Efficient Volumetric Reconstruction from Multiple Calibrated Cameras

by

Manish Jethwa

Bachelor of Science
Master of Engineering
Department of Engineering Science
University of Oxford, 1998

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering and Computer Science
at the Massachusetts Institute of Technology

September, 2004

Signature of Author: _____

Department of Electrical Engineering and Computer Science
September 21, 2004

Certified by: _____

Seth Teller
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by: _____

Arthur C. Smith
Professor of Electrical Engineering and Computer Science
Chair, Committee for Graduate Students

Efficient Volumetric Reconstruction from Multiple Calibrated Cameras

by Manish Jethwa

Submitted to the Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degree
of Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

The automatic reconstruction of large scale 3-D models from real images is of significant value to the field of computer vision in the understanding of images. As a consequence, many techniques have emerged to perform scene reconstruction from calibrated images where the position and orientation of the camera are known. Feature based methods using points and lines have enjoyed much success and have been shown to be robust against noise and changing illumination conditions. The models produced by these techniques however, can often appear crude when untextured due to the sparse set of points from which they are created. Other reconstruction methods, such as volumetric techniques, use image pixel intensities rather than features, reconstructing the scene as small volumetric units called voxels. The direct use of pixel values in the images has restricted current methods to operating on scenes with static illumination conditions. Creating a volumetric representation of the scene may also require millions of interdependent voxels which must be efficiently processed. This has limited most techniques to constrained camera locations and small indoor scenes.

The primary goal of this thesis is to perform efficient voxel-based reconstruction of urban environments using a large set of pose-instrumented images. In addition to the 3-D scene reconstruction, the algorithm will also generate estimates of surface reflectance and illumination. Designing an algorithm that operates in a discretized 3-D scene space allows for the recovery of intrinsic scene color and for the integration of visibility constraints, while avoiding the pitfalls of image based feature correspondence. The algorithm demonstrates how in principle it is possible to reduce computational effort over more naive methods. The algorithm is intended to perform the reconstruction of large scale 3-D models from controlled imagery without human intervention.

Thesis Supervisor: Seth Teller, Professor of Electrical Engineering and Computer Science

Acknowledgments

This thesis would not have been possible without the ongoing encouragement, support, and love of my family, especially my wife Andrea for her complete confidence in me.

I would like to thank my thesis advisor Seth Teller for his enthusiasm, patience, understanding, and kindness in helping deal with problems throughout my time at MIT. Thanks to Eric Grimson and Bill Freeman for serving as readers and providing thoughtful comments on the thesis.

I would also like to thank all past and present members of the Computer Graphics Group that I have been fortunate enough to interact with. A special thank you to the administrative staff at the Graphics Lab, Bryt Bradley, Adel Hanna and Neel Master, their willingness to address any type of problem is much appreciated. I would also like to thank Barb Cutler for her comments and feedback.

Thanks also to all the friends I've made at MIT, especially Ronak Bhatt. I also like remember my dear friend Bhuwan Singh who passed away during the writing of this thesis, we miss you Booboo.

In memory of my father
DHIRUBHA T. JETHWA
1947-1999

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The problem of reconstruction in computer vision is one of recovering information about the 3-D world, such as shape and color, from 2-D images. The automatic creation of digital models using reconstruction techniques enables the capture of realism that would be too complex and tedious to model by hand. The models of urban environments in particular have numerous applications in the fields of engineering for simulation, architecture for planning and also for games and movies in the entertainment industry.

The goal of this thesis is to present an efficient reconstruction algorithm that can create 3-D models of urban environments from images with known position and orientation. The algorithm is designed to operate iteratively in a discretized volumetric space and recover both shape and color information about the scene. Once converged, the scene is converted to a more concise surface representation. The algorithm presented in this thesis demonstrates the following characteristics:

- It is fully automatic, requiring only calibrated images as input and returning both volumetric and surface models as output.
- The algorithm is iterative, using shape and color estimates from each iteration as priors for subsequent iterations.
- It does not place constraints on the positions of camera within the scene.
- The algorithm scales linearly in the number of images used for the reconstruction.
- Is is designed to perform reconstruction at multiple resolutions with increasing levels of detail.
- The algorithm is easily parallelizable, with the information from each view processed separately and then combined globally across all views.
- It explicitly deals with illumination variation across images acquired in an outdoor environment.

