To appear in Int. Journal of Computer Vision, volume 35, number 2, 1999.

Photorealistic Scene Reconstruction by Voxel Coloring *

Steven M. Seitz
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Charles R. Dyer
Department of Computer Sciences
University of Wisconsin–Madison
Madison, WI 53706

Abstract

A novel scene reconstruction technique is presented, different from previous approaches in its ability to cope with large changes in visibility and its modeling of intrinsic scene color and texture information. The method avoids image correspondence problems by working in a discretized scene space whose voxels are traversed in a fixed visibility ordering. This strategy takes full account of occlusions and allows the input cameras to be far apart and widely distributed about the environment. The algorithm identifies a special set of invariant voxels which together form a spatial and photometric reconstruction of the scene, fully consistent with the input images. The approach is evaluated with images from both inward-facing and outward-facing cameras.

1 Introduction

The problem of acquiring photorealistic models of an environment from a set of input images has sparked recent interest in the computer vision community [47, 12, 31, 3, 35, 33] as a result of new graphics-oriented applications, like telepresence and virtual walkthroughs, that require the visualization of real objects and scenes. The central task is to synthesize images from new camera viewpoints of an observed scene. An ideal solution yields an image, for every camera viewpoint, that is *photorealistic*, i.e., is indistinguishable from what a real observer would see from the same viewpoint.

In this paper, we seek to define what photorealism means with respect to scene reconstruction techniques, and to present a practical algorithm for computing photorealistic scene reconstructions from images. While our focus is 3D reconstruction, our approach is equally useful as a method for obtaining dense pixel correspondence for use in image-based view synthesis techniques [8, 28, 30, 40, 5, 1][1]. As a step towards this goal, we propose two criteria that a *photorealistic* 3D reconstruction technique should meet and describe an algorithm for generating such a reconstruction from a set of photographs:

- Photo Integrity: The 3D reconstruction should reproduce the input images when projected to the input viewpoints, preserving color, texture and pixel resolution.

*This research has been supported in part by the National Science Foundation under Grant No. IRI-9530985, and by the Defense Advanced Research Projects Agency and Rome Laboratory, USAF, under Agreement No. F30602-97-1-0138.

[1] Numerous authors have argued that correspondence information is sufficient to synthesize new views of a scene, i.e., explicit 3D reconstruction is not required. However, the problem of obtaining correspondence that is sufficiently accurate for this task remains an area of active research.

- Broad Viewpoint Coverage: To enable accurate reprojections over a wide range of target viewpoints, the reconstruction should integrate numerous and widely-distributed input images.

The photorealistic scene reconstruction problem, as presently formulated, raises a number of unique challenges that push the limits of existing reconstruction techniques. Photo integrity requires that the reconstruction be dense and sufficiently accurate to reproduce the original images. This criterion poses a problem for existing feature- and contour-based techniques that do not provide dense shape estimates. While these techniques can produce texture-mapped models [47, 31, 3], accuracy is ensured only in places where features have been detected. The second criterion means that the input views may be far apart and contain significant occlusions. While some stereo methods [4, 17, 32] can cope with limited occlusions, handling visibility changes of greater magnitude appears to be beyond the state of the art.

Instead of approaching this problem as one of shape reconstruction, we formulate a *color reconstruction* problem, in which the goal is an assignment of colors (radiances) to points in an (unknown) approximately Lambertian scene. As a solution, we present a *voxel coloring* technique that traverses a discretized 3D space in a generalized "depth-order" to identify voxels that have a unique color, constant across all possible interpretations of the scene. This approach has several advantages over existing stereo and structure-from-motion approaches to pixel correspondence and scene reconstruction. First, occlusions are explicitly modeled and accounted for. Second, the cameras can be positioned far apart without degrading accuracy or run-time. Third, the technique integrates numerous images to yield dense reconstructions. Fourth, and most important, the method is shown to produce high quality synthetic views for a wide range of real input sequences and target viewpoints.

The remainder of this paper is structured as follows. Section 1.1 discusses related work on scene-space reconstruction techniques and summarizes capabilities and limitations of existing solutions. Section 2 introduces the voxel coloring paradigm as a means for determining correspondence and visibility in scene space. Section 3 defines the notion of *color invariants* and describes their importance for voxel coloring. Section 4 presents an efficient algorithm for computing a voxel coloring using a *layering* strategy, and Section 5 presents experimental results from applying the algorithm to real and synthetic input images.

1.1 Related Work

The voxel coloring algorithm presented in this paper works by discretizing scene space into a set of voxels that is traversed and colored in a special order. In this respect, the method is similar to Collins' *Space-Sweep* approach [9, 10] which performs an analogous scene traversal. In the latter approach, a plane is swept through the scene volume and votes are accumulated for points on the plane that project to edge features in the images. Scene features are identified by modeling the statistical likelihood of accidental accumulation and thresholding the votes to achieve a desired false positive rate. This approach is useful in the case of limited occlusions, but does not provide a general solution to the visibility problem. In addition, the Space-Sweep approach generates shape estimates only where edges were detected, i.e., it does not produce a dense reconstruction.

Seitz and Dyer [39] described a similar edge-based voting technique that used linear subspace intersections instead of a plane sweep to obtain feature correspondences. In this approach, each point or line feature in an image "votes" for the scene subspace that projects to that feature. Votes are accumulated when two or more subspaces intersect, indicating the possible

presence of a point or line feature in the scene. A restriction of this technique is that it detects correspondences only for features that appear in all input images.

Katayama et al. [21] described a related method in which images are matched by detecting lines through slices of an epipolar volume [6, 2], noting that occlusions can be correctly modeled by labeling lines in order of increasing slope. Our voxel traversal strategy yields a similar scene-space ordering but is not restricted to linear camera paths. However, their algorithm uses a reference image, thereby ignoring points that are occluded in the reference image but visible in other input images.

Narayanan et al. [33, 19] built a dome of cameras in order to reconstruct 3D models from time-varying imagery. They addressed the problem of occlusions by considering local clusters of cameras and computing stereo reconstructions from each cluster. The resulting partial models are subsequently merged [33, 11] to form a more complete reconstruction. This approach has proven to be most successful when the partial reconstructions to be merged are individually quite accurate, e.g., as obtainable by high-end laser range scanners. While Narayanan et al. reported very good results, more research is needed to assess the applicability of these methods for noisier stereo-derived models. A disadvantage of the method is that the original images are not used in the merging process so it is difficult to assess the photo integrity of the reconstructions.

Zitnick and Webb [50] described a scene space stereo technique that detects and reconstructs scene regions that appear unoccluded in a set of input images. They noted that the correspondence problem is ill-posed in the presence of occlusion, but can be solved when occlusion does not occur. By posing the problem as one of 3D surface extraction, they described how unoccluded regions may be identified and used to solve the correspondence problem for these regions.

Fua and Leclerc introduced a mesh-based stereo method [16] that evolves an initial mesh to be projectively consistent with a set of calibrated input images, using both shape-from-shading and stereo cues. Their consistency metric is similar to that which is used in this paper, except for their inclusion of a smoothness term to minimize deviations of the mesh from a plane. In contrast to the voxel coloring technique that we present in this paper, the mesh-based approach requires a good initial guess to converge properly, i.e., the mesh vertices should project to within a few pixels of their true locations[16], and the topology must be known a priori. These limitations are overcome using the voxel-based formulation presented in this paper.

Subsequent to the first publication of the voxel coloring technique [41], other promising multi-image 3D reconstruction techniques have recently emerged [46, 36, 13, 24]. Of these, most closely related are the level-set approach of Faugeras and Keriven [13] and the *Space Carving* approach of Kutulakos and Seitz [24]. The former approach formulates the 3D reconstruction procedure as a level-set evolution problem [43] in which a system of partial differential equations is iteratively solved on a dense voxel grid. In the Space Carving approach [24], a solid volume of voxels is progressively carved until convergence to a consistent scene reconstruction. Both of these methods provide many of the same advanges of voxel coloring and remove the constraint on camera geometry used in this paper. The price, however, is a significant penalty in run-time and ease of implementation. Unlike [13, 24], the voxel coloring algorithm operates in a single pass through the scene volume and has run-time complexity that is *independent* of the complexity of the scene being reconstructed. These properties enable reconstructions at near-real-time rates and are not shared by the level-set and Space Carving approaches.

Also related are recently developed panoramic stereo [30, 20] algorithms that avoid field of view problems by matching 360 panoramic images directly. Panoramic reconstructions can also be achieved using our approach, but without the need to build panoramic images (see Figs. 5(b) and 13).

Our objective in this paper is to synthesize new views of a scene from a set of input views that are widely distributed about the environment. Existing techniques are generally not well-suited for providing the correspondence information sufficient for this purpose, due to problems with occlusion, camera separation, or concavities. In particular, these approaches do not guarantee *consistent* reconstructions when occlusion is present, even under idealized conditions.

For instance, suppose we had two or more views $V_0, \ldots, V_n$ of a perfect Lambertian scene $\mathcal{S}$ under constant illumination, in which all internal and external camera parameters were precisely known. Furthermore, suppose all sources of error could be ignored, including those due to camera calibration, image quantization, and noise. While we could not hope to recover the true scene due to the aperture problem [34], we might reasonably expect to compute a *consistent* reconstruction, i.e., one corresponding to some scene $\mathcal{S}'$ that appears identical to $\mathcal{S}$ from the input viewpoints $V_0, \ldots, V_n$. However, existing algorithms do not guarantee consistent reconstructions when occlusion is present.

We believe that this shortcoming is due to the difficulty of reasoning about occlusion in image space, and instead advocate a scene space formulation for determining correspondence information. The advantages of this approach are two-fold: (1) it provides a framework for representing and analyzing the space of consistent scenes, and (2) the physical relations giving rise to occlusion are easily apparent and formalized. Consequently, scene space algorithms can be devised to take occlusions explicitly into account, and hence provide guarantees on the consistency of reconstructions.

2 The Voxel Coloring Problem

In this section we introduce a new scene space framework for scene reconstruction from multiple basis (input) views $V_0, \ldots, V_n$. In contrast to most existing approaches that solve for pixel correspondence as an initial step, we instead directly reconstruct a colored, voxel-based 3D scene that is consistent with all of the basis views. This scene can either be reprojected to synthesize new views as in [41], or used to compute correspondence maps for image-warping methods such as [8, 28, 30, 40, 38]. The approach has the unique feature that it ensures a consistent scene and set of image correspondence maps in the presence of arbitrary changes in scene visibility.

The voxel coloring problem is to assign colors (radiances) to voxels (points) in a 3D volume so as to achieve consistency with a set of basis images, as shown in Fig. 1. That is, rendering the colored voxels from each basis viewpoint should reproduce the original image as closely as possible. More formally, a 3D scene $\mathcal{S}$ is represented as a set of opaque Lambertian voxels (volume elements), each of which occupies a finite homogeneous scene volume centered at a point $\mathbf{V} \in \mathcal{S}$, and has an isotropic radiance $color(\mathcal{V}, \mathcal{S})$. We assume that the scene is entirely contained within a known, finite bounding volume. The set of all voxels in the bounding volume is referred to as the *voxel space* and denoted with the symbol $\mathcal{V}$. An image is specified by the set $\mathcal{I}$ of all its pixels, each centered at a point $\mathbf{p} \in \mathcal{I}$, and having irradiance $color(\mathbf{p}, \mathcal{S})$. For now, assume that pixels are infinitesimally small.

