Foundations of Image Understanding, L. S. Davis, ed., ©Kluwer, Boston, 2001, pp. 469-489.

Chapter 16

VOLUMETRIC SCENE RECONSTRUCTION FROM MULTIPLE VIEWS

Charles R. Dyer, University of Wisconsin

Abstract A review of methods for volumetric scene reconstruction from multiple views is presented. Occupancy descriptions of the voxels in a scene volume are constructed using shape-from-silhouette techniques for binary images, and shape-from-photo-consistency combined with visibility testing for color images.

1. INTRODUCTION

The more the marble wastes, the more the statue grows. *Michelangelo*

Automatic construction of photorealistic three-dimensional models of a scene from multiple images is important for applications such as interactive visualization of remote environments or objects by a virtual video camera, and virtual modification of a real scene for augmented reality tasks. While considerable research has focused on this problem, recently there has been considerable progress in developing techniques that build volumetric scene models. The goal of this paper is to describe methods that use this approach.

Scene reconstruction methods have traditionally been based on image matching, using either intensity-based (direct) methods or feature-based methods. This class includes all multi-view stereo techniques that compute correspondences across images and then recover 3D structure by triangulation and surface fitting. This approach is especially effective with video sequences, where tracking techniques simplify the correspondence problem. Some of the disadvantages of this approach are that (1) views must often be close together (i.e., small baseline) so that correspondence techniques are effective; (2) correspondences must be maintained over many views spanning large changes in viewpoint; (3) many partial models must often be computed with respect to a set of base viewpoints, and these surface patches must then be fused into a single, consistent model; (4) if sparse features are used, a parameterized surface model must be fit to the 3D points to obtain the final dense surface reconstruction; and (5) there is no explicit handling of occlusion differences between views.

An alternative approach to scene reconstruction is based on computations in three-dimensional scene space in order to construct the volumes or surfaces in the world that are consistent with the input images. We call this approach *volumetric scene modeling*. By replacing the image-based search problem used in the first approach by a scene-based search, volumetric scene modeling avoids the disadvantages listed above.

Volumetric scene modeling explicitly represents a world coordinate frame and the volume of space in which the scene occurs, and makes occupancy decisions about whether a volumetric primitive contains objects in the scene. Scene-space methods allow widely-separated views but generally depend on calibrated cameras to determine the absolute relationship between points in space and visual rays.

2. VOLUMETRIC REPRESENTATIONS

Volumetric modeling of scene space assumes there is a known, bounded area in which the objects of interest lie. This area is frequently assumed to be a cube surrounding the scene. The most common approach to representing this volume is as a regular tessellation of cubes, called voxels, in Euclidean 3-space, though other representations have been studied.

Octrees [10, 44, 55] enable more a space-efficient encoding when the scene contains large transparent areas. Disparity space representations [24, 58, 8] are defined relative to one camera frame, describing space in terms of unit disparity increments in (x, y, d) space. To explicitly model visibility with respect to a particular camera, a generalized disparity space, (x, y, d, k), can be used to indicate whether camera k can see voxel (x, y, d) [58]. Rather than use a fixed set of disparity planes, layer representations [4] define a scene-dependent collection of approximately planar regions in the scene.

If the input cameras are not completely calibrated and only the fundamental matrices are known for pairs of cameras, a projective space representation [25, 46] can be defined using two of the cameras as basis views, creating a (x_l, y_l, x_r) projective space voxelization. Ray space representations [27, 38] directly operate on the set of visual rays from each camera's optical center, avoiding volumetric discretization issues. See [57] for a recent review of some of these alternative representations. A survey on volumetric-based methods has also been written [51].

Scene reconstruction using a voxel-based representation is defined by an occupancy classification of each volume element into one of a discrete set of labels. This is usually a binary decision (transparent or opaque) or a ternary decision (transparent, opaque or unseen), though some methods include a real-valued degree of opacity. We assume in the remainder of this paper the use of a regular-tessellation, binary-valued voxel model.

From a reconstructed volumetric model, a polygonal surface representation can be constructed using the Marching Cubes algorithm [36]. If the model is used for synthesizing new views, a variety of efficient rendering methods are known, including raycasting, splatting, and shear-warp [39].

3. SHAPE FROM SILHOUETTES

In this section we describe methods of volumetric scene reconstruction from a set of silhouette images. A silhouette image is a binary image, with the value at a point indicating whether or not the visual ray from the optical center through that image point intersects an object surface in the scene. Thus each pixel is either a silhouette point or a background point. The binary images can be obtained by segmentation algorithms or blue-screen techniques. Combined with calibration information for each camera, each point in a silhouette defines a ray in scene space that intersects the object at some unknown depth along this ray. The union of these visual rays for all points in the silhouette defines a generalized cone within which the 3D object must lie.

The intersection of the generalized cones associated with a set of cameras defines a volume of scene space in which the object is guaranteed to lie. Of course, the volume only approximates the true 3D shape, depending on the number of views, the positions of the viewpoints, and the complexity of the object. In particular, since concave patches are not observable in any silhouette, a silhouette-based reconstruction encloses the true volume. Laurentini [31] characterized the best approximation, obtainable in the limit by an infinite number of silhouettes captured from all viewpoints outside the convex hull of the object, as the *visual hull*. For 2D scenes the visual hull is equal to the convex hull of the object. For 3D scenes the visual hull is contained in the convex hull, where concavities are not removed but hyperbolic regions are.

In practice, only a finite number of silhouettes are combined to reconstruct the scene, resulting in an approximation that includes the visual hull as well as other scene points. Figure 16.1 shows an example of the volume constructed from three 1D views of a 2D scene.

