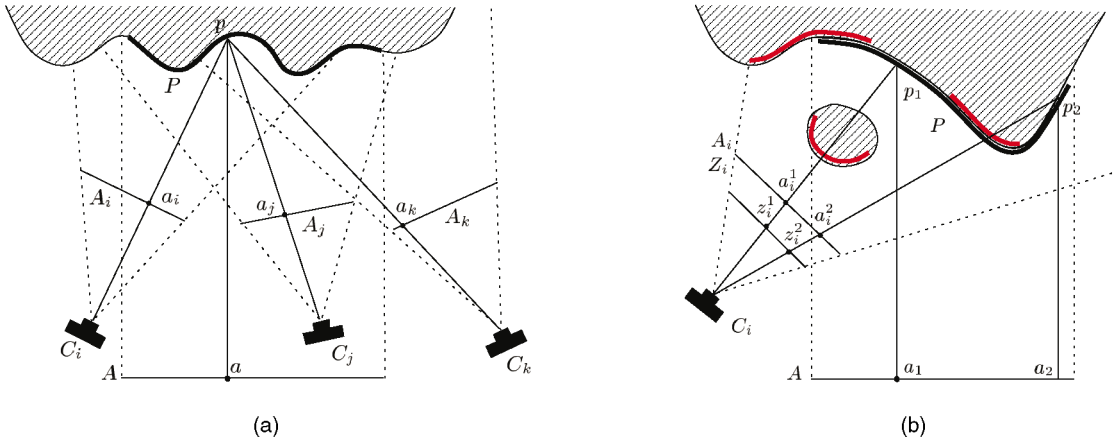


Fig. 9. (a) Two-dimensional diagram of texture remapping. An albedo map A is computed for patch P, which is parameterized as a height field onto the image plane of A. Three scans, S_i, S_j, and S_k, have camera frusta intersecting patch P and are considered for the remapping. For each pixel a on texture A, consider the corresponding point p on patch P. This point projects onto points a_i, a_j, and a_k in the albedo maps of the three scans and onto w_i, w_j, and w_k in the corresponding weight maps. The resulting albedo value for a is $(w_i a_i + w_j a_j + w_k a_k)/(w_i + w_j + w_k)$. This weighting scheme avoids discontinuities at transitions between patches or between scans within the same patch. (b) Examples of occlusions.

Fig. 8c are multiplied and rescaled to form the final weight image, shown in Fig. 8d. The last step in computing the final weights in Fig. 8d is to compute a smooth transition from areas with nonzero weight to the edge values which have a weight of zero. The weights are stored as 0-255 values. The quantization accounts for the large black area in the lower right of this example, where the values in both images Fig. 8b and Fig. 8c are low, and the values are further reduced by the transitioning to the nose silhouette edge. Because weights are generated per scan and not per patch, transitions across patch boundaries are not visible. Also, since the weights for each scan decrease with distance to the scan border, scan-to-scan boundaries are not visible.

The computation of a value for each texture pixel is illustrated in Fig. 9. In this example, an albedo map A is computed for patch P. The three scans S_i, S_j, and S_k have camera frusta that intersect the bounding box of patch P and are considered for the reconstruction.

A straightforward solution to the remapping problem is to use ray-casting. However, the time required to process all ray-mesh intersections makes this procedure prohibitively expensive. Another method to combine the albedo maps of the three scans into A is to define an orthogonal camera and then render P multiple times, with each of the albedo and weight maps from the three scans as its texture. The rendered images are accumulated using weighted averaging. This approach is also inefficient and produces incorrect results in the presence of occlusions. Consider the situation in Fig. 9b: Points p_1 and p_2 receive values from the maps associated with scan S_i, even though they are occluded when rendered from camera position C_i. Similarly to the method described in Section 4, depth values are used to filter out occluded points.