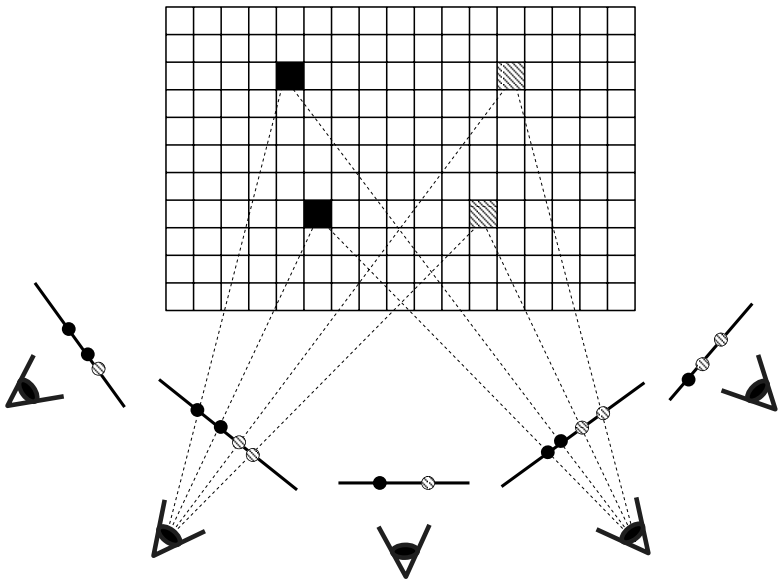


Figure 1: Voxel Coloring. Given a set of basis images and a grid of voxels, we wish to assign color values to voxels in a way that is consistent with all of the images.

Given an image pixel $\mathbf{p} \in \mathcal{I}$ and scene $\mathcal{S}$, we refer to the voxel $\mathbf{V} \in \mathcal{S}$ that is visible in $\mathcal{I}$ and projects to $\mathbf{p}$ by $\mathbf{V} = \mathcal{S}(\mathbf{p})$. A scene $\mathcal{S}$ is said to be *complete* with respect to a set of images if, for every image $\mathcal{I}$ and every pixel $\mathbf{p} \in \mathcal{I}$, there exists a voxel $\mathbf{V} \in \mathcal{S}$ such that $\mathbf{V} = \mathcal{S}(\mathbf{p})$. A complete scene is said to be *consistent* with a set of images if, for every image $\mathcal{I}$ and every pixel $\mathbf{p} \in \mathcal{I}$,

$$color(\mathbf{p}, \mathcal{I}) = color(\mathcal{S}(\mathbf{p}), \mathcal{S}) \tag{1}$$

We use the symbol $\aleph$ to denote the set of all consistent scenes. We may now define the voxel coloring problem formally:

> **Voxel Coloring Problem:** Given a set of basis images $\mathcal{I}_0, \ldots, \mathcal{I}_n$ of a static Lambertian scene and a voxel space $\mathcal{V}$, determine a subset $\mathcal{S} \subset \mathcal{V}$ and a coloring $color(\mathbf{V}, \mathcal{S})$, such that $\mathcal{S} \in \aleph$.

In order to solve this problem we must consider the following two issues:

- **Uniqueness**: Multiple voxel colorings may be consistent with a given set of images. How can the problem be well-defined?
- **Computation**: How can a voxel coloring be computed from a set of input images without combinatorial search?

Observe that a consistent voxel coloring *exists*, corresponding to the set of points and colors on surfaces of the true Lambertian scene[2]. Rarely, however, is the voxel coloring *unique*, given that a set of images can be consistent with more than one 3D scene. Determining a scene's spatial occupancy, i.e., $\mathcal{S}$, is therefore an ill-posed task because a voxel contained in one consistent scene may not be contained in another (see Fig. 2). Furthermore, a voxel may be contained in two consistent

[2]This argument holds only in the limit, when voxels are infinitesimally small, or else when the true scene is *sampled*, i.e., representable as a finite collection of axis-aligned cubes.

scenes, but have different colors in each (see Fig. 3). Consequently, additional constraints are needed in order to make the problem well-posed.

Computing voxel colorings poses another challenge. Observe that the underlying space is combinatorial: an $N \times N \times N$ grid of voxels, each with M possible color assignments yields 2^{N^3} possible scenes and M^{N^3} possible color assignments. Clearly, a bruce-force search through this space is not feasible.

3 Color Invariants

Given a multiplicity of solutions to the voxel coloring problem, the only way to recover intrinsic scene information is through *invariants*—properties that are satisfied by *every* consistent scene. For instance, consider the set of voxels that are contained in every consistent scene. Laurentini [26] described how these invariants, called *hard points*, could be recovered by volume intersection from silhouette images. Hard points provide absolute information about the true scene but are relatively rare; some images may yield none (see, for example, Fig. 2). In this section we describe a more frequently occurring type of invariant relating to color rather than shape.

> A voxel $\mathbf{V}$ is a **color invariant** with respect to a set of images if: (1) $\mathbf{V}$ is contained in a scene consistent with the images, and (2) for every pair of consistent scenes $\mathcal{S}$ and $\mathcal{S}'$, $\mathbf{V} \in \mathcal{S} \cap \mathcal{S}'$ implies $color(\mathbf{V}, \mathcal{S}) = color(\mathbf{V}, \mathcal{S}')$.

Unlike shape invariance, color invariance does not require that a point be contained in every consistent scene. As a result, color invariants are more prevalent than hard points. In particular, it will be shown that the union of all color invariants itself yields a consistent scene, i.e., a complete voxel coloring, as depicted in Fig. 4. Therefore, the voxel coloring problem can be reformulated as a well-posed problem by solving for the consistent scene corresponding to the set of color invariants. In order to make the problem *tractable*, however, additional constraints are needed.

3.1 The Ordinal Visibility Constraint

Note that color invariants are defined with respect to the set $\aleph$ of all consistent scenes—a combinatorial space. Clearly, an explicit search through this space is not computationally feasible. In order to make the problem tractable, we introduce a novel geometric constraint on camera placement relative to the scene that simplifies the analysis. This *ordinal visibility constraint* enables the identification of the set of color invariants as a limit point of $\aleph$. As such, they can be computed directly, via a single pass though the voxel space.

Let $\mathbf{P}$ and $\mathbf{Q}$ be scene points and $\mathcal{I}$ be an image from a camera centered at $\mathbf{C}$. We say $\mathbf{P}$ *occludes* $\mathbf{Q}$ if $\mathbf{P}$ lies on the line segment $\overline{\mathbf{CQ}}$. We require that the input cameras be positioned so as to satisfy the following constraint:

> **Ordinal visibility constraint:** There exists a real non-negative function $\mathcal{D} : \Re^3 \Rightarrow \Re$ such that for all scene points $\mathbf{P}$ and $\mathbf{Q}$, and input images $\mathcal{I}$, $\mathbf{P}$ occludes $\mathbf{Q}$ in $\mathcal{I}$ only if $\mathcal{D}(\mathbf{P}) < \mathcal{D}(\mathbf{Q})$.

We call such a function $\mathcal{D}$ *occlusion-compatible*. For some camera configurations, it is not possible to define an occlusion-compatible function. However, an occlusion-compatible function *does* exist for a broad range of practical configurations. For

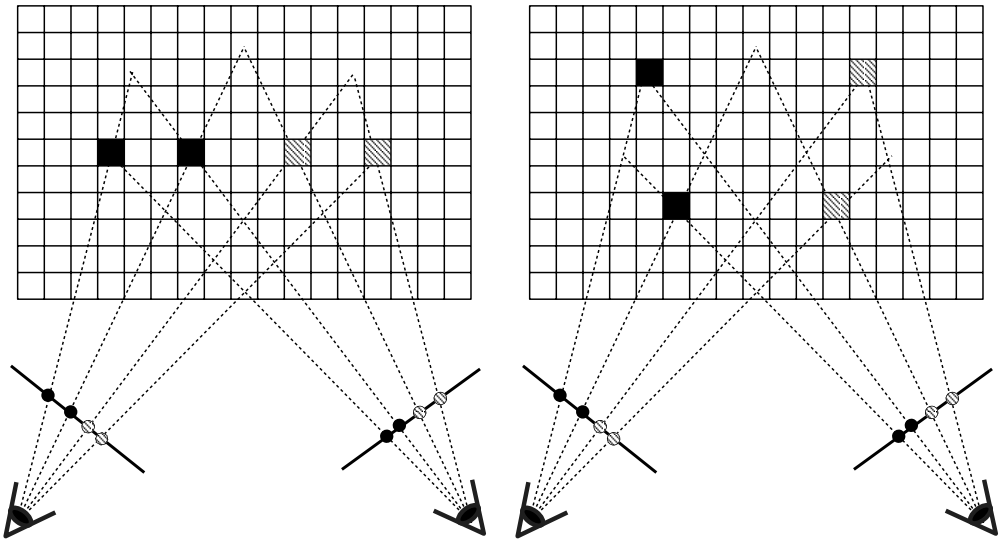


Figure 2: Spatial Ambiguity. Both voxel colorings appear identical from these two viewpoints, despite having no colored voxels in common.

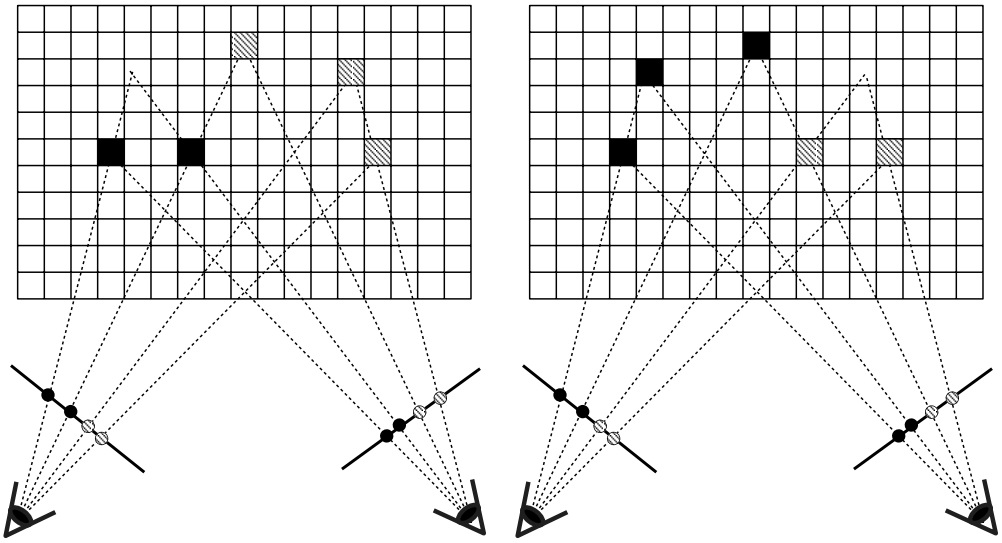


Figure 3: Color Ambiguity. Both voxel colorings appear identical from these two viewpoints. However, note the presence of a voxel (second row, center) that has a different color assignment in the two scenes.