Many algorithms have been developed for constructing volumetric models from a set of silhouette images [1, 5, 9, 31, 32, 33, 37, 38, 42, 43, 44, 53, 54, 55]. Starting from a bounding volume that is known to enclose the entire scene, the volume is discretized into voxels and the task is to create a voxel occupancy description corresponding to the intersection of back-projected silhouette cones.

The main step in these algorithms is the intersection test. Some methods back-project the silhouettes, creating an explicit set of cones that are then intersected either in 3D (e.g., [43, 54]), or in 2D after projecting voxels into the

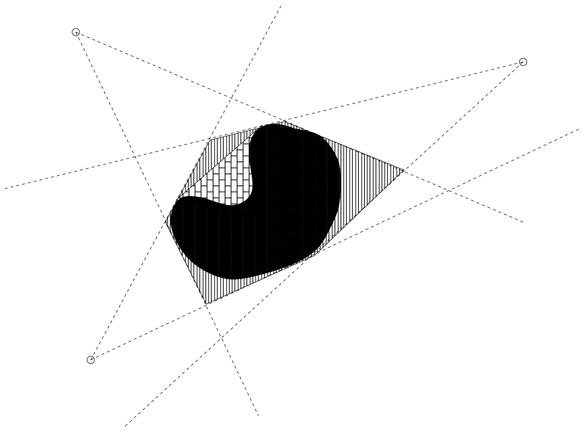


Figure 16.1 Reconstruction from three silhouettes. The generalized cones associated with the three images result in a reconstruction that includes the object (black), points in concavities that are not visible from any viewpoint outside the convex hull of the object (brick texture), and points that are not visible from any of the three given views (gray).

images (e.g., [23, 44]). Alternatively, it can be determined whether each voxel is in the intersection by projecting it into all of the images and testing whether it is contained in every silhouette [55].

To make this scene space traversal more efficient, most methods use an octree representation and test voxels in a coarse-to-fine hierarchy. That is, given an initial cube that encloses the entire scene, the current voxel is projected into all the images and tested to determine whether it intersects the silhouette in each image. If the projected voxel does not intersect the silhouette in at least one image, the voxel is removed, i.e., marked transparent. If the projected voxel intersects only silhouette pixels in every image, the voxel is marked opaque. Otherwise, the voxel intersects both background and silhouette points in the images, so it is subdivided into octants and each sub-voxel is processed recursively.

The shape-from-silhouette problem has also been formulated as an optimization problem where the global minimum of an energy function is computed [53]. The energy function contains a term that specifies the likelihood that a voxel is opaque or transparent based on the images' intensities, and a term that specifies the degree of smoothness of the labels in a neighborhood of voxels.

Shapes reconstructed from silhouettes have been used successfully in a variety of applications, including virtual reality [41], real-time human motion modeling [9, 40], constructing light fields and lumigraphs [22], and building an initial coarse scene model [14]. In applications such as real-time image-based rendering of dynamic scenes, where an explicit scene model is not an essential intermediate step, new views can be rendered directly using visual ray intersection tests [38].

An alternative approach to shape from silhouettes is the work on shape from occluding (aka apparent) contours (e.g., [11, 59]). In this case, rather than a volumetric reconstruction, a surface description is recovered from the occluding contours (i.e., borders of the silhouettes) in a dense sequence of views. These methods compute the object surface from the envelope of visual rays through corresponding points on successive images' occluding contours (often based on the epipolar parameterization), which are generated by surface points that slide over the object as the viewpoint changes. Because these methods are based on tracking image points and not on scene space analysis, they will not be discussed further here.

4. SHAPE FROM PHOTO-CONSISTENCY

When the input images are grayscale or color rather than the binary images processed by shape-from-silhouette methods, the additional photometric information can be used to improve the 3D reconstruction process. A set of images can be thought of as defining a set of constraints on a 3D scene reconstruction, in that a valid 3D scene model that is projected using the camera matrices associated with the input images must produce synthetic images that are the same as the corresponding real input images. This image-reproduction test verifies a hypothesized 3D scene model by comparing real and synthesized images rather than evaluating the consistency of 2D or 3D features that are derived from the input images. The definition of "same as" depends, of course, on the characteristics and accuracy of the scene model and on the rendering process. A complete scene model includes not only surface geometry but also surface reflectance models and scene illumination. There may be many 3D scenes that are consistent with a particular set of images, so image-reproduction consistency does not guarantee a unique reconstruction. The family of all reconstructions define an equivalence class with respect to the property "reproduction consistent with respect to a given set of photographs." In fact, without additional information or biases, this equivalence class of reconstructions is the best we can do based on direct comparison with the images.

Image-reproduction consistency can be defined as the *photo-consistency* [29, 47] of all visible surface points with respect to each image. That is, a point on a scene surface is photo-consistent with a set of images if, for each image in which that point visible, the image irradiance of that point is equal to the intensity at the corresponding image pixel.

Photo-consistency is related to Leclerc et al.'s [34, 35] *self-consistency* methodology for evaluating the performance of multi-view point correspondence algorithms. This method evaluates consistency of point correspondences across multiple images in terms of their consistency with respect to a single 3D surface element. So, whereas this methodology compares hypotheses derived from images for consistency with respect to a fixed world, photo-consistency takes the inverse approach of verifying a hypothesis about the shape of the world by testing the consistency of its projections with a given set of images.