This thesis contributes novel techniques for:

- The matching of colors across images in a probabilistic framework defined in the CIE-*Luv* color space.
- The detection and factoring out of illumination, resulting in a surface color estimate under canonical illumination where possible.
- Efficient methods of recovering 3-D scene information (shape and color) from multiple images through adaptive sampling and multi-resolution methods.
- Using the recovered volumetric model to estimate depth and surface representations of parts of the scene that can later be textured.

The algorithm is verified through testing on a number of real and synthetic data sets. The majority of real data used is part of the City Scanning Project [76,77].

1.1 City Scanning Project

The City Scanning Project [76] at MIT is aimed at performing Automatic Population of Geospatial Databases or APGD. The overall goal is to create models of urban environments automatically from large numbers of images. These models could then be used to perform interactive navigation (walk-throughs) of the imaged cities. Once created, these models can also be used for planning and simulation purposes.

1.1.1 Image Acquisition

The images for the City Scanning project are acquired through the use of a self contained platform called *Argus* (Figure 1.1). Argus uses a digital camera mounted on a computer controlled pan-and-tilt head. The internal parameters [22] of the camera, such as focal length and aspect ratio, are first estimated using Tsai's method [41,79]. In order to be able to provide image positions and orientations, Argus also carries with it a collection of navigational sensors including Global Positioning System (GPS), Inertial Measurement Units (IMU), odometry sensors and inclinometry. Once the platform is positioned in the scene to be imaged, the camera rotates while its optical center is constrained to a fixed point. The result is a hemispherical tiling of images (known as a *node*) that capture almost all the visual information from a particular view point. Figure 1.2 illustrates one such tiling. The images are acquired in HDR or High-Dynamic Range format allowing for the capture of a much wider range of image intensities. These images are acquired through combination of images at various exposures and storing a greater range of values by using floating point exponents to capture both the minimum and maximum intensity values in the scene. The time and date for each image is also recorded in the form of a time stamp [76]. The distances of neighboring nodes (or camera base-lines) are on the order of 10 meters. In totality, a dataset consists of thousands of nodes and covers almost a square kilometer. Figure 1.3 shows a subset of the nodes as panoramic mosaics.