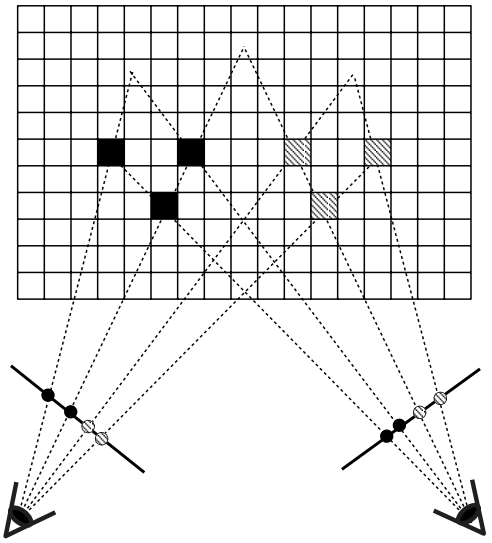
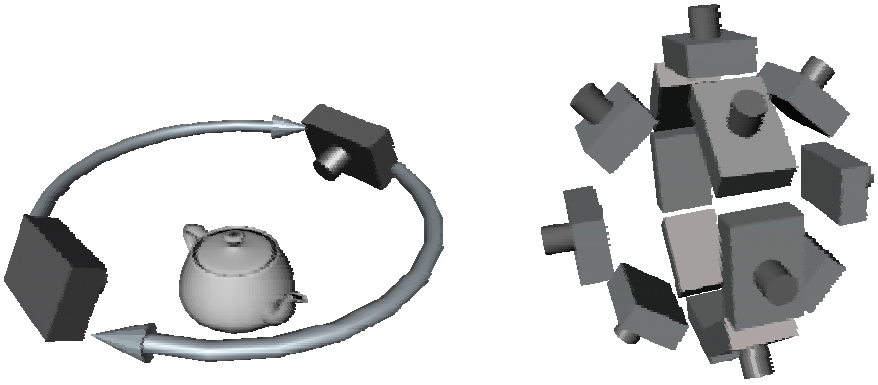


Figure 4: Color Invariants. Each of these six voxels has the same color in every consistent scene in which it is contained. The collection of all such *color invariants* forms a consistent voxel coloring denoted $\overline{\mathcal{S}}$, as depicted above. Note that the voxel with two color assignments in Fig. 3 is not contained in $\overline{\mathcal{S}}$.



(a) **(b)**

Figure 5: Compatible Camera Configurations. Both of the following camera configurations satisfy the ordinal visibility constraint: (a) an overhead inward-facing camera moved 360 degrees around an object, and (b) a rig of outward-facing cameras distributed around a sphere.

instance, suppose the cameras are distributed on a plane and the scene is entirely below that plane, as shown in Fig. 5(a). For every such viewpoint, the relative visibility of any two scene points depends entirely on which point is closer to the plane, so we may define $\mathcal{D}$ to be distance to the plane. More generally, the ordinal visibility constraint is satisfied whenever **no scene point is contained within the convex hull $\mathcal{C}$ of the camera centers**. Here we use the occlusion-compatible function $\mathcal{D}_{\mathcal{C}}(\mathbf{P})$, defined to be the Euclidean distance from $\mathbf{P}$ to $\mathcal{C}$. For convenience, $\mathcal{C}$ is referred to as the *camera volume*. Fig. 5 shows two useful camera configurations that satisfy this constraint. Fig. 5(a) depicts an inward-facing overhead camera rotating 360° around an object. Ordinal visibility is satisfied provided the camera is positioned slightly above the object. The constraint also enables "panoramic" configurations of outward-facing cameras, as in Fig. 5(b).

3.2 Properties of Color Invariants

To establish that color invariants exist, let $\mathcal{I}_0, \ldots, \mathcal{I}_n$ be a set of images for which the ordinal visibility constraint is satisfied. For a given image point $\mathbf{p} \in \mathcal{I}_j$ define $\mathbf{V}_\mathbf{p}$ to be the voxel in $\{\mathcal{S}(\mathbf{p}) \mid \mathcal{S}\ consistent\}$ that is closest to the camera volume. We claim that $\mathbf{V}_\mathbf{p}$ is a color invariant. To establish this, observe that $\mathbf{V}_\mathbf{p} \in \mathcal{S}$ implies $\mathbf{V}_\mathbf{p} = \mathcal{S}(\mathbf{p})$, for if $\mathbf{V}_\mathbf{p} \neq \mathcal{S}(\mathbf{p})$, $\mathcal{S}(\mathbf{p})$ must be closer to the camera volume, a violation of our assumptions. It follows from Eq. (1) that $\mathbf{V}_\mathbf{p}$ has the same color in every consistent scene, i.e., $\mathbf{V}_\mathbf{p}$ is a color invariant.

Note that the preceding argument demonstrated not only that color invariants exist, but that *every* pixel in the basis images has a corresponding color invariant. We denote the collection of these color invariants as $\overline{\mathcal{S}}$:

$$\overline{\mathcal{S}} \;=\; \{\mathbf{V}_\mathbf{p} \mid \mathbf{p} \in \mathcal{I}_i,\ 0 \leq i \leq n\}$$

It is easily shown that $\overline{\mathcal{S}}$ is a consistent scene. Note that $\overline{\mathcal{S}}$ is complete, since it contains a voxel corresponding to each pixel in the basis images. To show that it is consistent, for each $\mathbf{V} \in \overline{\mathcal{S}}$, choose $\mathbf{p} \in \mathcal{I}_i$, $0 \leq i \leq n$, such that $\mathbf{V} = \overline{\mathcal{S}}(\mathbf{p})$. Define

$$color(\mathbf{V}, \overline{\mathcal{S}}) := color(\mathbf{p}, \mathcal{I}_i) \tag{2}$$

To show that this coloring is well defined, suppose $\mathbf{p} \in \mathcal{I}_i$ and $\mathbf{q} \in \mathcal{I}_j$ are two points such that $\overline{\mathcal{S}}(\mathbf{p}) = \mathbf{V} = \overline{\mathcal{S}}(\mathbf{q})$. Let $\mathcal{S}$ be a consistent scene such that $\mathbf{V} \in \mathcal{S}$. By the definition of $\overline{\mathcal{S}}$, it follows that $\mathcal{S}(\mathbf{p}) = \mathbf{V} = \mathcal{S}(\mathbf{q})$. Hence, by Eq. (1),

$$color(\mathbf{p}, \mathcal{I}_i) = color(\mathbf{V}, \mathcal{S}) = color(\mathbf{q}, \mathcal{I}_j)$$

Therefore Eq. (2) is a well-defined voxel coloring and is consistent with the basis images.

Fig. 4 shows $\overline{\mathcal{S}}$ for the pair of images in Figs. 2 and 3. These six voxels have a unique color interpretation, constant in every consistent scene. They also comprise the closest consistent scene to the cameras in the following sense—every point in each consistent scene is either contained in $\overline{\mathcal{S}}$ or is occluded by points in $\overline{\mathcal{S}}$. An interesting consequence of this distance bias is that neighboring image pixels of the same color produce cusps in $\overline{\mathcal{S}}$, i.e., protrusions toward the camera volume. This phenomenon is clearly shown in Fig. 4, where the black and gray points form two separate cusps. Also, observe that $\overline{\mathcal{S}}$ is

not a minimal reconstruction; removing the two closest points in Fig. 4 still leaves a consistent scene.

In summary, the following properties of color invariants have been shown:

- Every voxel in $\overline{\mathcal{S}}$ is a color invariant
- $\overline{\mathcal{S}} \subset \aleph$, i.e., $\overline{\mathcal{S}}$ is a consistent scene
- $\overline{\mathcal{S}}$ represents a *limit point* of $\aleph$, corresponding to the consistent scene that is closest to the camera volume

4 A Voxel Coloring Algorithm

We now describe how to compute $\overline{\mathcal{S}}$ via a single pass through a discretized scene volume, by exploiting the ordinal visibility constraint. This constraint limits the possible basis view configurations, but the benefit is that visibility relationships are greatly simplified. In particular, it becomes possible to partition the scene into a series of voxel *layers* that obey a monotonic visibility relationship: for *every* input image, voxels only occlude other voxels that are in subsequent layers. Consequently, visibility relationships are resolved by evaluating voxels one layer at a time.

4.1 Layered Scene Decomposition

To formalize the idea of visibility ordering, we define the following partition of 3D space into voxel layers of uniform distance from the camera volume:

$$\mathcal{V}_d = \{V \mid \mathcal{D}(V) = d\} \tag{3}$$

$$\mathcal{V} = \bigcup_{i=1}^{r} \mathcal{V}_{d_i} \tag{4}$$

where $d_1, \ldots, d_r$ is an increasing sequence of numbers and $\mathcal{D}$ is an occlusion-compatible function.

For the sake of illustration, consider a set of views positioned along a line facing a two-dimensional scene, as shown in Fig. 6 (a). Choosing $\mathcal{D}$ to be orthogonal distance to the line gives rise to a series of parallel linear layers that move away from the cameras. Notice that for any two voxels $\mathbf{P}$ and $\mathbf{Q}$, $\mathbf{P}$ can occlude $\mathbf{Q}$ from a basis viewpoint only if $\mathbf{Q}$ is in a higher layer than $\mathbf{P}$. The simplification of visibility relationships for this special case of colinear views was previously noted by Katayama et al. [21].

The linear case is easily generalized for any set of cameras satisfying the ordinal visibility constraint. Fig. 6 (b) shows a layer partition for the case of outward-facing cameras. This type of camera geometry is useful for acquiring *panoramic* scene visualizations, as in [30, 20]. One valid set of layers corresponds to a series of rectangles radiating outward from the camera volume. Layer 0 is the axis-aligned bounding box $\mathcal{B}$ of the camera centers and the subsequent layers are determined by uniformly expanding the box one unit at a time. This set of layers corresponds to an occlusion-compatible function given by the L_∞ distance to $\mathcal{B}$.

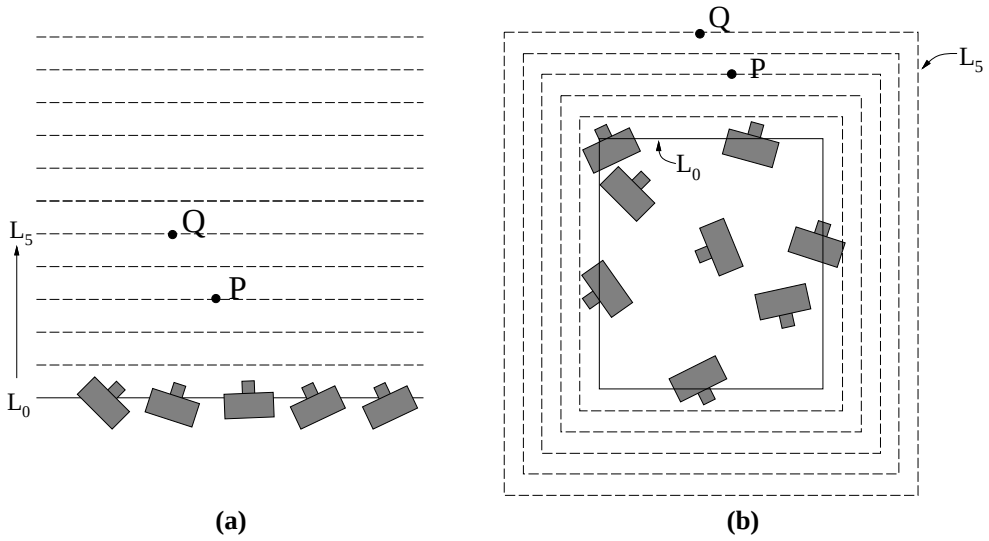


Figure 6: Layered Scene Traversal. Voxels can be partitioned into a series of layers of increasing distance from the camera volume. (a) Layers for cameras along a line. (b) Layers for cameras in a plane.

Decomposing a 3D scene into layers can be done in the same manner. In the 3D case the layers become surfaces that expand outward from the camera volume. An especially useful layering strategy is the 3D analog of Fig. 6(b), in which each layer is an axis-aligned cube. The advantage of this choice of layers is that layers are computed and traversed very efficiently.

4.2 Voxel Consistency

To compensate for the effects of image quantization and noise, suppose now that the images are discretized on a grid of finite non-overlapping pixels. If a voxel $\mathbf{V}$ is not fully occluded in image $\mathcal{I}_j$, its projection overlaps a nonempty set of image pixels, π_j. Without noise or quantization effects, a consistent voxel should project to a set of pixels with equal color values. In the presence of these effects, we evaluate the correlation $\lambda_{\mathbf{V}}$ of the pixel colors to measure the likelihood of voxel consistency. Let s be the standard deviation and m the cardinality of $\bigcup_{j=0}^{n} \pi_j$. One possible choice of correlation function is specified by thresholding the color space error:

$$\lambda_{\mathbf{V}} = s \tag{5}$$

Alternatively, a statistical measure of voxel consistency can be used. In particular, suppose the sensor error (accuracy of irradiance measurement) is normally distributed[3] with standard deviation σ_0. The consistency of a voxel can be estimated using the likelihood ratio test, distributed as χ^2 with $n-1$ degrees of freedom [15]:

$$\lambda_{\mathbf{V}} = \frac{(m-1)s^2}{\sigma_0^2} \tag{6}$$

[3] Here we make the simplifying assumption that σ_0 does not vary as a function of image intensity.