Use of the photo-consistency constraint avoids several difficulties found in multi-image stereo reconstruction methods based on finding point correspondences [6, 26] or contour correspondences [11, 59] between images, and then triangulating to recover 3D scene structure. First, assuming camera projection matrices are known, photo-consistency requires image rendering using forward projection and pixel comparison operations, whereas accurate point correspondences are notoriously difficult to compute, especially in regions of nearly homogeneous intensity. Given a 3D scene model, photo-consistency decides only whether the projection of the model is consistent with all of the images and therefore individual point correspondences are not needed (assuming they are not required to construct a hypothesized scene model). Second, obtaining *dense* correspondences is especially hard, meaning that correspondence-based methods will either skip points, resulting in a sparse reconstruction that does not use all the image points, or else correspondence errors will lead to low-quality models even when image consistency is high. Szeliski [56] made a similar argument in justifying his use of image-prediction error for evaluating stereo algorithms.

Building 3D reconstructions based on photo-consistency requires the ability to test whether a surface point could produce the image irradiance values at the pixels into which the point projects in each image. To render these pixels requires additional assumptions or knowledge about the camera models and scene model, i.e., camera positions, scene geometry, surface reflectance, and illumination. One important special case is when all surfaces are assumed to satisfy a Lambertian reflectance model so that every surface appears equally bright in all directions regardless of the illumination. In this case, once the camera models and scene geometry are specified, an image can be synthesized and its photo-consistency can be tested against the input images. Throughout this paper we assume that a Lambertian model approximately holds for all scene surfaces.

Even if knowledge about surface reflectance and camera models is given, a potentially combinatorial problem remains in finding one or more scene reconstructions that are photo-consistent with the set of input images. There are two issues here: First, since there may be many photo-consistent reconstructions, which one or ones should be computed and how do they relate to the other legal interpretations? Second, because photo-consistency for a surface point associated with a given voxel in the scene requires knowledge of which images the voxel is visible in, i.e., for which images the voxel contains the closest surface along the visual ray from the camera. Efficient methods of performing this visibility test are essential to making this approach work.

Combining the ability to test a voxel's visibility with respect to the cameras, with a predicate that evaluates the voxel's photo-consistency with respect to the image irradiance values where the voxel is visible, is the basis for a family of methods that produce a reconstruction by classifying each voxel as either containing only scene points (labeled opaque) or not (labeled transparent). One advantage of this approach is that it makes no surface smoothness assumptions beyond that implicit in the resolution of an individual voxel, and therefore complex shapes can be reconstructed. Figure 16.2 shows an overview of the approach.

In order to avoid simultaneous recovery of surface reflectance (BRDF) and illumination, photo-consistency has been applied in the case where the surfaces are all assumed to be Lambertian. In this case, photo-consistency can be defined in a number of ways. The simplest is to project a voxel's centroid into each image and threshold the variance of the colors of the pixels at those points. A less noise-sensitive method uses the colors of all the pixels that a voxel's projection overlaps. This has been done using an F test, computed by the ratio of the variances of the voxel's pixels' colors and the colors of pixels associated with a known homogeneous surface, under the null hypothesis that the variances are equal [47]. A threshold on this ratio determines the photo-consistency of the voxel. Experiments using other definitions of photo-consistency have also been conducted [7, 14, 19, 29].

5. VOXEL VISIBILITY USING PLANE-SWEEP

Because of the complicated, non-local dependencies between scene surfaces that determine a voxel's visibility, methods that simplify visibility testing are important. One such simplification is made possible if there exists a topological sort of all voxels using the partial ordering relation "voxel x can be occluded by voxel y from any one of the camera viewpoints." That is, if the line segment connecting the center of voxel x and the optical center of any one of the cameras intersects voxel y, then x occurs after y in the ordering.[1] When such an ordering exists, we can traverse the voxels in this order and guarantee that when a voxel is visited, all possible occluders of the voxel with respect to every camera have previously been visited. Thus, visibility testing is dependent only on the labels of voxels visited previously, enabling a one-pass algorithm.

With a single camera or a set of cameras all lying on the same side of a plane, voxels can be ordered by increasing distance from the plane, resulting in a sequence of voxel planes at increasing distance from all the cameras [30]. This is the basis for Collins's plane-sweep algorithm [12], which counts the number of image features that back-project to each voxel, marking voxels as scene