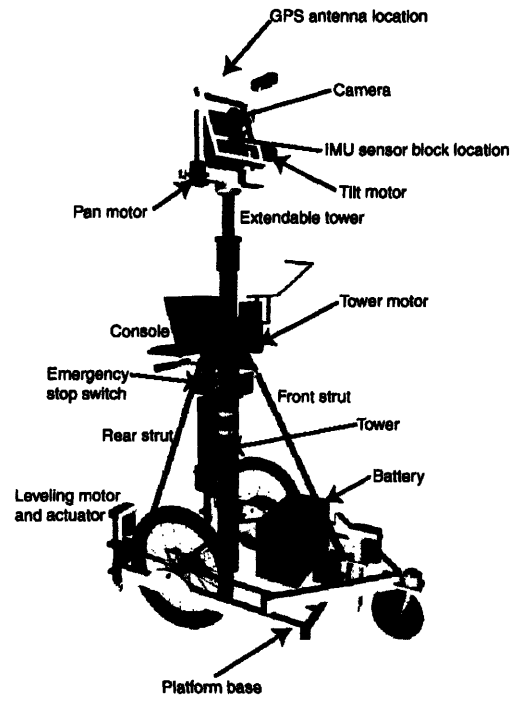


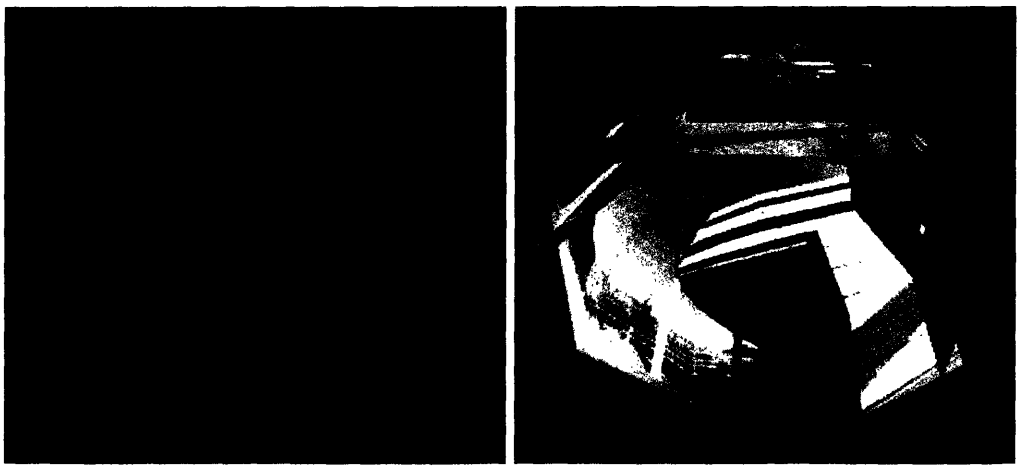
Figure 1.1. Argus: The self contained platform used to acquire images for the City Scanning Project. The platform is equipped with GPS, IMU, odometry and inclinometry.

■ 1.1.2 Image Registration

Despite the number of sensors on-board Argus, the errors in the camera positions and orientations obtained after sensor fusion are still significant. While the accuracies of these measurements are insufficient to be directly used for scene recovery, they can be used as initial estimates to various refinement processes [2, 14] in order to recover pose estimates with greater accuracy.

One method of improving the accuracy of camera positions is described by Coorg [14]. Pose refinements are computed by selecting corresponding features such as corners across multiple images. Feature are projected out into 3-D as ray and the image of these rays in the other views is known as an epipolar line. The node positions are then iteratively adjusted through incremental translations and rotations to minimize the epipolar error (the distance from points to their epipolar lines) for the selected features.

A different method for pose refinement is given by Antone [2]. Here, detected edges in the image are projected out to infinity in order to identify salient vanishing points in the images. Images are then rotated to align vanishing points and therefore recover the relative orientation between images. Once all cameras have been correctly rotated, relative translations between nodes are estimated using a probabilistic formulation that



Tiling Outline Tiled Images

Figure 1.2. An example of the hemispherical tiling of images from the City Scanning Project.

Figure 1.3. A view of a subset of nodes collected as part of the City Scanning Project dataset. In totality, it consists of over a thousand nodes.

statistically obtains feature matches across images without explicit correspondences. Either one or both of these methods can be used to recover pose estimates for the cameras that are sufficiently accurate to be used in the reconstruction process [14].

1.1.3 Scene Reconstruction

The data from the City Scanning data set presents many challenges. The small errors that are present in the calibration of the cameras can lead to more significant errors in the final reconstruction. In addition to these errors, the images are also captured in an uncontrolled environment in the presence of varying illumination from the sun and sky. In addition to the objects we wish to reconstruct in the scene, such as the buildings, the images also contain other objects such as trees and transient objects such as people and cars. Trees often prove to be difficult objects to reconstruct due to their fine complex structure [70]. Transient objects cause problems because they create moving occlusions that cannot be captured if the scene is assumed to be static. Fortunately, these problems can be alleviated by increasing the number of images and basing the algorithm on consensus estimates where all or part of each object targeted for reconstruction is required to be visible in the majority of the available views.