If σ_0 is unknown, it can be estimated by imaging a homogeneous surface and computing the standard deviation s_0 of m' image pixels. In this case, Eq. (6) should be replaced with

$$\lambda_{\mathbf{V}} = \frac{s^2}{s_0^2} \tag{7}$$

which has an F distribution with $m-1$ and $m'-1$ degrees of freedom.

4.3 A Single-Pass Algorithm

In order to evaluate the consistency of a voxel, we must first compute π_j, the set of pixels that overlap $\mathbf{V}$'s projection in $\mathcal{I}_j$. Neglecting occlusions, it is straightforward to compute a voxel's image projection, based on the voxel's shape and the known camera configuration. We use the term *footprint*, following [49] to denote this projection, corresponding to the intersection with the image plane of all rays from the camera center intersecting the voxel. Accounting for occlusions is more difficult, however, and we must take care to include only the images and pixel positions from which $\mathbf{V}$ should be *visible*. This difficulty is resolved by using the ordinal visibility constraint to visit voxels in an occlusion-compatible order and *marking* pixels as they are accounted for.

Initially, all pixels are unmarked. When a voxel is visited, π_j is defined to be the set of unmarked pixels that overlap $\mathbf{V}$'s footprint in $\mathcal{I}_j$. When a voxel is evaluated and found to be consistent, all pixels in π_j are marked. Because of the occlusion-compatible order of voxel evaluation, this strategy is sufficient to ensure that π_j contains only the pixels from which each voxel is visible, i.e., $\overline{\mathcal{S}}(\mathbf{p}) = \mathbf{V}$ for each $\mathbf{p} \in \pi_j$. Note that by assumption voxels within a layer do not occlude each other. Therefore, the pixel marking phase can be delayed until after all the voxels in a layer are evaluated.

The complete voxel coloring algorithm can now be presented as follows:

```
S̄ = ∅                                                    Initialize the reconstruction
for i = 1, ..., r do                                     Iterate through the layers
   for every V ∈ V_{d_i} do                              Iterate through voxels in the layer
      for j = 0, ..., n do                               Project the voxel to each image
         compute footprint ρ of V in I_j
         π_j = {p ∈ ρ | p unmarked}
      end for j
      compute λ_V                                        Evaluate voxel consistency
      if ⋃_{j=0}^{n} π_j is not empty and λ_V < thresh then
         S̄ = S̄ ∪ {V}                                     Color the voxel
         π = π ∪ ⋃_{j=0}^{n} π_j                         Remember image pixels to mark
      end if
   end for V
   mark pixels in π
end for i
```

The threshold, $thresh$, corresponds to the maximum allowable correlation error. An overly conservative (small) value of $thresh$ results in an accurate but incomplete reconstruction. On the other hand, a large threshold yields a more complete reconstruction, but one that includes some erroneous voxels. Instead of thresholding correlation error, it is possible to optimize for model *completeness*. In particular, a completeness threshold $tcomp$ may be chosen that specifies the minimum allowable percentage of image pixels left unmarked. For instance, $tcomp = 75\%$ requires that at least three quarters of the (non-background) image pixels correspond to the projection of a colored voxel.

Given $tcomp$, we seek the minimum value of $thresh$ that yields a voxel coloring achieving this completeness threshold. Since completeness increases monotonically with $thresh$, it is sufficient to run the single-pass algorithm for a succession of increasing values of $thresh$, stopping when $tcomp$ is achieved. Alternatively, a binary search on $thresh$ may be used to decrease the number of iterations.

4.4 Discussion

The voxel coloring algorithm visits each of the N^3 voxels exactly once and projects it into every image. Therefore, the time complexity of voxel coloring is: $O(N^3 n)$. To determine the space complexity, observe that evaluating one voxel does not require access to or comparison with other voxels. Consequently, voxels need not be stored in main memory during the algorithm; the reconstruction will simply be output one voxel at a time. Only the images and one-bit mark masks need to be allocated. The fact that the space and time complexities of voxel coloring are linear in the number of images is essential so that large numbers of images can be processed at once.

The algorithm differs from stereo and feature tracking techniques in that it does not perform window-based image correla-

tion during the reconstruction process. Correspondences are found during the course of scene traversal by voxel projection. A disadvantage of this searchless strategy is that it requires very precise camera calibration to achieve the triangulation accuracy of stereo methods. The effects of calibration and quantization errors are most significant at high-frequency regions such as image edges. Preserving high-frequency image content requires a higher voxel sampling rate because of Nyquist considerations. However, smaller voxels result in fewer pixels being integrated in the correlation step and are therefore more sensitive to calibration errors. An extension would be to compensate for high-frequency regions in the correlation step, for instance by detecting and treating edges specially.

Accuracy and run-time also depend on the voxel resolution, a parameter that can be set by the user or determined automatically to match the pixel resolution, calibration accuracy, and computational resources. An extension would be to use *hierarchical* representations like octrees [37, 45] in which the voxel resolution is locally adapted to match surface complexity.

Like previous work in stereo, voxel coloring makes use of the Lambertian assumption to simplify correlating pixels from different viewpoints. While this assumption provides a reasonable model for matte surfaces that scatter light approximately uniformly in all directions, it is not well-suited for reconstructing highly specular surfaces. Because specularities are not explicitly modeled, they will cause artifacts in the reconstruction, caused by the removal of surface voxels on the specularity that are deemed inconsistent with the Lambertian model. It may be possible to mitigate these artifacts by identifying specular highlights as a preprocessing step [7, 22] or by using a consistency criterion that models non-Lambertian effects [42, 24].

Importantly, the voxel coloring approach reconstructs only one of the potentially numerous scenes consistent with the input images. Consequently, it is susceptible to aperture problems caused by image regions of near-uniform color. These regions cause cusps in the reconstruction (see Fig. 4), since voxel coloring yields the reconstruction closest to the camera volume. This is a bias, just like smoothness is a bias in stereo methods, but one that guarantees a consistent reconstruction even with severe occlusions.

4.5 Optimizations

Much of the work of the algorithm lies in the computations of π_j and $\lambda_{\mathbf{V}}$. For simplicity, our implementation used a square mask to approximate voxel footprints, and Eq. (5) to test voxel consistency. Alternative footprint models are discussed in the volume rendering literature, e.g., [49, 25].

While our implementation did not make use of this, additional speedups are possible by exploiting the uniform discretization of space and simple layer geometry. Choosing planar or polyhedral layers enables the use of texture-mapping graphics hardware to calculate voxel footprints, an entire layer at a time. This strategy enables more accurate estimates of voxel footprints and offloads most of the computation to the graphics co-processor. For instance, the projection by $\mathbf{\Pi}_i$ of a plane layer

$$\mathbf{V}_{u,v} = \mathbf{V}_{0,0} + u\mathbf{D}_X + v\mathbf{D}_Y$$

can be expressed in matrix form by $\mathbf{\Pi}_i \mathbf{V}_{u,v} = \mathbf{H}_i [u\ v\ 1]^T$, where the 3×3 homography $\mathbf{H}_i$ is given by

$$\mathbf{H} = [\mathbf{\Pi}_i \mathbf{D}_X \mid \mathbf{\Pi}_i \mathbf{D}_Y \mid \mathbf{\Pi}_i \mathbf{V}_{0,0}]$$

Instead of projecting the layer onto each image, it is preferable to reverse-map each image onto the layer by computing

$$\hat{\mathcal{I}}_i = \mathbf{H}_i^{-1} \mathcal{I}_i$$

This procedure allows the voxel projections to be directly integrated; the footprint of voxel $\mathbf{V}_{u,v}$ in $\mathcal{I}_i$ is simply the pixel at position $[u\ v\ 1]^T \in \hat{\mathcal{I}}_i$. This reverse-mapping strategy is similar to that of Collins [9], who used a diffusion operator in place of the texture-mapping formulation.

5 Experimental Results

In order to evaluate the performance of the voxel coloring algorithm for view synthesis, it was applied to images of a variety of real scenes. Images of synthetic scenes were also used to facilitate analysis of error characteristics, and to simulate camera configurations that were not physically realizable in the lab.

5.1 Results on Real Images

The first experiment demonstrates the view synthesis capabilities of the voxel coloring algorithm, as applied to images of real objects. Calibrated images were captured with the aid of a computer-controlled pan-tilt head and a fixed overhead camera, as shown in Fig. 8. This image-acquisition strategy is similar to that used in [45]. An object was placed on the head and rotated 360 degrees in front of a color camera (Sony XC-999 with 1/2" CCD sensor and 12mm, F1.4 lens) positioned approximately 30cm horizontally from the object's center and 25cm vertically above its base. Tsai's method [48] was used to calibrate the camera with respect to the head, by rotating a known object and manually selecting image features for three pan positions. The calibration error was approximately 3%. Fig. 7 shows selected images for two objects: a toy dinosaur (6cm x 8cm x 10cm) and a rose. In each case 21 input images (640×486 resolution) were captured by rotating the object nearly 360 degrees in increments of about 17 degrees. A problem with this acquisition approach is that the illumination effectively changes as the object rotations, thereby violating the Lambertian assumption. To compensate, the error threshold was set relatively high: 18% pixel correlation error was allowed for the dinosaur, and 12% for the rose.

Table 1 compares sizes, run times (on a 250MHz MIPS R4400 processor), and reprojection errors for voxel colorings computed from the dinosaur toy at four different resolutions. Square voxels were used and the grid volume was held constant. The resolution was specified by the grid dimensions, indicating the total number of voxels in the volume (width x depth x height). Each row in Table 1 represents a resolution doubling, i.e., an 8-fold increase in the number of voxels, relative to the previous row. The run time increases proportionately, although not quite by a factor of 8 due to an additive overhead factor of the algorithm. These run times do not include image acquisition, calibration, or background thresholding. The "Voxels

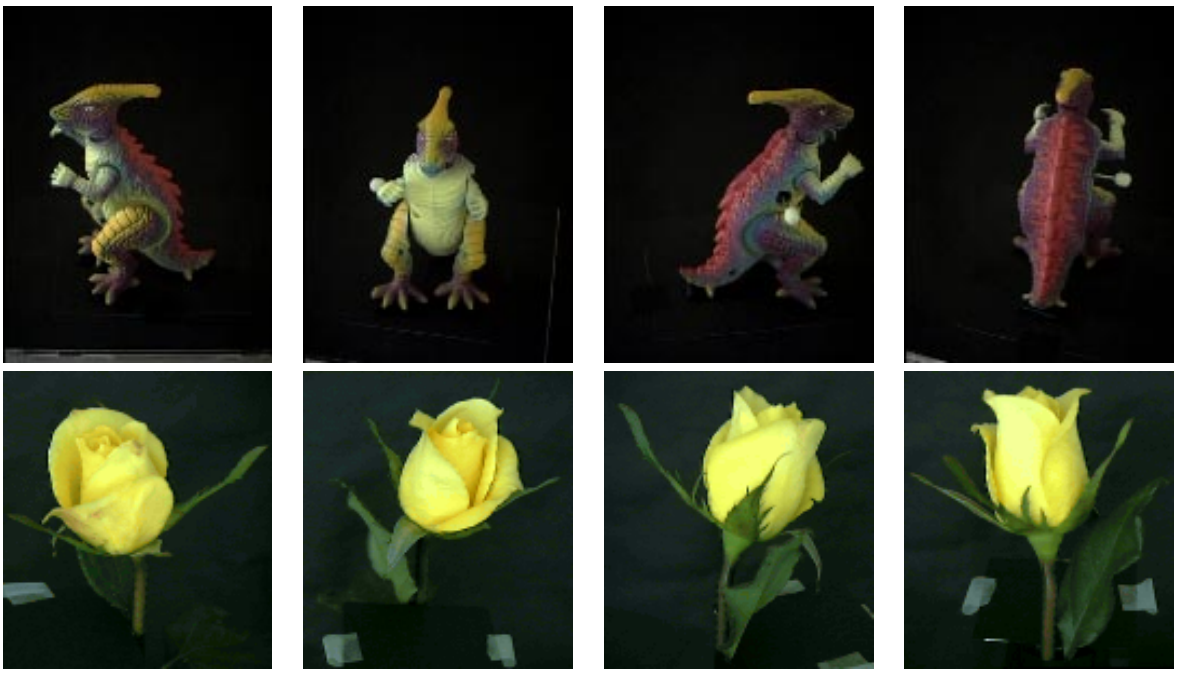


Figure 7: Selected basis images for a dinosaur toy (top) and a rose (bottom). 21 images were taken in all, spanning close to a 360° rotation of each object.