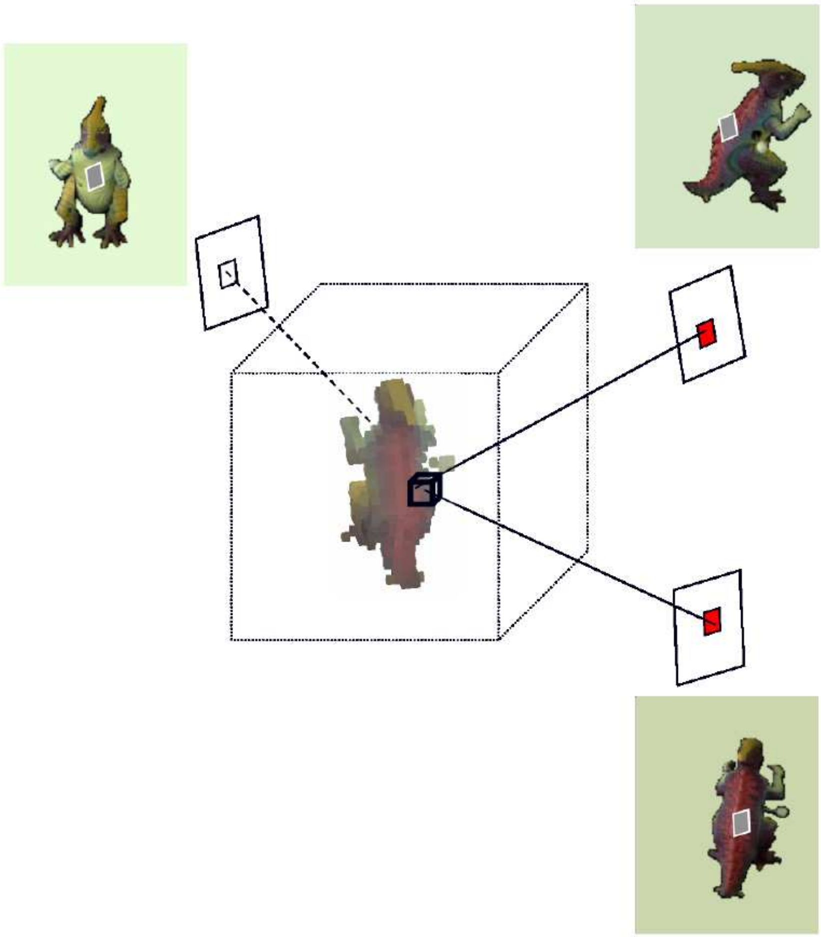


Figure 16.2 Overview of the approach to scene reconstruction based on photo-consistency. A voxel is included in the scene model if for those images in which the voxel is visible, here the two on the right, the pixels' colors are photo-consistent.

feature points when the count exceeds a non-coincidence threshold. While using an ordered traversal of scene space, the method does not explicitly handle inter-feature occlusion. Others have used a similar feature-based plane-sweep algorithm for detecting planar regions [3, 13].

Seitz and Dyer [47] showed that topological sorting of voxels is also possible for any camera configuration in which the scene volume lies outside the convex hull of the cameras' optical centers. Instead of planes at increasing distances from the cameras, this results in a partition of scene space into an expanding front of layers at increasing distances from the cameras' convex hull.

6. VOXEL COLORING

For camera configurations that allow voxels to be topologically sorted, one pass through the voxels is sufficient to create a photo-consistent reconstruction because the visibility of a voxel is completely determined when it is visited. Each voxel can be labeled after testing the photo-consistency of all of the visible pixels it projects to. Assuming that a binary labeling is desired, specifying whether each voxel is opaque (i.e., is part of an object) or transparent (i.e., contains only free space), an implementation either can assume that scene space is initially transparent and label a voxel opaque if it passes the photo-consistency test [47], or can assume that scene space is initially solid and label a voxel transparent if it fails the photo-consistency test [29]. The first approach is like clay modeling, whereas the second is akin to sculpture. Note that the first version results in a set of opaque voxels that define the scene surfaces (though it could be modified to also mark interior voxels as opaque), whereas the second results in a volumetric description of the scene. The first approach was used in a one-pass algorithm called Voxel Coloring [47]. Figure 16.3 shows results using a set of views of a rose.

Once a volumetric model is built, modifying the scene is possible using image editing operations applied to any one of the input images. Because of the known relationships between the pixels in every image and the scene voxels, changes made in one image are easily propagated to the model as well as to each of the other views [48].

Up to the limits of the photo-consistency test and the voxel resolution, the result of this algorithm is a scene model that is consistent with all of the pixels in all of the input images. But since in general there may be many scenes that are consistent with a set of input images, what relation does the one produced by the algorithm have with all the other possible solutions? Kutulakos and Seitz [29] showed that because the space-carving procedure removes opaque voxels until every border voxel[2] is photo-consistent, the closest photo-consistent voxel along each visual ray is guaranteed to be on the surface of the final shape. Thus the reconstruction found by the algorithm is maximal in that it is the union of all other photo-consistent scene reconstructions. For this reason it is called the *photo hull*.

(a) (b) (c)

Figure 16.3 Reconstruction using the Voxel Coloring algorithm. (a) One of 21 input images taken from above a rose. (b) Image rendered from the volumetric model for the same view as (a). (c) A novel view rendered from the model. (From [47], with permission.)

7. SPACE CARVING

For general camera configurations, a one-pass algorithm is not possible because voxels cannot be topologically sorted with respect to all viewpoints. Instead, a multi-pass procedure can be used that makes multiple plane-sweep passes, evaluating each voxel in the current plane of voxels using the subset of cameras and voxels that are in front of that plane. In other words, each pass sweeps a plane at a different orientation through scene space, and at each voxel photo-consistency is tested using only the cameras and other voxels that are on one side of the plane that contains the given voxel. As each pass may change the labels of voxels that affect the visibility, and therefore the labeling, of other voxels, multiple plane-sweep passes are necessary until no change occurs.

Kutulakos and Seitz [29] proved that using the space carving labeling strategy with an iterative, multi-plane-sweep traversal guarantees that border voxels will be successively removed until no non-photo-consistent voxel exists and the remaining shape is the photo hull. In order to limit the number of plane orientations that are swept, a fixed set of planes, say those that are parallel to the sides of the cube defining the original scene volume, can be used, resulting in an approximation of the photo hull. To guarantee that the true photo hull is constructed using a fixed set of sweep planes, at the end of each iteration of sweeps, the visibility of each voxel can be checked with respect to every camera, the photo-consistency test can be applied and additional border voxels removed. This method is called the Space Carving algorithm [29]. Figure 16.4 shows the result of Space Carving using six plane-sweeps at each iteration. If it is known that the scene contains a single connected shape, this method can include an additional test that prevents the carving process from disconnecting

(a) (b) (c)

Figure 16.4 Reconstruction using the Space Carving algorithm. (a) One of 100 input images of a hand. (b–c) Two novel views. (From [29] with permission.)