Once accurate positions and orientations are available, they can be directly used as input together with their respective images to recover 3-D information about the scene. Previous methods has been developed to perform reconstruction on the City Scanning data [13,15,52]. Coorg [13] estimates the positions of facades in the scene by searching for correlations of projected edges found in the images. This method requires that the objects in the scene to be reconstructed consist of straight line edge features that can be detected and matched. In [52], Miller estimates surface patches or surfels through dense intensity based correlation across images. The results presented for synthetic results are promising although the challenges of real images such as noise and image clutter prove to be difficult to handle. Spatial smoothness constraints must also be enforced to removed unwanted outliers. Cutler [15] describes a further reconstruction stage that uses a strategy to combine the surfaces produced by the methods of Coorg and Miller. This fusion of surfaces enables more robust reconstruction through the removal of outliers. Finally, valid surfaces are extended to create faceted models of the scene.

In this thesis, the problem of scene reconstruction is tackled via a different approach. Firstly, no assumptions are made regarding the shape of objects in the scene. Our approach estimates the shape and color of the scene through volumetric based reconstruction using voxels and is therefore suitable for arbitrarily shaped objects. Secondly, large scale illumination variations across images are handled explicitly by estimating the lighting conditions present at the time of capture through direct sampling of the images. The observed colors in the images are adjusted to compensate for the estimated lighting and these new color estimates then used to recompute the shape and color of the scene. In this way, a volumetric model of the scene is iteratively acquired. Finally, since volumetric models are unsuitable for all but a few applications, we describe a method for the derivation of an additional surface representation from the same framework via depth estimation in each image.

■ 1.2 Thesis Overview

An overview of the algorithm is shown schematically in Figure 1.4. The calibrated images that serve as input are shown on the left hand side of the figure. The main module of the algorithm, which performs the volume reconstruction, is shown centrally and consists of three distinct sub-parts. The opacity estimation algorithm forms the basis of volume reconstruction method being used and is described in Chapter 2. It can be shown that recovering the shape and coloring of the scene are coupled problems and that estimating them directly is a hard problem, especially in the general case where lighting conditions are also allowed to vary across images. Our approach uses an iterative method to estimate the volume probabilistically from the information available in the images.

In Chapter 3, we describe the illumination estimation module which tackles this issue by modeling outdoor illumination present in the scene at the time of capture directly from the portions of image that contain sun and sky. The acquired sky models are used to estimate large scale lighting variations that occur across images, and together with the opacity estimates, can then be used to estimate the scene color as described in Chapter 4. A novel probabilistic color matching technique is presented that enables the matching of colors across images in the presence of lighting variations. This problem of matching colors across images is presented in relation to work on color constancy together with an investigation on the options of suitable color-spaces. Results are presented for testing the color matching technique on both real and synthetic data.

Various optimizations to the volume reconstruction algorithm are presented in Chapter 5 as modifications to the basic algorithm in Chapter 2. These modifications are made in order to reduce the algorithmic complexity, in both time and space, through methods such as adaptive sampling in the images and projected rays, lazy voxel allocation, multi-resolution reconstruction and a description of how the algorithm is designed using a Message Passing Interface (MPI) such that the computational work can be distributed over multiple machines and processed in parallel.

Voxel representations of complex scenes can often be memory intensive and also lack the visual appeal of more concise surface representations. In Chapter 6, we address this problem by demonstrating how surfaces can be estimated directly using the same algorithmic framework. We first describe a method for extracting dense depth estimates and associated variances from the recovered volumetric model. The uncertainties in these depth estimates are then minimized through the use of Bayesian Belief Propagation in the images.

In Chapter 7, we present the implementation details of the entire algorithm, including a step-by-step look at the algorithm from initialization to final output model, demonstrating how the modules from the previous chapters are integrated. The results of executing the algorithm on a number of synthetic and real datasets are presented in Chapter 8. In Chapter 9 we conclude with a summary of the overall contributions of this work, a discussion of limitations of the algorithm, and future research directions.