Figure 8: Image Acquisition Setup. Basis images were captured by placing an object on a calibrated pan-tilt head and rotating the object in front of a stationary video camera. The camera was placed above the object to satisfy the ordinal visibility constraint.

Grid Dimensions	Voxels Evaluated	Voxels Colored	Run Time	Reprojection Error
$20 \times 24 \times 29$	13,920	902	3 sec	9.38%
$41 \times 49 \times 58$	116,522	4,898	11 sec	8.01%
$83 \times 99 \times 116$	953,172	21,174	62 sec	7.48%
$166 \times 199 \times 233$	7,696,922	71,841	435 sec	7.20%

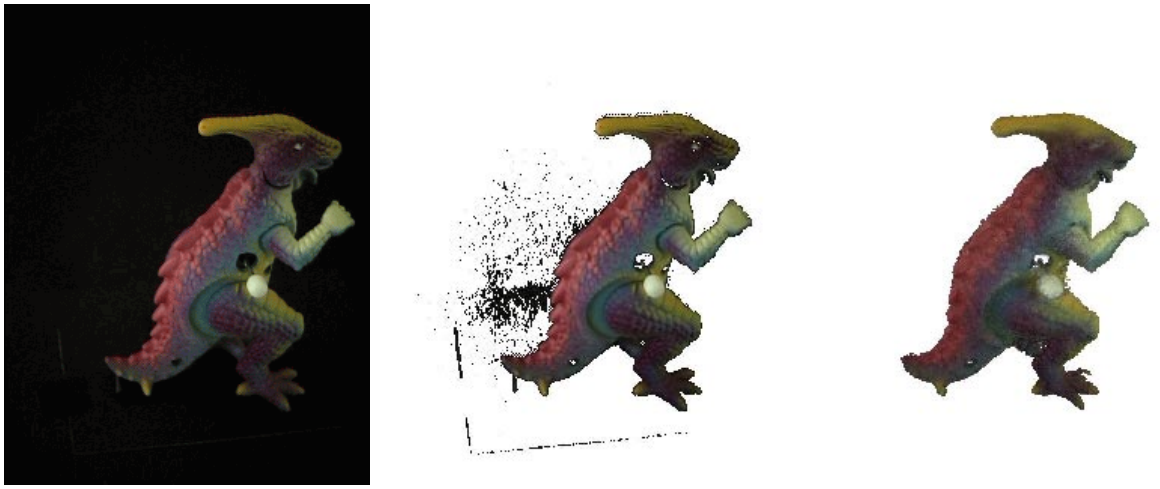
Table 1: Voxel Resolution Effects. This table compares the size, run time, and reprojection error for voxel colorings of the dinosaur toy using different grid sizes. Each row represents an 8-fold increase in the number of voxels relative to the previous row. The corresponding models are shown in Fig. 9.

Colored" column indicates the number of voxels in the final reconstruction. Notice that the voxels colored column increases more slowly than the voxels evaluated column. This is attributed to the fact that the voxel coloring algorithm reconstructs only points on the *surface*, i.e., not in the interior of the object. It would therefore be expected that for manifold surfaces, the number of voxels colored would increase only by a factor of 4 when the resolution is doubled. The final column of Table 1 gives the reprojection error, which measures the root-mean-squared error of pixels in synthesized images for each of the 21 input viewpoints. Clearly, increasing the voxel resolution yields a significant improvement in reprojection error.

Fig. 9 shows the voxel colorings of the dinosaur toy described in Table 1. To facilitate reconstruction, a black background was used and the images were thresholded to eliminate most of the background points. While background segmentation is not strictly necessary, leaving this step out results in background-colored voxels scattered around the edges of the scene volume. The background threshold may be chosen conservatively since removing most of the background pixels is sufficient to eliminate this background scattering effect. Fig. 9(c) shows the highest resolution reconstruction from a viewpoint corresponding to one of the input images. For comparison, the original image is also shown in Fig. 9(a). Note that even fine details such as the wind-up rod on the dinosaur were reconstructed, as seen more clearly in (g).

Fig. 9(d-g) compare reconstructions from a new viewpoint, different than the basis views, for each of the voxel resolutions reported in Table 1. The resolution doubles at each step from (d) to (g). Note that even the lowest resolution model, shown in (d), preserves the rough features of the model and produces a very reasonable reprojection. The fact that this model was computed in 3 seconds suggests that the algorithm is potentially suitable for interactive applications like teleconferencing, in which models must be generated in real time. Increasing the resolution adds fine details in shape and texture and decreases the blocky artifacts caused by large voxels, as seen in (e-g). For comparison, Fig. 11 shows the shaded underlying voxel grids without color information. Note that color greatly augments the visual quality of the reconstruction.

Results for the rose are shown in Fig. 10. (a) shows an input image and (b) a synthesized view for the same viewpoint, demonstrating the photo integrity of the reconstruction. The rose represents a difficult case because it has little texture and is made up of surfaces like leaves and petals that are extremely thin. The reconstruction, consisting of approximately 70,000 voxels, captures the appearance of the rose very accurately, and preserves most of these fine features. (c) and (d) show synthesized views from new views that are close to and far away from the basis views, respectively. Overall, the image in (d) is quite good, but it exhibits some interesting artifacts. Specifically, the leaves appear to contain holes when viewed from below. These holes are not visible from the basis viewpoints and so were not filled in by the algorithm—they represent unreconstructible regions in the scene.



Input Image
(a)

Thresholded Image
(b)

Model Reprojection
(c)

902 voxels
(d)

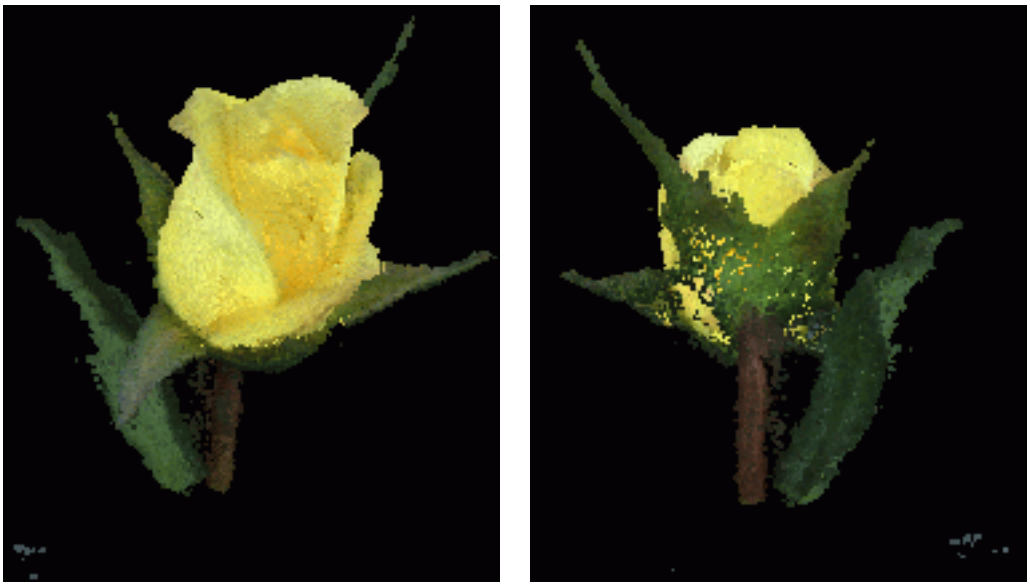
4,898 voxels
(e)

21,174 voxels
(f)

71,841 voxels
(g)

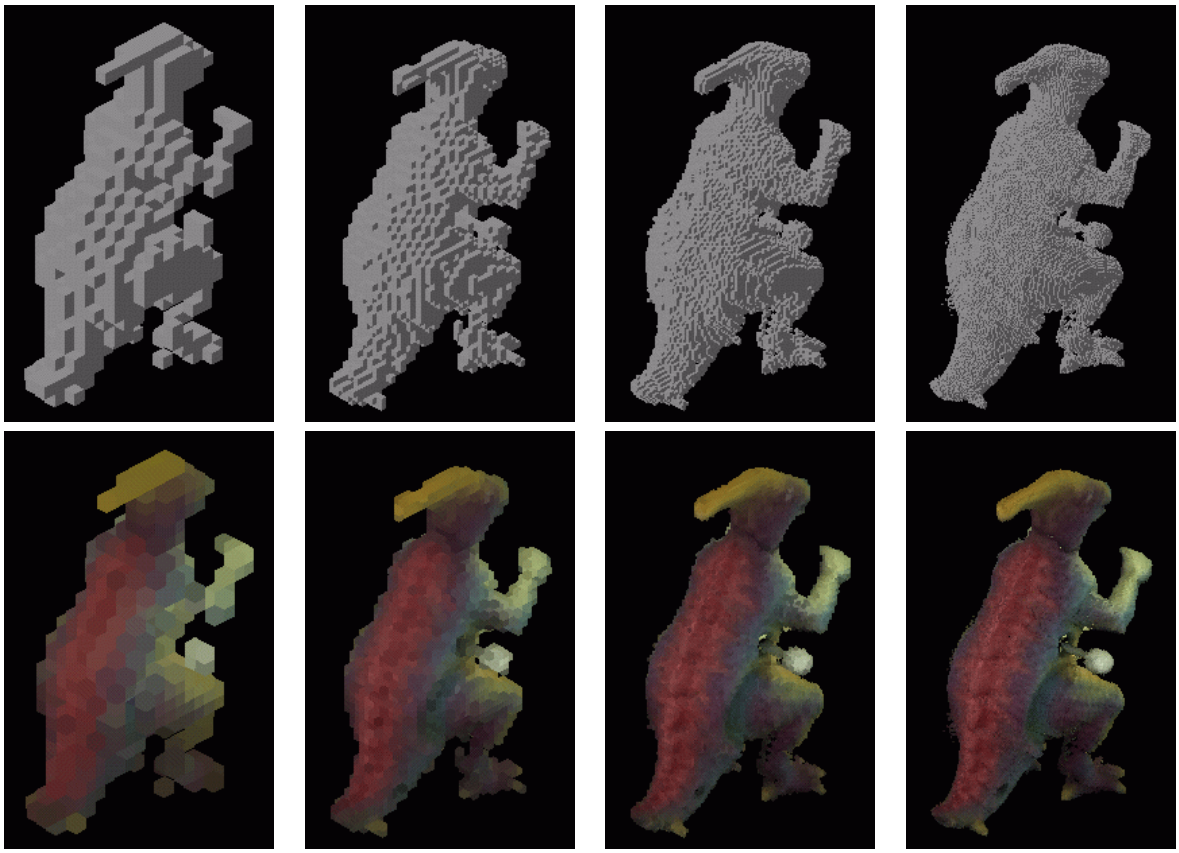
Figure 9: Voxel Coloring of a Dinosaur Toy. Original image (a) is thresholded (b) to eliminate most of the background pixels. The voxel coloring (c) was computed from 21 thresholded images of the object undergoing a 360° rotation. (d-g) show the reconstruction from a new viewpoint at different voxel resolutions, where the voxel width is progressively doubled.