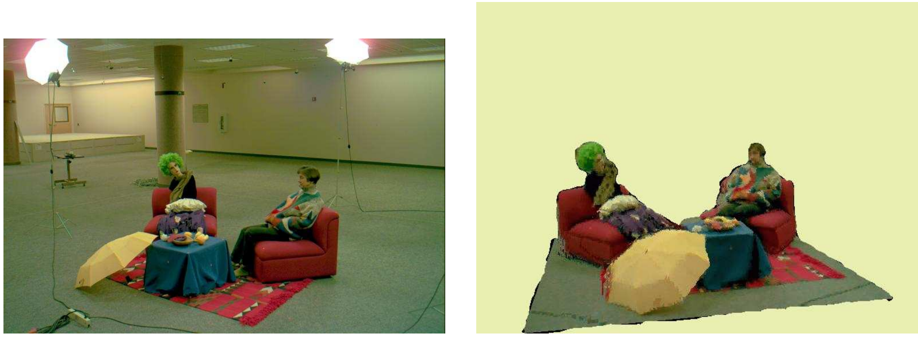
the shape, by ensuring that a border voxel is not removed if it would locally disconnect the opaque voxels in its neighborhood.

An alternative to the multi-sweep approach used by Space Carving is to simply iterate over every border voxel, successively removing those that are not photo-consistent, until no change occurs in a complete pass over the surface. The result is also the photo hull. When a voxel is visited its visibility in every image must be determined, and without the plane-sweep constraint this test is more complicated. To improve its efficiency, this Generalized Voxel Coloring algorithm [15] maintains a data structure that indicates for each pixel the address of the closest opaque voxel along the pixel's visual ray. This data structure can be updated less frequently than after each single voxel is carved, at the possible cost of additional iterations. Figure 16.5 shows a scene reconstructed using this method.

Faugeras and Keriven [20] developed a related approach based on the level set method [49]. An initial scene-bounding surface represented in voxel space evolves towards the objects in the scene until a matching criterion based on normalized cross-correlation is minimized. The current surface is used to determine the visibility of voxels in the images, the direction of evolution of the surface, and the speed of evolution. The method assumes Lambertian objects. The evolving surface is smooth, and is biased towards minimizing the total area of the objects.

8. BETTER RECONSTRUCTIONS

While the results of the algorithms described in the previous two sections show impressive photorealism both in reproducing input images and in synthesizing novel views, the accuracy of the reconstructed scene models can be improved in a number of ways.



(a) (b)

Figure 16.5 Reconstruction using the Generalized Voxel Coloring algorithm. (a) One of 24 input images. (b) A novel view. (From [51] with permission.)

The relation between the photo hull and the true scene depends on many factors. First, it depends on the degree to which the surface reflectance function used in the photo-consistency test accurately models the true surface reflectance. Second, errors in estimating surface orientation and illumination at a discrete voxel may cause errors in the photo-consistency test, depending on the nature of the reflectance function. Third, discrete voxels can cause aliasing artifacts. Fourth, the reliance on a threshold to determine photo-consistency leads to voxel classification errors.

Photo-consistency evaluation errors have two effects on the reconstructed shape: holes or false concavities where the threshold is too high, causing carving to go too far, and fattening or false convexities where the threshold is too low, causing carving to stop too soon. Experiments have been performed [7] to investigate reconstruction errors as a function of the photo-consistency threshold. The hole problem can be partially handled by hole filling methods [16]. Another strategy is to post-process the photo hull to obtain a better final scene model. Slabaugh et al. [50] took this approach by formulating the refinement process as an optimization problem, which iteratively added or removed border voxels until the sum of squared differences between the input images and the scene model rendered in each camera was minimized. Simulated annealing and greedy methods were used. This process can be thought of as a way to spatially vary the threshold used to decide on photo-consistency.

The fattening problem is particularly pronounced in regions of low color variation when two different surface points have similar radiance, causing false-positive photo-consistency at voxels that are in front of the true surface. Figure 16.6 shows this effect for synthetic images in which an intensity gradient across the image is varied.