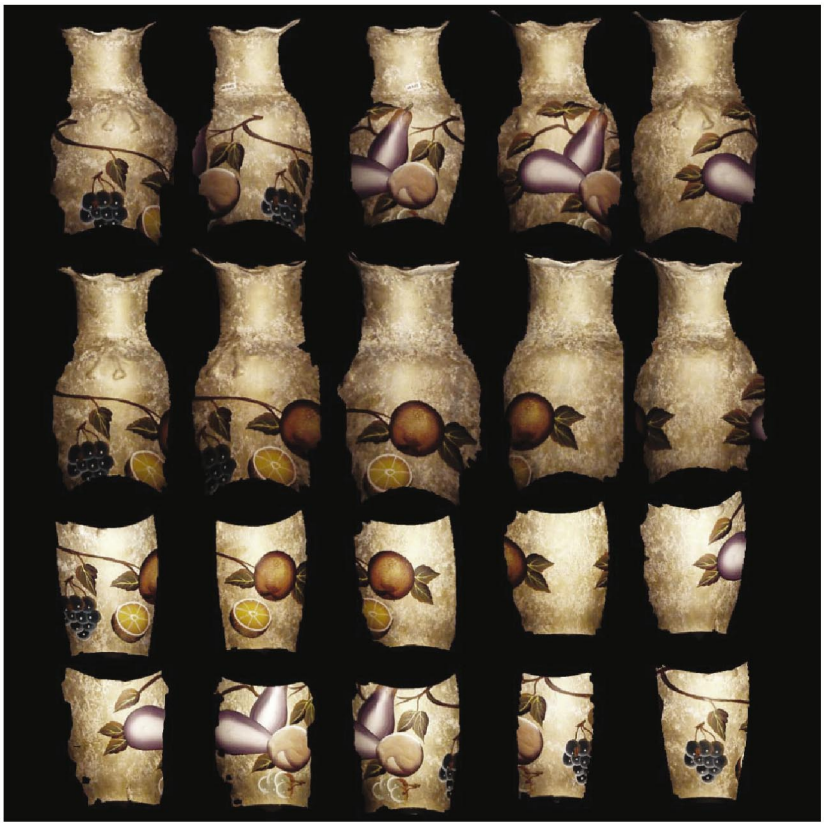
Specifically, to compute an albedo map A for a patch P, accumulation buffers for albedo, normals, and weights are created and initialized. P is then sampled at points corresponding to each pixel of A. This can be done by defining an orthogonal camera that matches the projection of P onto A, then scan-converting the primitives in patch P into a z-buffer.

We use the inverse viewing transformation to convert each pixel (u, v) and the associated depth to world coordinates that are stored in a *point map*. We start the processing of each overlapping scan by loading its albedo, normal, weight, and depth map, and by defining a perspective camera matching the actual camera that was used to capture the intensity images. We then consider each pixel of A. We retrieve the corresponding patch point from the point map and use the viewing transformation to map the point into pixel coordinates (u, v) and its depth relative to the perspective camera. We compare the depth of the point with the value stored in the depth map Z_i for the scan. If the depth is larger than the stored value (minus a small offset to account for numerical errors), then the point is occluded and, therefore, no longer considered. Otherwise, we fetch the albedo, normal, and weight values at location (u, v) in the corresponding maps and update the accumulation buffers. After all the scans have been processed, accumulated albedo and normal values are divided by the accumulated weight.

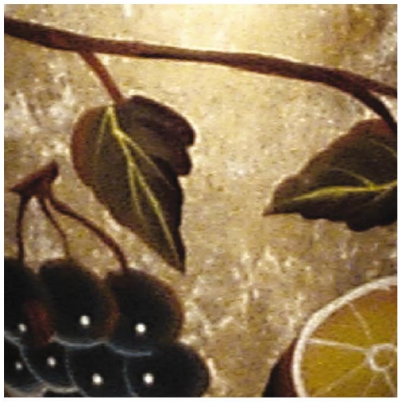
By presampling the patch geometry and by fetching values from all maps simultaneously, computations are streamlined. Occlusions are handled simply and efficiently by comparing depth values in precomputed depth buffers. Image and geometric information are loaded on demand to allow processing of large sets of scans that do not fit in memory. The only information stored for all scans is the view-frustum of the capture camera and the bounding box of each patch to allow view frustum culling at the bounding box level.

6 Results

We demonstrate our new methods with results for three qualitatively different test objects. One is a 26 cm tall vase, with few geometric features and sharp surface color variations. The second is a 12 cm tall piece of clay with no surface color variations, but a lot of small scale geometric variations. The third object is a section of a large marble



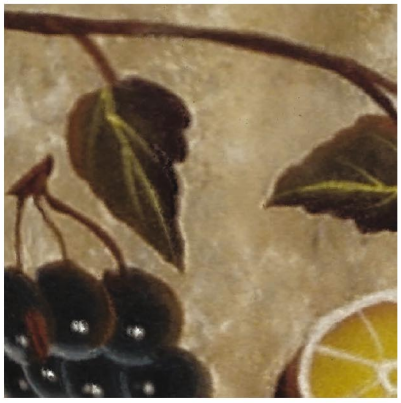
(a)



(b)

(c)

(d)



(e)

(f)

(g)

Fig. 10. The vase data set. (a) The 20 scans used to reconstruct the vase. (b) Intensity image captured by the scanner. (c) Computed albedo map. (d) Reconstructed albedo without image-based registration. (e) Same as (d) with image-based registration. (f) Photograph of the vase. (g) Rendering of a computer model of the vase reconstructed from the 20 scans.