(a) **(b)**



(c) **(d)**

Figure 10: Voxel Coloring of a Flower. Input image (a) is shown next to the projection of a voxel coloring for the same viewpoint (b). The reconstruction (70K voxels) captures image appearance very accurately for new views, such as (c), that are near the input viewpoints. Artifacts become more prevalent when the new viewpoint is far away from the input views, as in (d).



902 voxels (d) **4,898 voxels (e)** **21,174 voxels (f)** **71,841 voxels (g)**

Figure 11: Shaded (top) and colored (bottom) voxel models of a dinosaur toy at different resolutions.

(a) **(b)** **(c)**

Figure 12: Comparison of voxel coloring and silhouette-based reconstruction. Input image (a) is shown next to reconstructions rendered at the same viewpoint. (b): voxel coloring reconstruction. (c): silhouette-based reconstruction.

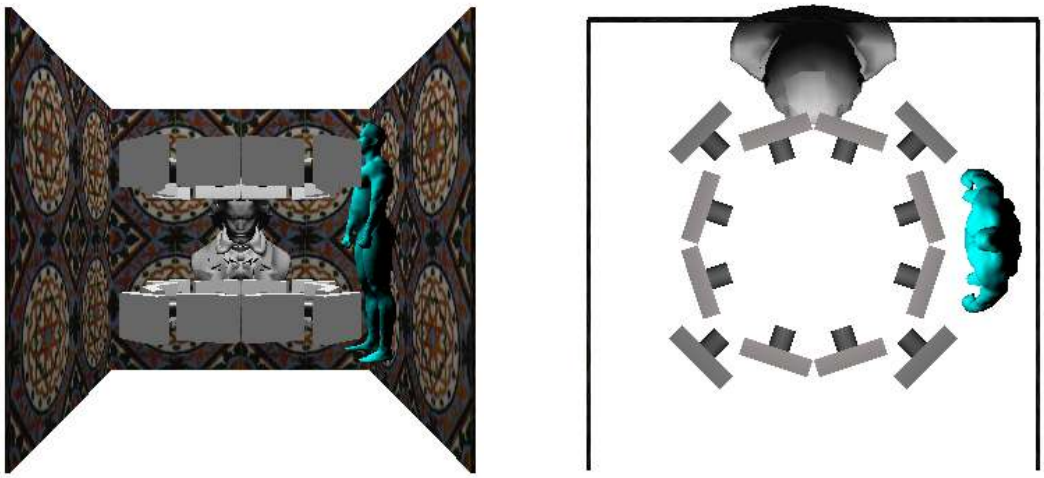


Figure 13: Placement of input camera viewpoints for a *panoramic* synthetic room scene. The camera positions and scene are shown together, from a frontal (left), and overhead (right) perspective.

Because background segmentation was used in the first two experiments, we had the opportunity to compare the quality of the reconstructions with silhouette-based methods such as volume intersection [29, 45, 26, 44]. To perform this comparison, we reconstructed the flower model twice, once with a threshold of 12%, and once with a threshold of ∞. Note that an infinite threshold produces a reconstruction that is consistent only with the silhouettes, i.e., color information is not used in the computation. Fig. 12 compares an input image with both of these reconstructions. Note that the silhouette-based model is too conservative–it contains voxels that lie outside the true shape. These errors lead two two noticeable artifacts: first, the silhouette model is too large. Second, the coloring of the model is of very poor quality because points on the model project to pixels of different colors in different views and therefore cannot be colored consistently. In contrast, voxel coloring provides a way of ensuring color consistency by allowing the user to provide an upper bound on the allowable color-space reprojection error for each voxel on the model (12% in the example in Fig. 12(c)).

5.2 Results on Synthetic Images

In order to evaluate the performance of the voxel coloring algorithm for *panoramic* scenes, basis images were generated by placing several cameras in a synthetic room scene. The room consisted of three texture-mapped walls and two shaded figures. The figures, a bust of Beethoven and a scanned Cyberware model of a human figure, were illuminated diffusely from a downward-oriented light source at infinity. 24 cameras were placed at different positions and orientations *inside* the room, as shown in Fig. 13.

The geometry of this scene and the camera configuration would pose significant problems for previous image-based reconstruction methods. In particular, the room interior is highly concave, making accurate reconstruction by volume intersection or other contour-based methods impractical. Furthermore, the numerous cameras and large amount of occlusion would create difficulty for most stereo approaches. Notable exceptions include panorama-based stereo approaches [30, 20] that are well-suited for room reconstructions. However, these methods require that a panoramic image be constructed for each camera location prior to the stereo matching step, a requirement that is avoided by the voxel coloring approach. This requirement does not enable camera configurations such as the one shown in Fig. 13.

Fig. 14 compares the original and reconstructed models of the room from new viewpoints. The reconstruction contained 320,000 colored voxels, out of approximately 50 million candidate voxels examined, and required 45 minutes to compute. The voxel coloring reproduced images from the room interior extremely accurately, as shown in (b). A pixel correlation error threshold of 2.4% was used to account for image quantization. As a result of these errors, some fine details were lost, e.g., in the face of the Beethoven bust. The overhead views (d) and (f) more clearly show some discrepancies between the original and reconstructed models. For instance, the reconstructed walls are not perfectly planar, as some points lie just off the surface. This point drift effect is most noticeable in regions where the texture is locally homogeneous, indicating that texture information is important for accurate reconstruction. The quality of the overhead view shown in (d) is especially commendable, given that the viewpoint is very far away from the input views. The extreme overhead view (f) is worse than that of (d) but clearly shows that the overall shape of the scene was very well captured by reconstruction.

A second set of experiments was conducted to evaluate the sensitivity of the approach to factors of texture density, image noise, and voxel resolution. To simplify the analysis of these effects, the experiments were performed using a 2D implementation of the voxel coloring method for which the scene and cameras lie in a common plane. Fig. 15(a) shows the synthetic scene (an arc) and the positions of the basis views used in these experiments.

Texture is an important visual cue, and one that is exploited by voxel coloring. To model the influence of texture on reconstruction accuracy, a series of reconstructions were generated in which the texture was systematically varied. The spatial structure of the scene was held fixed. The texture pattern was a cyclic linear gradient, specified as a function of frequency θ and position $t \in [0, 1]$:

$$intensity(t) = 1 - |1 - 2 * frac(\theta * t)|$$

$frac(x)$ returns the fractional portion of x. Increasing the frequency parameter θ causes the density of the texture to increase accordingly. Fig. 15(b-j) show the reconstructions obtained by applying voxel coloring for increasing values of θ. For comparison, the corresponding texture patterns and the original arc shapes are also shown. In (b), the frequency is so low that the quantized texture pattern is uniform. Consequently, the problem reduces to reconstruction from silhouettes and the result is similar to what would be obtained by volume intersection [29, 45, 26]. Specifically, volume intersection would yield a closed diamond-shaped region; the reconstructed V-shaped cusp surface in (b) corresponds to the set of surfaces of this diamond that are visible from the basis views.

Doubling θ results in a slightly better reconstruction consisting of two cusps, as shown in (c). Observe that the reconstruction is accurate at the midpoint of the arc, where a texture discontinuity occurs. Progressively doubling θ produces a

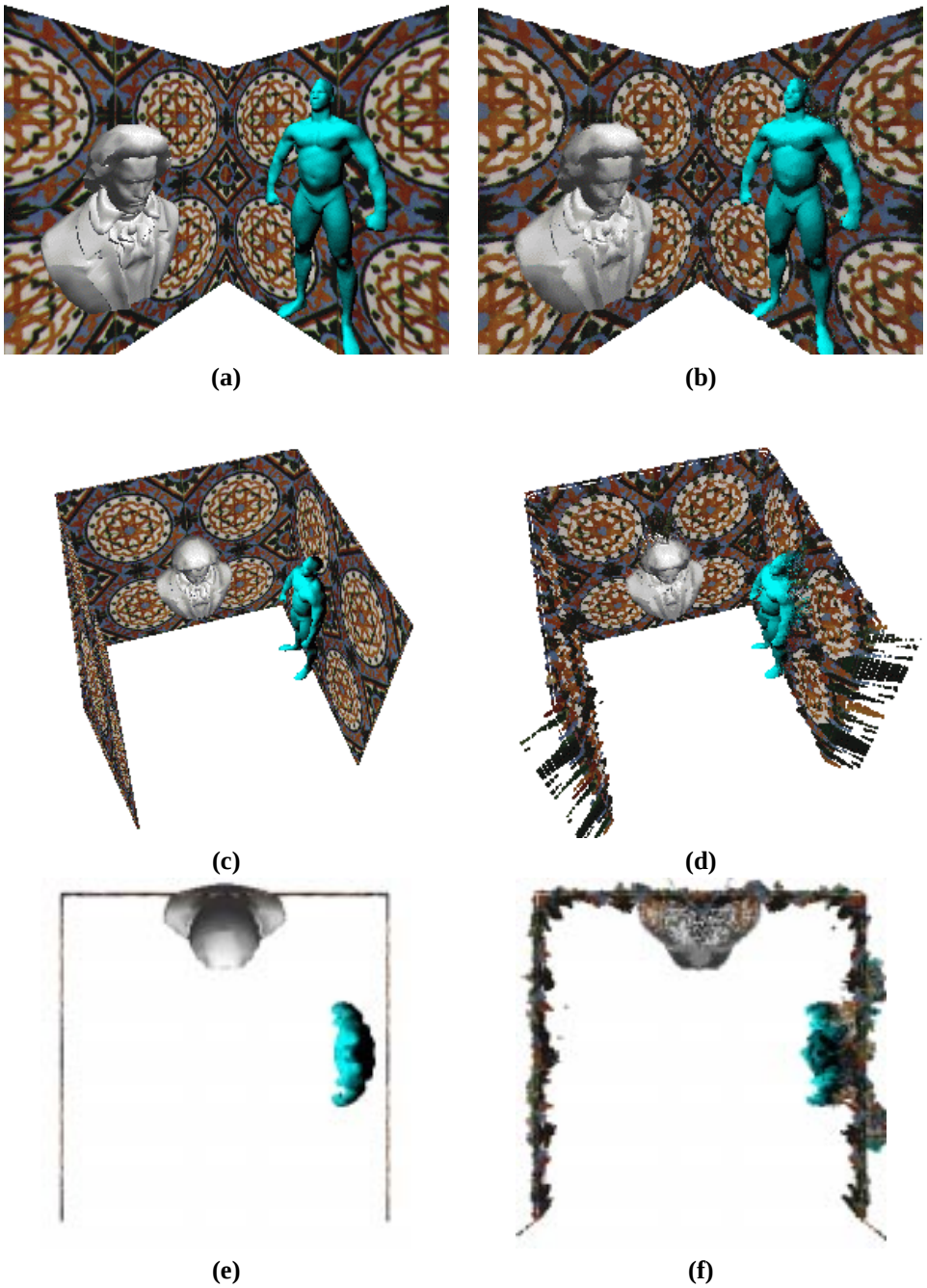


Figure 14: Panoramic room scene reconstruction. Renderings of the true scene are shown at left and the reconstructed scene at right. (a) and (b): an input viewpoint, from *inside* the room. (c) and (d): a new viewpoint from above the room. (e) and (f): an extreme overhead view. Fidelity is best near the input viewpoints and degrades smoothly as the camera moves further away.