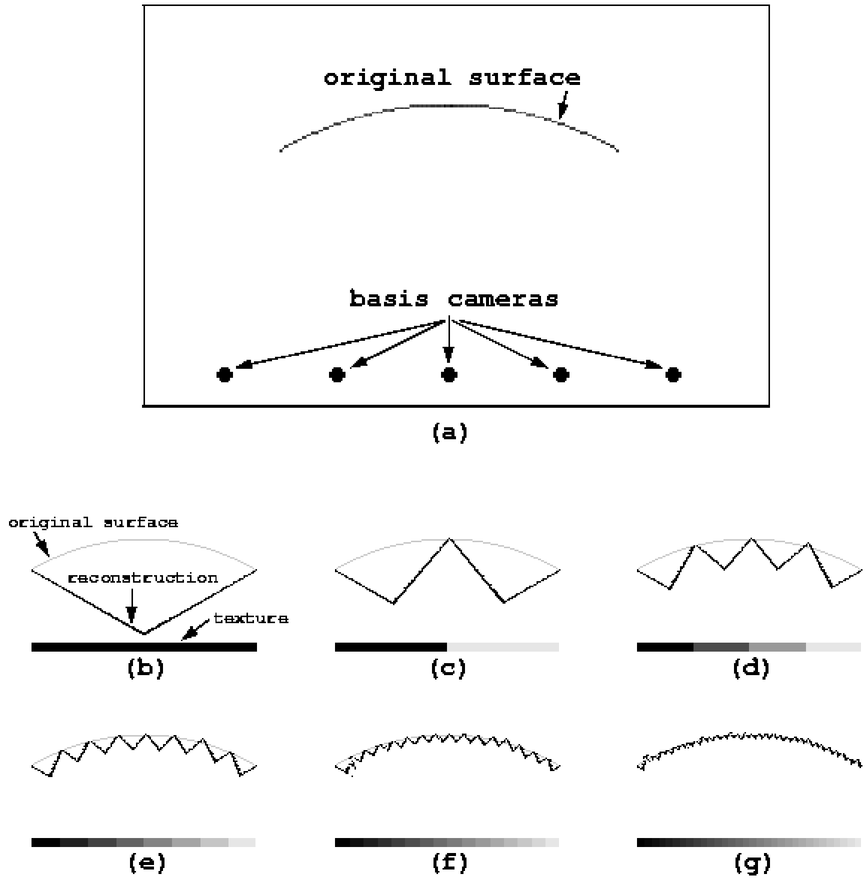


Figure 16.6 Effects of texture density on reconstruction. (a) A concave scene surface and the positions of five input cameras. (b) Reconstruction when the surface has constant radiance gives the same result as shape-from-silhouettes. (c–g) Successively better reconstructions result when the surface is textured with an intensity gradient that doubles in frequency with each image. (From [47] with permission.)

Scene model reconstruction accuracy can also be improved by minimizing the distance between the projected silhouettes and the actual silhouettes [14]. Another way of dealing with this aliasing issue, which is especially problematic when the voxel resolution is coarse, is to assign a degree-of-opacity value to each voxel. A value of 0 indicates that the voxel is completely transparent, and value of 1 indicates that it is entirely opaque. Values between 0 and 1 indicate a mixed voxel, i.e., a voxel that intersects both scene points and free space, so is partially opaque and partially transparent. Using opacity values is especially useful for rendering because the synthesized views will be anti-aliased [17].

Szeliski and Golland [58] used a multi-step process to initially estimate and then refine decisions about voxel visibility, opacity, and color. A non-linear optimization method was used in the final step to compute voxel opacities, which is especially difficult for mixed voxels. Energy minimization used the difference between the projected model and the images, a smoothness constraint on the colors and opacities, and a prior distribution on the opacities.

9. EXTENSIONS

In this section we mention some extensions and improvements that have been investigated.

Calibration Errors. The methods we have described for building volumetric scene models have assumed accurately calibrated cameras, so the projections of a voxel determine corresponding pixels in the images. In practice, of course, calibration may be inaccurate. To handle this situation, Kutulakos [28] defined an Approximate Space Carving algorithm that constructs an approximation of the photo hull called an r-consistent volume. R-consistency is defined by weakening the definition of photo-consistency so that if a voxel projects to pixel x in one image and to pixel y in a second image, there is a pixel within distance r of y that has the same color as pixel x. Varying r generates a nested family of approximations, with the photo hull being the tightest approximation to the true scene. This property makes it possible not only to cope with calibration errors, but to build a multi-scale hierarchy of fine-to-coarse volumetric descriptions.

Large-Scale Environments. Scaling up volumetric modeling methods for use with large-scale environments is difficult because if the voxel resolution is set fine enough to encode the smallest important scene details, a uniform tessellation will require an unmanageable number of voxels. One way to cope with this problem is to use an octree representation, using coarse-resolution voxels to represent large areas of free space and the interiors of large objects, and fine-resolution voxels in areas of fine scene detail [45]. Another approach is to increase the size of the voxels with distance from the center of the environment, with unbounded voxels at the borders of the environment [52].

Partly Transparent Scenes. Some recent work has considered scenes containing partly transparent objects. DeBonet and Viola [18] used an optimization method that, like that of Szeliski and Golland [58], computes a real-valued opacity value as well as a color at each voxel. Their method, called Roxels, searches for a linear combination of the colors at the voxels along each visual ray through the entire voxel space that, when projected and composited into the cameras, minimizes the errors of the input images. Dachille et al. [17] recovered color and opacity values at each voxel using the SART reconstruction technique [2]. Other tomographic techniques may also be applicable to this problem [21].

Dynamic Scenes. While dynamic scenes can be represented as sequences of static volumetric descriptions, it is preferable to exploit temporal coherence. One simple way is to initialize the space carving process at each time step using a slightly fattened photo hull created for the previous time step, thus eliminating the need to carve many voxels at each step. This idea has been used in combination with an octree representation of voxel space [45]. Another approach is to build a 6D volumetric representation that links two 3D voxel descriptions at consecutive time steps [60]. Photo-consistency can be applied to 6D voxels, indicating whether or not two corresponding 3D voxels at successive time steps project to pixels of the same color in multiple views.

10. CONCLUSIONS

Significant steps have been taken toward the ability to construct 3D scene models that are both geometrically and photometrically accurate from an arbitrary set of images. While improvements in geometric accuracy, recovery and use of more realistic surface reflectance and illumination models, and better methods for large-scale, complex environments, are still needed, the results are increasingly photorealistic.

Acknowledgments

The support of the National Science Foundation under Grant No. IIS-9988426 is gratefully acknowledged.

Notes

1. A better definition: If y intersects the cone defined by a camera's visual rays that meet voxel x's front face, x occurs after y.
2. A border voxel is an opaque voxel that is adjacent to a transparent voxel.

References

[1] N. Ahuja and J. Veenstra. Generating octrees from object silhouettes in orthographic views. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11(2):137–149, 1989.

[2] A. H. Andersen and A. C. Kak. Simultaneous Algebraic Reconstruction Technique (SART): A superior implementation of the ART algorithm. *Ultrasonic Imaging*, 6:81–94, 1984.

[3] C. Baillard and A. Zisserman. A plane-sweep strategy for the 3D reconstruction of buildings from multiple images. In *Proc. 19th ISPRS Congress*

and Exhibition, 2000.

[4] S. Baker, R. Szeliski, and P. Anandan. A layered approach to stereo reconstruction. In *Proc. Computer Vision and Pattern Recognition Conf.*, pages 434–441, 1998.