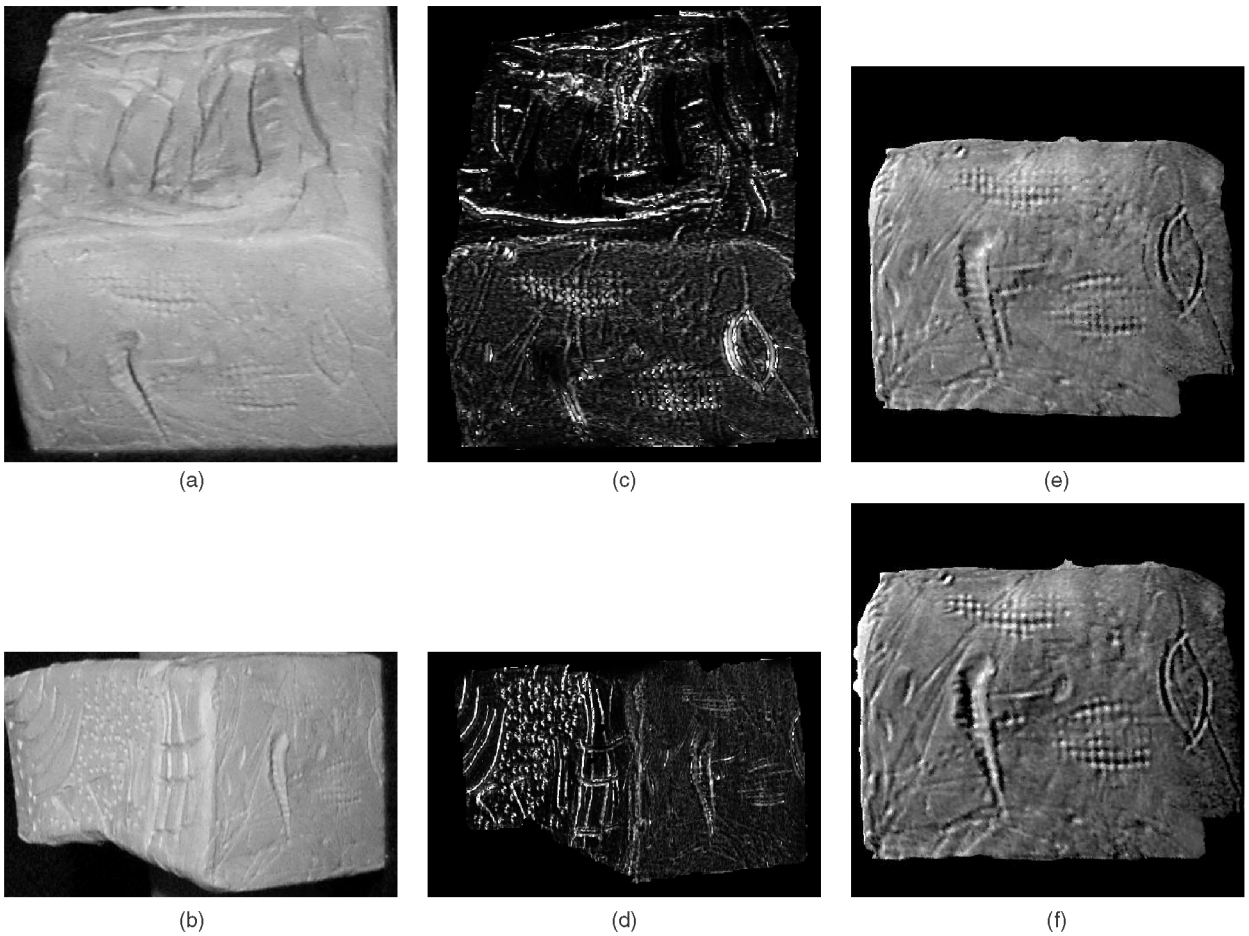


Fig. 11. (a), (b) Intensity images associated with two scans of a clay object. (c), (d) Detail images for the two scans, derived from the normal maps. (e) Result after ICP registration (lit from the far left to reveal relief.) (f) Result after the texture alignment. Most of the ghosting has been removed, and the true features are more clearly defined.

statue with subtle variations in both geometry and color, shown in Fig. 1.