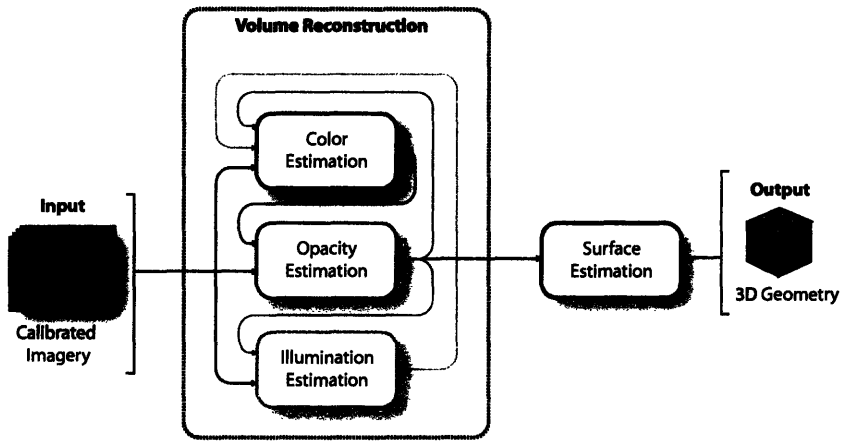


Figure 1.4. System Overview: A schematic diagram showing an overview of the reconstruction algorithm: from calibrated images (left), to 3-D model (right).

■ 1.2.1 Terminology

A list of terminology used throughout this thesis is given below.

- **External Camera Parameters:** The position and orientation of the camera in some absolute coordinate system at the time of image capture. The position of the camera is also known as the optical center and is defined as the point of convergence for all rays entering the camera.
- **Internal Camera Parameters:** Properties of the camera such as principal point, focal length, aspect ratio and skew. These depend on the camera model being used. In this thesis, we assume the pinhole camera model.
- **Calibrated Imagery:** Images with known internal and external parameters.
- **Node:** A collection of images acquired while rotating the camera but keeping the optical center fixed. The images typically tile a hemisphere.
- **Voxel:** A unit of volume used in the reconstruction. The reconstruction volume typically contains over a million voxels.
- **Observation:** The point in an image to which a point in the 3-D world is projected.
- **High Dynamic Range:** Image format allowing the encoding of greater range of values than traditional formats.

1.3 Summary

The general problem of recovering 3-D scene information from images is a difficult one. In this thesis, we present a reconstruction algorithm based on probability that aims to estimate both shape and color from calibrated imagery. Large scale lighting variations are also estimated for outdoor scenes and used to adjust image colors for improved coloring of the final reconstruction. The algorithm is designed as part of the City Scanning Project that aims to recover 3-D models of urban environments from thousands of calibrated images. The following Chapter presents an overview of previous related work in 3-D reconstruction together with a detailed description of the volume reconstruction method adopted as the basis for our system.

Chapter 2

Related Work

In this chapter, we describe the basic approach adopted to perform volume reconstruction. We begin our discussion with a review of previous work on the more general problem of recovering 3-D geometry from images. Particular attention is given to the application of volumetric methods to solve the recovery problem. We describe in detail one such algorithm which forms the basis of our system.

2.1 Background

The general problem of 3-D shape recovery (or reconstruction) from images is still largely unsolved and remains a difficult hurdle in the field of computer vision. When imaging a scene, much of the 3-D information regarding the scene is lost. Its recovery is an ill-posed problem since an infinite number of scene configurations can lead to the same image. In order to make this problem more tractable, many researchers have made simplifying assumptions regarding the scene. An overview of previous research and respective assumptions in the field of 3-D scene reconstruction is presented.

2.1.1 Photogrammetry

This field primarily deals with the process of measuring topographical geometry through the use of aerial photographs [71,83]. As this area of 3-D reconstruction predates digital computers, analogue techniques were used to physically reproject images using devices such as a stereoscopic viewer. Many existing techniques require substantial human input in the form of identifying corresponding points in various views of a scene. More recently however, automatic techniques are emerging [29].