[5] B. G. Baumgart. Geometric modeling for computer vision. Technical Report Artificial Intelligence Laboratory Memo AIM-249, Stanford University, 1974.

[6] P. A. Beardsley, P. H. S. Torr, and A. Zisserman. 3D model aquisition from extended image sequence. In B. Buxton and R. Cipolla, editors, *Computer Vision – ECCV '96 (Proc. 4th European Conf. on Computer Vision, Volume II)*, volume 1065 of *Lecture Notes in Computer Science*, pages 683–695. Springer-Verlag, 1996.

[7] A. Broadhurst and R. Cipolla. A statistical consistency check for the space carving algorithm. In *Proc. 11th British Machine Vision Conf.*, pages 282–291, 2000.

[8] Q. Chen and G. Medioni. A volumetric stereo matching method: Application to image-based modeling. In *Proc. Computer Vision and Pattern Recognition Conf.*, volume 1, pages 29–34, 1999.

[9] G. K. M. Cheung, T. Kanade, J-Y. Bouguet, and M. Holler. A real time system for robust 3D voxel reconstruction of human motions. In *Proc. Computer Vision and Pattern Recognition Conf.*, volume 2, pages 714–720, 2000.

[10] C. H. Chien and J. K. Aggarwal. Volume surface octrees for the representation of 3D objects. *Computer Vision, Graphics and Image Processing*, 36:100–113, 1986.

[11] R. Cipolla and A. Blake. Surface shape from the deformation of apparent contours. *Int. J. Computer Vision*, 9(2):83–112, 1992.

[12] R. T. Collins. A space-sweep approach to true multi-image matching. In *Proc. Computer Vision and Pattern Recognition Conf.*, pages 358–363, 1996.

[13] S. Coorg and S. Teller. Extracting textured vertical facades from controlled close-range imagery. In *Proc. Computer Vision and Pattern Recognition Conf.*, volume 1, pages 625–632, 1999.

[14] G. Cross and A. Zisserman. Surface reconstruction from multiple views using apparent contours and surface texture. In A. Leonardis, F. Solina, and R. Bajcsy, editors, *Confluence of Computer Vision and Computer Graphics*, pages 25–47. Kluwer, 2000.

[15] W. B. Culbertson, T. Malzbender, and G. Slabaugh. Generalized voxel coloring. In B. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision*

Algorithms: Theory and Practice (Proc. Int. Workshop on Vision Algorithms), volume 1883 of *Lecture Notes in Computer Science*, pages 100–115. Springer-Verlag, 2000.

[16] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proc. SIGGRAPH 96*, pages 303–312, 1996.

[17] F. Dachille, K. Mueller, and A. Kaufman. Volumetric backprojection. In *Proc. Volume Visualization and Graphics Symposium*, pages 109–117, 2000.

[18] J. S. DeBonet and P. Viola. Roxels: Responsibility weighted 3D volume reconstruction. In *Proc. Seventh Int. Conf. on Computer Vision*, pages 418–425, 1999.

[19] P. Eisert, E. Steinbach, and B. Girod. Multi-hypothesis, volumetric reconstruction of 3-D objects from multiple calibrated camera views. In *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, pages 3509–3512, 1999.

[20] O. Faugeras and R. Keriven. Complete dense stereovision using level set methods. In H. Burkhardt and B. Neumann, editors, *Computer Vision – ECCV '98 (Proc. 5th European Conf. on Computer Vision, Volume I)*, volume 1406 of *Lecture Notes in Computer Science*, pages 379–393. Springer-Verlag, 1998.

[21] D. T. Gering and W. M. Wells III. Object modeling using tomography and photography. In *Proc. IEEE Workshop on Multi-View Modeling and Analysis of Visual Scenes*, pages 11–18, 1999.

[22] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. In *Proc. SIGGRAPH 96*, pages 43–54, 1996.

[23] T. H. Hong and M. Shneier. Describing a robot's workspace using a sequence of views from a moving camera. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 7:721–726, 1985.

[24] S. S. Intille and A. F. Bobick. Disparity-space images and large occlusion stereo. In J-O. Eklundh, editor, *Computer Vision – ECCV '94 (Proc. 3rd European Conf. on Computer Vision, Volume II)*, volume 801 of *Lecture Notes in Computer Science*, pages 179–186. Springer-Verlag, 1994.

[25] M. Kimura, H. Saito, and T. Kanade. 3D voxel construction based on epipolar geometry. In *Proc. Int. Conf. Image Processing*, pages 135–139, 1999.

[26] R. Koch, M. Pollefeys, and L. Van Gool. Multi viewpoint stereo from uncalibrated video sequences. In H. Burkhardt and B. Neumann, editors, *Computer Vision – ECCV '98 (Proc. 5th European Conf. Computer Vision,*

Volume I), volume 1406 of *Lecture Notes in Computer Science*, pages 55–71. Springer-Verlag, 1998.

[27] K. N. Kutulakos. Shape from the light field boundary. In *Proc. Computer Vision and Pattern Recognition Conf.*, pages 53–59, 1997.

[28] K. N. Kutulakos. Approximate N-view stereo. In D. Vernon, editor, *Computer Vision – ECCV 2000 (Proc. 6th European Conf. on Computer Vision, Part I)*, volume 1842 of *Lecture Notes in Computer Science*, pages 67–83. Springer-Verlag, 2000.

[29] K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. *Int. J. of Computer Vision*, 38(3):199–218, 2000.

[30] M. S. Langer and S. W. Zucker. Shape-from-shading on a cloudy day. *J. Opt. Soc. Am. A*, 11(2):467–478, 1994.

[31] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 16(2):150–162, 1994.