For the vase, the major challenges are accurate alignment for an object with few geometric features and maintainance of sharp textures. The use of image-based alignment improves the geometric alignment, and maintains sharpness, as illustrated in Fig. 10. Twenty scans, shown textured-mapped with the original intensity images in Fig. 10a, were acquired to cover the full vase. Fig. 10b shows a portion of one of the five captured intensity images for one of the scanner positions. The acquired image contains both specific illumination and some noise. The albedo map obtained from the five calibrated images, mapped on the corresponding scanned geometry, is shown in Fig. 10c. The effects of illumination have been removed, but noise remains. Fig. 10d shows the albedo maps reconstructed from multiple overlapping scans, using geometry-based registration only. The noise has been damped out by averaging, but the imperfect alignment causes ghosting artifacts. The same map after the image-based registration refinement is shown in Fig. 10e. On the 2 to 3 mm thick grapes stems, ghosting artifacts are far less apparent. A photograph of the vase, lit by a small lamp from the front, is shown in Fig. 10f. The full reconstructed vase, lit from the direction of the viewpoint with parallel light, is rendered in Fig. 10g. We did not attempt to simulate the directional characteristics of the small lamp illuminating the vase in the photograph. The viewing parameters are similar, but not identical. The 20 scans used comprise a total of 418,000 range data points. The final reconstructed model is a 200,000 triangle mesh. We started with 150 sample points per detail image. For each scan, we limited the number of matching pairs used to a maximum of 10 in each region of overlap with another scan. For this test, the image-based registration required a total of 115 minutes on a 450MHz Pentium II, using 35MB of memory and six iterations. Texture reconstruction was completed in 30 minutes.

The clay object is shown in Fig. 11. Fig. 11a and Fig. 11b show intensity images associated with two scans of the clay. The two scans have one side of the block in common. The common side can be identified visually; however, the illumination for the two scans is quite different. For these scans, which show no albedo variations, detail images were computed from the normals images. To obtain values that were independent of the coordinate system, the detail images were computed by taking the dot product of the normal at each pixel with the average of normals in a small neighborhood. The resulting detail images for the two scans are shown in Fig. 11c and Fig. 11d. Since the common