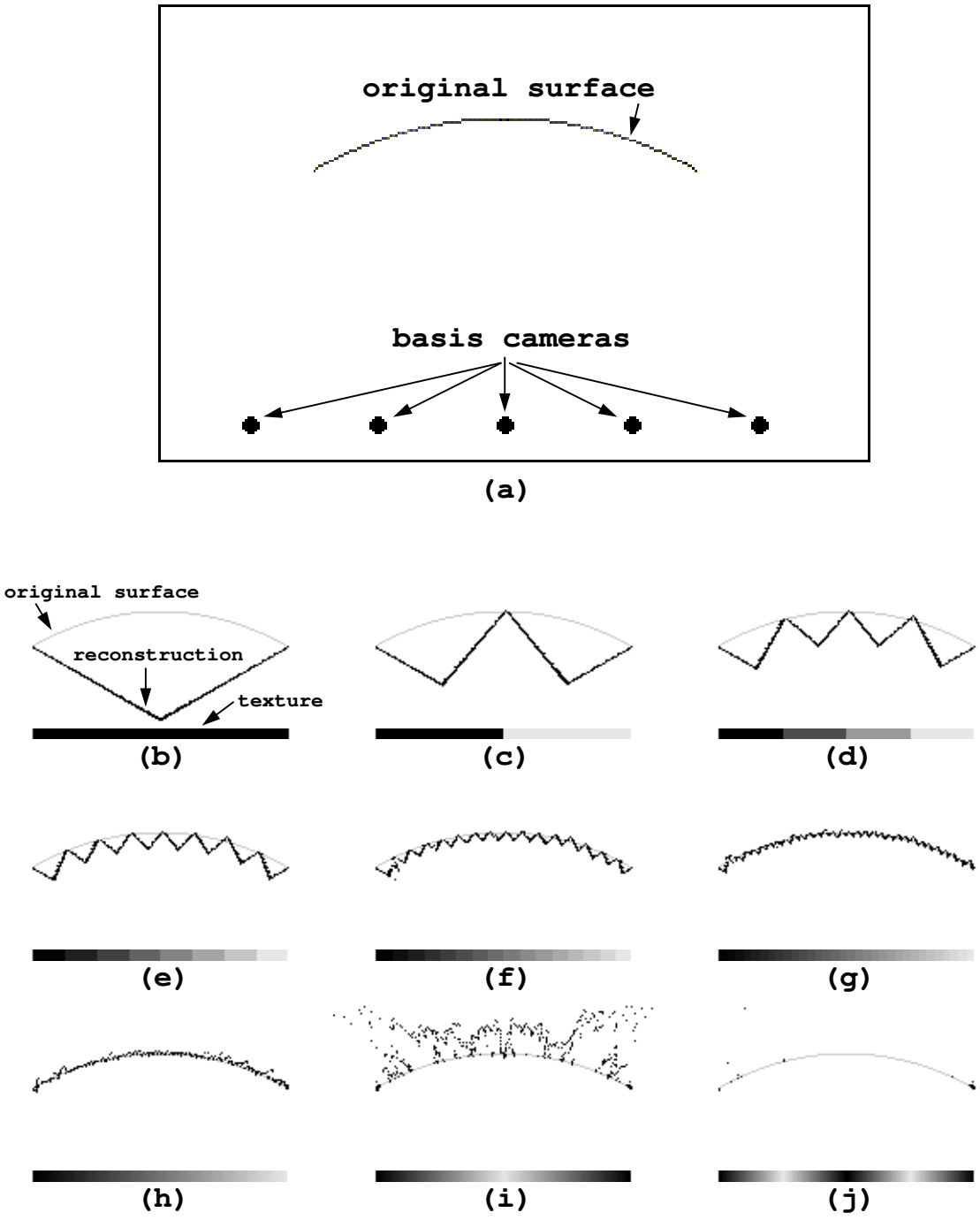


Figure 15: Effects of Texture Density on Voxel Reconstruction. (a): A synthetic arc is reconstructed from five basis views. The arc is textured with a cyclic gradient pattern with a given frequency. Increasing the frequency makes the texture denser and causes the accuracy of the reconstruction to improve, up to a limit. In the case of (b), the texture is uniform so the problem reduces to reconstruction from silhouettes. As the frequency progressively doubles (c-j), the reconstruction converges to the true shape, until a certain point beyond which it exceeds the image resolution (i-j).

series of more accurate reconstructions (d-h) with smaller and smaller cusps that approach the true shape. When θ exceeds a certain point, however, the reconstruction degrades. This phenomenon, visible in (i) and (j), results when the projected texture pattern exceeds the resolution of the basis images, i.e., when the Nyquist rate is exceeded. After this point, accuracy degrades and the reconstruction ultimately breaks up.

Fig. 15 illustrates the following two points: (1) reconstruction accuracy is strongly dependent upon surface texture, and (2) the errors are highly *structured*. To elaborate on the second point, reconstructed voxels drift from the true surface in a predictable manner as a function of local texture density. When the texture is locally homogeneous, voxels drift *toward* the camera volume. As texture density increases, voxels move monotonically away from the camera volume, toward the true surface. As texture density increases even further, beyond the limits of image resolution, voxels continue to move away from the cameras, and away from the true surface as well, until they are ultimately eliminated from the reconstruction.

We next tested the performance of the algorithm with respect to additive image noise. To simulate noise in the images, we perturbed the intensity of each image pixel independently by adding a random value in the range of $[-\sigma, \sigma]$. To compensate, the error threshold was set to σ. Fig. 16 shows the resulting reconstructions. The primary effect of the error and corresponding increase in the threshold was a gradual drift of voxels away from the true surface and toward the cameras. When the error became exceedingly large, the reconstruction ultimately degenerated to the "no texture" solution shown in Fig. 15(b). This experiment indicates that image noise, when compensated for by increasing the error threshold, also leads to structured reconstruction errors; higher levels of noise cause voxels to drift progressively closer to the cameras.

The final experiment evaluated the effects of increasing the voxel size on reconstruction accuracy. In principle, the voxel coloring algorithm is only correct in the limit, as voxels become infinitesimally small. In particular, the layering strategy is based on the assumption that points within a layer do not occlude each other. For very small voxels this no-occlusion model is a reasonable approximation. However, as voxels increase in size, the model becomes progressively less accurate. It is surprising, therefore, that the algorithm appears to produce good results even for very large voxel sizes, as seen in Fig. 9(d-e).

To more carefully observe the effects of voxel size, we ran the voxel coloring algorithm on the scene in Fig. 15(a) for a sequence of increasing voxel sizes. Fig. 16 shows the results—the reconstructions are close to optimal, up to the limits of voxel resolution, independent of voxel size. Again, this empirical result is surprising, given the obvious violation of the layering property which is the basis of the algorithm. Some effects of this violation are apparent; some voxels are included in the reconstruction that are clearly invisible, i.e., totally occluded by other voxels from the basis views. For instance, observe that in the reconstruction for voxel size = 10, the top-left and top-right voxels could be deleted without affecting scene appearance from the basis views. These extra voxels are artifacts of the large voxel size and the violation of the layering property. However, these effects are minor and do not adversely affect view synthesis in that adding these voxels does not change scene appearance for viewpoints close to the input images.

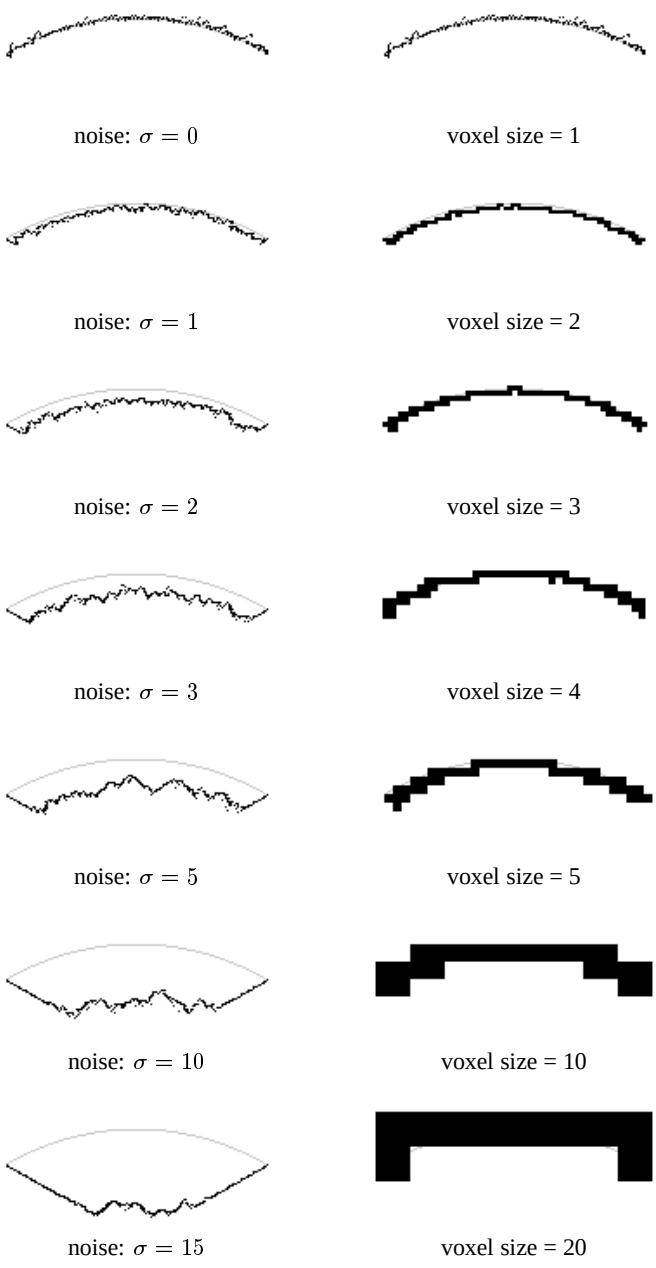


Figure 16: Effects of Image Noise and Voxel Size on Reconstruction. Image noise was simulated by perturbing each pixel by a random value in the range $[-\sigma, \sigma]$. Reconstructions for increasing values of σ are shown at left. To ensure a full reconstruction, the error threshold was also set to σ. Increasing noise caused the voxels to drift from the true surface (shown as light gray). The effects of changing voxel size are shown at right. Notice that the arc shape is reasonably well approximated even for very large voxels.

6 Discussion

This paper addressed the problem of view synthesis from numerous basis views distributed widely about a scene. This problem is especially challenging due to the difficulty of computing reliable correspondence information from views that are far apart. A main goal was to determine intrinsic ambiguities and limitations of what is reconstructible, and also to derive a practical algorithm for correspondence computation and view synthesis.

A primary contribution of this paper was the *voxel coloring* framework for analyzing ambiguities in image correspondence and scene reconstruction. A similar theory was previously developed for the special case of volume intersection, i.e., reconstruction from silhouettes [26, 23, 27]. The results in this paper can be viewed as a generalization of the volume intersection problem for the broader class of textured objects and scenes. The voxel coloring framework enabled the identification and computation of *color invariants*—points having the same color in every possible scene reconstruction, consistent with a set of basis images.

A second important contribution was the voxel coloring algorithm for computing pixel correspondence from a set of basis images. A key element was the *ordinal visibility constraint*, a novel constraint on the configuration of camera viewpoints that enabled an efficient solution to the correspondence problem. This is the first practical algorithm capable of generating provably-consistent dense correspondence maps from a set of input images, in the presence of occlusion. It is therefore useful not just for view synthesis, but for other applications that require correspondence, e.g., motion analysis and 3D scene reconstruction.