[32] A. Laurentini. How far 3D shapes can be understood from 2D silhouettes. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17(2):188–195, 1995.

[33] A. Laurentini. How many 2D silhouettes does it take to reconstruct a 3D object? *Computer Vision and Image Understanding*, 67(1):81–87, 1997.

[34] Y. G. Leclerc, Q-T. Luong, and P. Fua. Characterizing the performance of multiple-image point-correspondence algorithms using self-consistency. In B. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice (Proc. Int. Workshop on Vision Algorithms)*, volume 1883 of *Lecture Notes in Computer Science*, pages 100–115. Springer-Verlag, 2000.

[35] Y. G. Leclerc, Q-T. Luong, and P. Fua. Measuring the self-consistency of stereo algorithms. In D. Vernon, editor, *Computer Vision – ECCV 2000 (Proc. 6th European Conf. Computer Vision, Part II)*, volume 1842 of *Lecture Notes in Computer Science*, pages 282–298. Springer-Verlag, 2000.

[36] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *Proc. SIGGRAPH 87*, pages 163–169, 1987.

[37] W. N. Martin and J. K. Aggarwal. Volumetric description of objects from multiple views. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 5(2):150–158, 1983.

[38] W. Matusik, C. Buehler, R. Raskar, S. J. Gortler, and L. McMillan. Image-based visual hulls. In *Proc. SIGGRAPH 2000*, pages 369–374, 2000.

[39] M. Meissner, J. Huang, D. Bartz, K. Mueller, and R. Crawfis. A practical evaluation of popular volume rendering algorithms. In *Proc. Volume Visualization and Graphics Symposium*, pages 81–90, 2000.

[40] S. Moezzi, A. Katkere, D. Kuramura, and R. Jain. Reality modeling and visualization from multiple video sequences. *IEEE Computer Graphics and Applications*, 16(6):58–63, 1996.

[41] S. Moezzi, L-C. Tai, and P. Gerard. Virtual view generation for 3D digital video. *IEEE Multimedia*, 4(1):18–26, 1997.

[42] W. Niem. Error analysis for silhouette-based 3D shape estimation from multiple views. In *Proc. Int. Workshop on Synthetic-Natural Hybrid Coding and Three-Dimensional Imaging*, 1997.

[43] H. Noborio, S. Fukada, and S. Arimoto. Construction of the octree approximating three-dimensional objects by using multiple views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):769–782, 1988.

[44] M. Potmesil. Generating octree models of 3D objects from their silhouettes in a sequence of images. *Computer Vision, Graphics and Image Processing*, 40:1–20, 1987.

[45] A. C. Prock and C. R. Dyer. Towards real-time voxel coloring. In *Proc. 1998 Image Understanding Workshop*, pages 315–321, 1998.

[46] H. Saito and T. Kanade. Shape reconstruction in projective grid space from large number of images. In *Proc. Computer Vision and Pattern Recognition Conf.*, volume 2, pages 49–54, 1999.

[47] S. M. Seitz and C. R. Dyer. Photorealistic scene reconstruction by voxel coloring. *Int. J. of Computer Vision*, 35(2):151–173, 1999.

[48] S. M. Seitz and K. N. Kutulakos. Plenoptic image editing. In *Proc. Sixth Int. Conf. on Computer Vision*, pages 17–24, 1998.

[49] J. A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 2nd edition, 1999.

[50] G. Slabaugh, B. Culbertson, T. Malzbender, and R. Schafer. Improved voxel coloring via volumetric optimization. Technical Report 3, Center for Signal and Image Processing, Georgia Institute of Technology, 2000.

[51] G. Slabaugh, B. Culbertson, T. Malzbender, and R. Schafer. A survey of methods for volumetric scene reconstruction from photographs. Technical Report 1, Center for Signal and Image Processing, Georgia Institute of Technology, 2001.

[52] G. G. Slabaugh, T. Malzbender, and W. B. Culbertson. Volumetric warping for voxel coloring on an infinite domain. In M. Pollefeys, L. Van Gool, A. Fitzgibbon, and A. Zisserman, editors, *3D Structure from Multiple Images of Large-Scale Environments and Applications to Virtual and*

Augmented Reality (Proc. SMILE 2000), Lecture Notes in Computer Science, pages 41–50. Springer-Verlag, 2001.

[53] D. Snow, P. Viola, and R. Zabih. Exact voxel occupancy with graph cuts. In *Proc. Computer Vision and Pattern Recognition Conf.*, volume 1, pages 345–352, 2000.

[54] S. K. Srivastava and N. Ahuja. Octree generation from object silhouettes in perspective views. *Computer Vision, Graphics and Image Processing*, 49:68–84, 1990.

[55] R. Szeliski. Rapid octree construction from image sequences. *Computer Vision, Graphics and Image Processing: Image Understanding*, 58(1):23–32, 1993.

[56] R. Szeliski. Prediction error as a quality metric for motion and stereo. In *Proc. Seventh Int. Conf. on Computer Vision*, pages 781–788, 1999.

[57] R. Szeliski. Stereo algorithms and representations for image-based rendering. In *Proc. 10th British Machine Vision Conf.*, pages 314–328, 1999.

[58] R. Szeliski and P. Golland. Stereo matching with transparency and matting. *Int. J. of Computer Vision*, 32(1):45–61, 1999.

[59] R. Vaillant and O.D. Faugeras. Using extremal boundaries for 3D object modelling. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14:157–173, 1992.

[60] S. Vedula, S. Baker, S. Seitz, and T. Kanade. Shape and motion carving in 6D. In *Proc. Computer Vision and Pattern Recognition Conf.*, volume 2, pages 592–598, 2000.