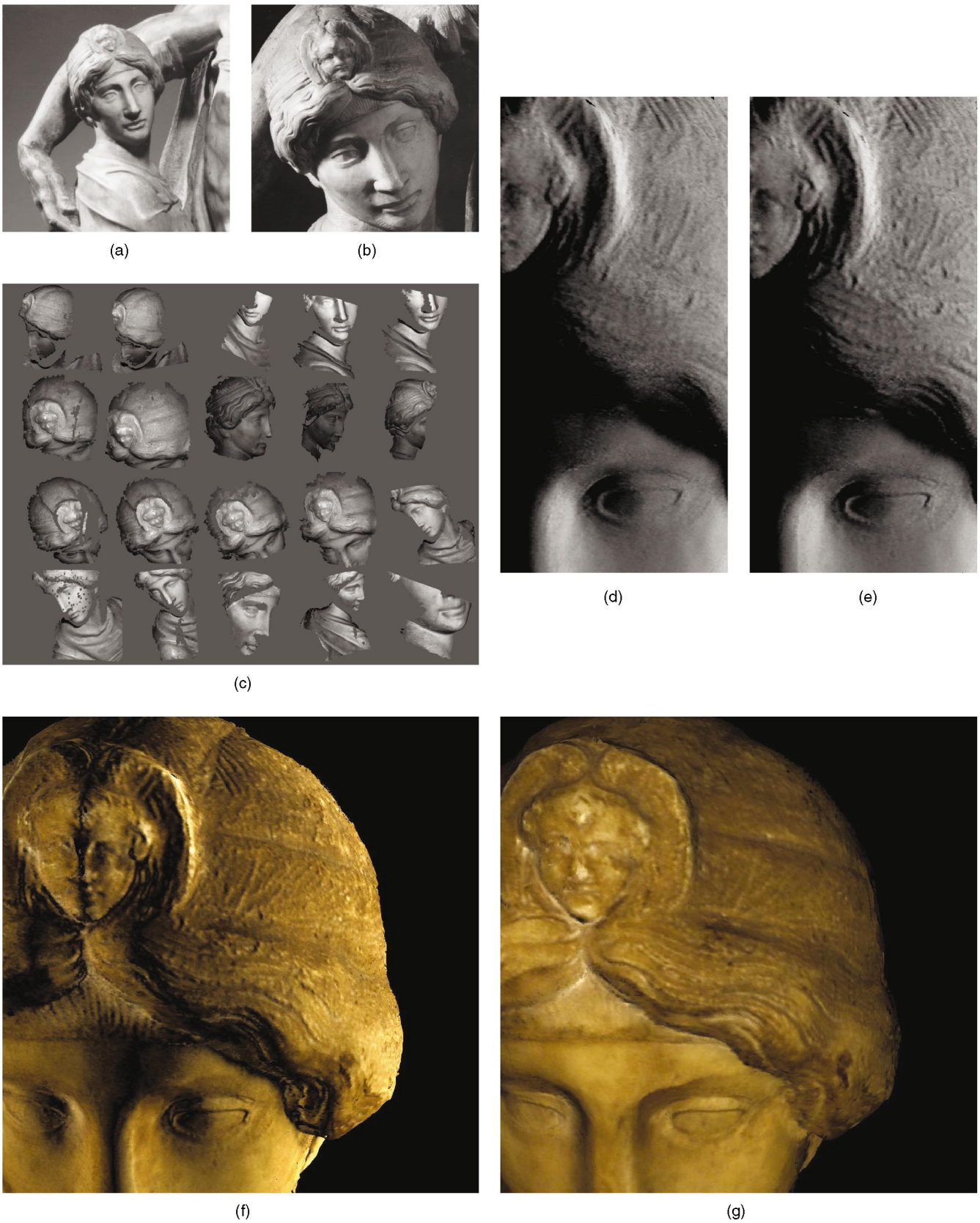


Fig. 12. (a) Black and white photograph showing the section of statue in Fig. 1 in context. (b) Black and white photograph showing a closeup view. (c) The 20 scans used to reconstruct the head. (d), (e) Normal map before and after image-based registration. (f), (g) The reconstructed model of the statue rendered under two different lighting conditions.

surface between the two scans is almost flat, ICP did a poor job aligning the scans, as shown in the result in Fig. 11e (lit from the far left to reveal relief.) Comparing the small features in Fig. 11e with the features apparent in the original intensity images shows considerable ghosting. Fig. 11f shows the result after the texture alignment. Most of the ghosting has been removed and the true features are more clearly defined.

The views required to scan the section of statue in Fig. 12 were restricted by its setting shown in Fig. 12a. The 20 scans captured are shown in Fig. 12c. For the statue, the major challenge is obtaining sharp details in the normals maps that indicate small toolmarks in the geometry, as seen in the black and white photograph in Fig. 12b. Fig. 12d and Fig. 12e show a comparison of the reconstructed normals maps before and after image-based registration. Illumination from a glancing angle accentuates the fine details on the head ornament. The details are clearer after the image-based registration. The reconstructed model can be relit using the detail normals and albedo. Fig. 12f and Fig. 12g show the statue under two different lighting conditions. The 20 scans used comprise a total of 207,000 range data points. The final reconstructed model is a 200,000 triangle mesh. As for the vase, we started with 150 sample points per detail image and limited the number of matching pairs used to a maximum of 10 for each pair of overlapping scans. Image-based alignment of the 20 source scans completed in 148 minutes for six iterations. Texture reconstruction required an additional 35 minutes.

7 Conclusions

We have demonstrated an approach for generating texture maps for the efficient representation of scanned objects with high visual quality. We use a novel image-based registration algorithm that takes into account high-resolution information in the captured images. The geometric alignment is improved by matching image structure around automatically selected points. The refined alignment allows us to reconstruct sharp textures. Our texture reconstruction approach uses a unique weighting scheme to combine the best information at each texture pixel that reduces noise and which eliminates the evidence of any scan or patch boundaries.