The algorithm has several novel features that make it especially attractive for view synthesis tasks:

- **Generality**: The low-level voxel representation can approximate any surface type and easily models discontinuities. It is therefore well-suited for modeling real-world objects and scenes of complex geometry and topology.
- **Flexible Acquisition**: The cameras may be arbitrarily far apart without degrading reconstruction accuracy. Indeed, the algorithm performs best when cameras are distributed widely about the scene.
- **Panoramic Visibility**: The voxel coloring method can synthesize views for any camera position and orientation. The fact that it is applicable for both inward- and outward-facing camera configurations makes it well-suited for a wide variety of scenes, from small objects to panoramic room interiors.
- **Insensitivity to Occlusion**: Changes in visibility are fully modeled by the algorithm and impose no performance penalty.
- **Efficiency**: The algorithm performs only a single pass through the scene volume and exploits regular operations that can be performed using texture-mapping graphics hardware. We are currently investigating a real-time implementation of the algorithm that would run on existing graphics workstations. The technique is also space efficient in that only *surface* voxels are stored, i.e., voxels that are visible in at least one basis image.
- **Scalability**: By varying the voxel size, the algorithm can be tailored to available computing resources. Empirical evidence demonstrates that the algorithm performs well for a range of different voxel sizes.

While the *ordinal visibility constraint* is needed to support a single-pass algorithm, it also rules out certain input camera configurations. Although the scene can surround the cameras, as in the room scene in Fig. 14, the cameras cannot surround the object or scene. This is not a serious limitation in controlled lab environments where the camera configuration may be designed with the ordinal visibility constraint in mind. For instance, in our experiments the cameras were raised slightly above the scene to be reconstructed (Fig. 8). However, it could be problematic for other situations, e.g., it would not allow reconstruction from a video sequence obtained by walking all the way around a large object with a video camera. We are currently investigating methods for handling these types of camera motions and configurations. One solution would be to segment the basis images into sets that individually satisfy the ordinal visibility constraint, run the algorithm on each set separately, and then merge the results. Automatic methods for performing this segmentation task would simplify this approach. Another possible approach would be to extend the voxel coloring algorithm to directly handle general camera configurations, e.g., by using a multi-pass approach [24].

The results in Section 5 indicate that low-contrast regions and noise produce reconstruction errors, e.g., *cusps*, that are highly structured. For view synthesis tasks, this is a potential disadvantage, since structured noise can produce perceptible artifacts. The fact that these errors are deterministic and well-understood indicates that they are potentially detectable, and thus could be attenuated. One solution would be to perform a post-processing phase, analogous to dithering [14], in which errors are diffused to mitigate these artifacts. An alternative method would be to identify textureless regions and other error sources in the basis images and treat these features specially in the reconstruction process.

Two key challenges for future work are modeling specularities and transparency. Strong specularities cause noticeable artifacts in reconstructions obtained by voxel coloring due to the inherent limitations of the underlying Lambertian model. It may be possible to remove these artifacts by enriching the consistency criterion to include non-Lambertian effects, e.g., by solving for surface normal and reflectance in the reconstruction process. This is a topic that we are currently investigating [42, 24]. A second challenge is modeling transparency effects, caused both by variations in surface opacity as well as *mixed-pixels* caused by image discretization [46]. Transparency is problematic because multiple scene points generally contribute to the color of a single image pixel. Consequently, scene consistency cannot be evaluated on a point-by-point basis, as proposed in this paper. Promising initial results by Szeliski and Golland [46], indicate that global optimization methods provide a potential solution the transparency problem and are worthy of future study.

The notion of *color invariance* bears resemblence to the problem of *color constancy* (c.f. [18]) which has a long history in the computer vision literature. The color constancy problem is to determine, from one or more images, a description of scene material properties (e.g., surface reflectance) that does not depend on scene illumination. However, invariance to illumination is quite different than the notion of color invariance used in this paper; in the latter case, illumination is held *fixed*. Rather than separating reflectance from illumination, color invariants encode scene radiance directly, which is sufficient to synthesize new views of the scene with illumination held constant.

References

[1] Shai Avidan and Amnon Shashua. Novel view synthesis in tensor space. In *Proc. Computer Vision and Pattern Recognition Conf.*, pages 1034–1040, 1997.

[2] H. Harlyn Baker and Robert C. Bolles. Generalizing epipolar-plane image analysis on the spatiotemporal surface. *Int. J. of Computer Vision*, 3(1):33–49, 1989.

[3] Paul Beardsley, Phil Torr, and Andrew Zisserman. 3D model acquisition from extended image sequences. In *Proc. European Conf. on Computer Vision*, pages 683–695, 1996.

[4] Peter N. Belhumeur and David Mumford. A Bayesian treatment of the stereo correspondence problem using half-occluded regions. In *Proc. Computer Vision and Pattern Recognition Conf.*, pages 506–512, 1992.

[5] David Beymer and Tomaso Poggio. Image representations for visual learning. *Science*, (272):1905–1909, 1996.

[6] Robert C. Bolles, H. Harlyn Baker, and David H. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *Int. J. of Computer Vision*, 1(1):7–55, 1987.

[7] G. Brelstaff and A. Blake. Detecting specular reflections using lambertian constraints. In *Proc. 2nd Int. Conf. on Computer Vision*, pages 297–302, 1988.

[8] Shenchang Eric Chen and Lance Williams. View interpolation for image synthesis. In *Proc. SIGGRAPH 93*, pages 279–288, 1993.

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

[10] Robert T. Collins. Multi-image focus of attention for rapid site model construction. In *Proc. Computer Vision and Pattern Recognition Conf.*, pages 575–581, 1997.

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

[12] Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *Proc. SIGGRAPH 96*, pages 11–20, 1996.

[13] O. D. Faugeras and R. Keriven. Variational principles, surface evolution, PDE's, level set methods and the stereo problem. *IEEE Trans. on Image Processing*, 7(3):336–344, 1998.

[14] R. W. Floyd and L. Steinberg. An adaptive algorithm for spatial gray scale. In *SID 75, Int. Symp. Dig. Tech. Papers*, page 36. 1975.

[15] John E. Freund. *Mathematical Statistics.* Prentice Hall, Englewood Cliffs, NJ, 1992.

[16] Pascal Fua and Yvan G. Leclerc. Obejct-centered surface reconstruction: Combining multi-image stereo and shading. *Int. J. of Computer Vision*, 16:35–56, 1995.

[17] Davi Geiger, Bruce Landendorf, and Alan Yuille. Occlusions and binocular stereo. In *Proc. European Conf. on Computer Vision*, pages 425–433, 1992.

[18] Glenn E. Healey, Steven A. Shafer, and Lawrence B. Wolff, editors. *Physics-based vision: Principles and Practice, COLOR*, pages 205–299. Jones and Bartlett, Boston, MA, 1992.

[19] Takeo Kanade, Peter Rander, and P. J. Narayanan. Virtualized reality: Constructing virtual worlds from real scenes. *IEEE Multimedia*, 4(1):34–46, 1997.

[20] Sing Bing Kang and Richard Szeliski. 3-D scene data recovery using omnidirectional multibaseline stereo. In *Proc. Computer Vision and Pattern Recognition Conf.*, pages 364–370, 1996.

[21] Akihiro Katayama, Koichiro Tanaka, Takahiro Oshino, and Hideyuki Tamura. A viewpoint dependent stereoscopic display using interpolation of multi-viewpoint images. In *Proc. SPIE Vol. 2409A*, pages 21–30, 1995.

[22] G. J. Klinker and S. A. Shafer. A physical approach to color image understanding. *Int. J. of Computer Vision*, 4(7):7–38, 1990.

[23] Kiriakos N. Kutulakos and Charles R. Dyer. Global surface reconstruction by purposive control of observer motion. *Artificial Intelligence*, 78:147–177, 1995.

[24] Kiriakos N. Kutulakos and Steven M. Seitz. A theory of shape by space carving. Technical Report 692, Computer Science Dept., University of Rochester, Rochester, New York, May 1998.

[25] David Laur and Pat Hanrahan. Hierarchical splatting: A progressive refinement algorithm for volume rendering. In *Proc. SIGGRAPH 91*, 1991.

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

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

[28] Stephane Laveau and Olivier Faugeras. 3-D scene representation as a collection of images. In *Proc. Int. Conf. on Pattern Recognition*, pages 689–691, 1994.

[29] W. N. Martin and J. K. Aggarwal. Volumetric description of objects from multiple views. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 5(2):150–158, 1991.

[30] Leonard McMillan and Gary Bishop. Plenoptic modeling. In *Proc. SIGGRAPH 95*, pages 39–46, 1995.

[31] Saied Moezzi, Arun Katkere, Don Y. Kuramura, and Ramesh Jain. Reality modeling and visualization from multiple video sequences. *IEEE Computer Graphics and Applications*, 16(6):58–63, 1996.

[32] Yuichi Nakamura, Tomohiko Matsuura, Kiyohide Satoh, and Yuichi Ohta. Occlusion detectable stereo–occlusion patterns in camera matrix. In *Proc. Computer Vision and Pattern Recognition Conf.*, pages 371–378, 1996.

[33] P. J. Narayanan, Peter W. Rander, and Takeo Kanade. Constructing virtual worlds using dense stereo. In *Proc. Sixth Int. Conf. on Computer Vision*, pages 3–10, 1998.

[34] Tomaso Poggio, Vincent Torre, and Christof Koch. Computational vision and regularization theory. *Nature*, 317:314–319, 1985.

[35] Marc Pollefeys, Reinhard Koch, and Luc Van Gool. Self-calibration and metric reconstruction in spite of varying unknown internal camera parameters. In *Proc. Sixth Int. Conf. on Computer Vision*, pages 90–91, 1998.

[36] Sebastien Roy and Ingemar J. Cox. A maximum-flow formulation of the N-camera stereo correspondence problem. In *Proc. Sixth Int. Conf. on Computer Vision*, pages 492–499, 1998.

[37] H. Samet. The quadtree and related hierarchical data structures. *ACM Computing Surveys*, 16:187–260, 1984.

[38] Daniel Scharstein. Stereo vision for view synthesis. In *Proc. Computer Vision and Pattern Recognition Conf.*, pages 852–858, 1996.

[39] Steven M. Seitz and Charles R. Dyer. Complete structure from four point correspondences. In *Proc. Fifth Int. Conf. on Computer Vision*, pages 330–337, 1995.

[40] Steven M. Seitz and Charles R. Dyer. View morphing. In *Proc. SIGGRAPH 96*, pages 21–30, 1996.

[41] Steven M. Seitz and Charles R. Dyer. Photorealistic scene reconstruction by voxel coloring. In *Proc. Computer Vision and Pattern Recognition Conf.*, pages 1067–1073, 1997.

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

[43] J. A. Sethian. *Level Set Methods*. Cambridge University Press, Cambridge, UK, 1996.

[44] Steve Sullivan and Jean Ponce. Automatic model construction, pose estimation, and object recognition from photographs using triangular splines. In *Proc. Sixth Int. Conf. on Computer Vision*, pages 510–516, 1998.

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

[46] Richard Szeliski and Polina Golland. Stereo matching with transparency and matting. In *Proc. Sixth Int. Conf. on Computer Vision*, pages 517–524, 1998.

[47] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: A factorization method. *Int. J. of Computer Vision*, 9(2):137–154, 1992.

[48] R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf cameras and lenses. *IEEE Trans. Robotics and Automation*, 3(4):323–344, 1987.

[49] Lee Westover. Footprint evaluation for volume rendering. In *Proc. SIGGRAPH 90*, pages 367–376, 1990.

[50] C. Lawrence Zitnick and Jon A. Webb. Multi-baseline stereo using surface extraction. Technical Report CMU-CS-96-196, Carnegie Mellon University, Pittsburgh, PA, November 1996.