We are considering future work in a number of directions. We are exploring the interleaving of geometric ICP, image-based alignment, and line-of-sight error correction in our pipeline. We conjecture that accounting for line-of-sight errors during alignment will produce better registration results. We also seek to improve our method for generating texture patches. Currently, we use a relatively simple greedy approach that does not guarantee an optimal set of patches.

While three-dimensional scanning is being used with increasing frequency, it remains a relatively expensive and labor-intensive process. Significant research is required to refine the entire acquisition and model building pipeline. Focusing on methods that employ inexpensive hardware to produce high visual quality will make this technology accessible to a much broader audience.

References

[1] C. Rocchini, P. Cignoni, C. Montani, and R. Scopigno, "Multiple Textures Stitching and Blending on 3D Objects," *Proc. 10th Eurographics Workshop Rendering,* June 1999.

[2] M. Polleyfeys, R. Koch, M. Vergauwen, and L. Van Gool, "Hand-Held Acquisition of 3D Models with a Video Camera," *Proc. Second Int'l Conf. 3-D Digital Imaging and Modeling,* pp. 14-23, Oct. 1999.

[3] F. Bernardini and H. Rushmeier, "The 3D Model Acquisition Pipeline," *Eurographics 2000 State of the Art Reports,* Eurographics Associations, 2000.

[4] G. Roth, "Registering Two Overlapping Range Images," *Proc. Second Int'l Conf. 3-D Digital Imaging and Modeling,* pp. 191-200, Oct. 1999.

[5] D. Zhang and M. Hebert, "Experimental Analysis of Harmonic Shape Images," *Proc. Second Int'l Conf. 3-D Digital Imaging and Modeling,* pp. 209-219, Oct. 1999.

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

[7] Y. Chen and G.G. Medioni, "Object Modeling by Registration of Multiple Range Images," *Image and Vision Computing,* vol. 10, no. 3, pp. 145-155, 1992.

[8] 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, May 1996.

[9] K. Pulli, "Multiview Registration for Large Data Sets," *Proc. Second Int'l Conf. 3-D Digital Imaging and Modeling,* pp. 160-168, Oct. 1999.

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

[11] G. Turk and M. Levoy, "Zippered Polygon Meshes from Range Images," *Proc. SIGGRAPH '94,* A. Glassner, ed., pp. 311-318, July 1994.

[12] B. Curless and M. Levoy, "A Volumetric Method for Building Complex Models from Range Images," *SIGGRAPH 96 Conf. Proc.,* H. Rushmeier, ed., pp. 303-312, Aug. 1996.

[13] A. Johnson and S. Kang, "Registration and Integration of Textured 3-D Data," *Proc. Int'l Conf. Recent Advances in 3-D Digital Imaging and Modeling,* pp. 234-241, May 1997.

[14] 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, Oct.-Dec. 1999.

[15] M. Soucy, G. Godin, and M. Rioux, "A Texture-Mapping Approach for the Compression of Colored 3D Triangulations," *The Visual Computer,* vol. 12, pp. 503-513, 1996.

[16] S.R. Marschner, "Inverse Rendering for Computer Graphics," PhD thesis, Cornell Univ., 1998.

[17] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle, "Multiresolution Analysis of Arbitrary Meshes," *SIGGRAPH '95 Conf. Proc.,* R. Cook, ed., pp. 173-182, Aug. 1995.

[18] A.W.F. Lee, W. Sweldens, P. Schröder, L. Cowsar, and D. Dobkin, "MAPS: Multiresolution Adaptive Parameterization of Surfaces," *SIGGRAPH '98 Conf. Proc.,* M. Cohen, ed., pp. 95-104, July 1998.

[19] J. Maillot, H. Yahia, and A. Verroust, "Interactive Texture Mapping," *Proc. SIGGRAPH '93,* pp. 27-34, 1993.

[20] P. Neugebauer and K. Klein, "Texturing 3D Models of Real World Objects from Multiple Unregistered Photographics Views," *Computer Graphics Forum,* vol. 18, no. 3, pp. C–245-C–256, 1999.

[21] C. Schütz, T. Jost, and H. Hügli, "Multi-Feature Matching Algorithm for Free-Form 3D Surface Registration," *Proc. Int'l Conf. Pattern Recognition,* Aug. 1998.

[22] S. Weik, "Registration of 3-D Partial Surface Models Using Luminance and Depth Information," *Proc. Int'l Conf. Recent Advances in 3-D Digital Imaging and Modeling,* pp. 93-100, May 1997.

[23] K. Pulli, "Surface Reconstruction and Display from Range and Color Data," PhD thesis, Dept. of Computer Science and Eng., Univ. of Washington, 1997.

[24] R. Szeliski and H. Shum, "Creating Full Panoramic Mosaics and Environment Maps," *Proc. SIGGRAPH '97,* pp. 251-258, 1997.

[25] E. Gagnon, J.-F. Rivest, M. Greenspan, and N. Burtnyk, "A Computer-Assisted Range Image Registration System for Nuclear Waste Cleanup," *IEEE Trans. Instrumentation and Measurement,* vol. 48, no. 3, pp. 758-762, 1999.

[26] R.C. Gonzalez and R.E. Woods, *Digital Image Processing*. Addison-Wesley, 1993.
[27] Y. Sato, M. Wheeler, and K. Ikeuchi, "Object Shape and Reflectance Modeling from Observation," *Proc. SIGGRAPH '97*, pp. 379-388, 1997.
[28] R.J. Woodham "Photometric Method for Determining Surface Orientation from Multiple Images," *Optical Eng.*, vol. 19, pp. 139-144, 1980.
[29] H. Rushmeier, F. Bernardini, J. Mittleman, and G. Taubin, "Acquiring Input for Rendering at Appropriate Level of Detail: Digitizing a Pietà," *Proc. Ninth Eurographics Workshop Rendering*, June 1998.
[30] M. Levoy et al., "The Digital Michelangelo Project: 3D Scanning of Large Statues," *SIGGRAPH 2000 Conf. Proc.*, K. Akeley, ed., pp. 131-144, 2000.
[31] K. Pulli, H. Abi-Rached, T. Duchamp, L.G. Shapiro, and W. Stuetzle, "Acquisition and Visualization of Colored 3D Objects," *Proc. Int'l Conf. Pattern Recognition*, pp. 11-15, Aug. 1998.
[32] H. Rushmeier and F. Bernardini, "Computing Consistent Normals and Colors from Photometric Data," *Proc. Second Int'l Conf. 3-D Digital Imaging and Modeling*, pp. 99-108, Oct. 1999.
[33] E. Ofek, E. Shilat, A. Rappoport, and M. Werman, "Multiresolution Textures from Image Sequences," *IEEE Computer Graphics & Applications*, vol. 17, no. 2, pp. 18-29, Mar. 1997.
[34] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. Salesin, "Synthesizing Realistic Facial Expressions from Photographs," *Proc. SIGGRAPH '98*, pp. 75-84, 1998.

Fausto Bernardini received the Laurea in Ingegneria Elettronica from the Università La Sapienza, Rome, Italy, in 1990, and the PhD degree in computer science from Purdue University in 1996. He joined IBM in 1997. Currently, he is a research staff member and manager of the Visual and Geometric Computing Group at the IBM T.J. Watson Research Center. His research interests include interactive visualization of large models, 3D scanning and reconstruction, and geometric modeling. He has been one of the principal investigators in a research effort to build a digital model of Michelangelo's Florentine Pietà. Dr. Bernardini is the author of more than 40 papers and patents on applied computational geometry and computer graphics.

Ioana M. Martin received the PhD degree in computer science from Purdue University in 1996. She is a research staff member in the Visual Technologies Department at the IBM T.J. Watson Research Center. Her research interests include interactive computer graphics, visualization, and image processing. She has developed and implemented methods for processing and rendering large data sets, including progressive refinement and multiresolution techniques with applications to structural biology. Since 1998, when she joined IBM, she has worked on various projects, including adaptive delivery of 3D models over networks, 3D data compression, model reconstruction, and subdivision surface modeling. She is a member of the IEEE Computer Society.

Holly Rushmeier received the BS, MS and PhD degrees in mechanical engineering from Cornell University in 1977, 1986, and 1988, respectively. She is a research staff member at the IBM T.J. Watson Research Center. Since receiving the PhD degree, she has held positions at the Georgia Institute of Technology and the National Institute of Standards and Technology. In 1990, she was selected as a US National Science Foundation Presidential Young Investigator. She has served as papers chair or cochair for the ACM SIGGRAPH Conference, the IEEE Visualization Conference, and the Eurographics Rendering Workshop. From 1996 to 1999, she was editor-in-chief of the *ACM Transactions on Graphics*. Her research interests include data visualization, rendering algorithms, and acquisition of input data for computer graphics image synthesis.