2.1.2 Stereo Vision

These techniques recover 3-D shape using a minimum of two images. The process usually involves the matching of corresponding *features* (usually points or lines) across images and recovering 3-D positions through triangulation [6,20]. The points and lines can be found automatically using image processing techniques such as corner [30] and edge [10] detection respectively. While this technique works fairly well for images taken from nearby camera positions, it tends to be less effective on general image pairs due

to the inherent tradeoff between the inter-camera distance (*baseline*) and matching ability. Images produced with small baselines are easy to correspond since the images (and therefore features) appear very similar, however the 3-D estimates obtained by triangulation from small baselines can be unreliable. Wide baseline matching provides accurate 3-D information, but matching across disparate images proves to be extremely difficult [64] since the images are likely to differ increasingly with larger baselines. Multi-baseline stereo algorithms [7,56] attempt to overcome this drawback by using the information present in many images simultaneously. Methods that use omni-directional stereo [35] exploit the wide field of view to easily match features along baselines while obtaining accurate 3-D depth estimates perpendicular to the baseline. In this thesis, we make use of a large number of omni-directional images with varying baselines for ease of matching and improved accuracy from triangulation.

■ 2.1.3 Shape From Shading

Shading information derived from brightness variations across an image can be used to deduce an object's shape from a small number of images [6, 31]. While this approach has been fairly successful in recovering shape from images acquired in tightly controlled environments (i.e., small, uniformly-colored objects; fixed and known lighting), complex scenes such as those found in an outdoor environment in which texture, reflection, occlusion, and shadows are present prove to be too difficult. Recovering 3-D scene properties from a single image has therefore only enjoyed limited success, thereby encouraging the use of multiple images to accomplish the same task.

■ 2.1.4 Structure from Motion

One of the major obstacles for vision algorithms in recovering 3-D information lies in finding corresponding features (points and lines) across images. These *correspondences* can then be used to recover camera pose and the same features projected out and triangulated to obtain 3-d structure. An alternative approach suggests tracking features across an image sequence (video frames) and using the constraints generated to recover both 3-D shape and camera information. Algorithms in this class can be classified into two types: *batch* methods that perform reconstruction using all the constraints at once, and *on-line* methods that incorporate a single image (the next image in the sequence) into an existing description of the 3-D scene. Online or recursive methods [3] have the advantage that they can be used in active vision (e.g., the navigation of mobile robots), where images are only available incrementally. However, as they depend on the first few images to initialize the algorithm, they tend to be less accurate than batch methods where the entire set of images can be analyzed collectively. One of the more successful applications of the batch technique has been in the special case of orthographic projection [78] where, due to the linearity of the constraints, standard techniques like singular value decomposition [63] can be used to solve constraints imposed by tracking. However, it has had only a limited practical impact due to the orthographic assumption together with the difficulties in dealing with occlusion. More general algorithms [19, 53, 73, 75]

that address perspective projection have also been described.

There are a few disadvantages to this approach for recovery of 3-D information. First, since both structure and motion are to be recovered from a sequence of images, it is essential that a large set of features be tracked reliably in order to produce enough constraints to robustly solve the problem. In practice, this implies that images in a sequence need to be spaced very closely (small baselines) so that tracking succeeds in corresponding many features across the image stream. Unfortunately, the cost of acquiring such an image sequence becomes prohibitive if large-scale models are being reconstructed. In sequences where the baselines are small relative to the size of the scene, more images are required to improve the accuracy of the reconstruction since good triangulation requires large baselines. Another disadvantage of these techniques is that they usually only provide *sparse* 3-D information which can be difficult to convert to a complete 3-D representation (such as a CAD model) suitable for computer graphics rendering. These features are therefore usually only used as vertices of a piecewise planar surface to create a simple, crude model of the scene.

■ 2.1.5 Image-Based Rendering

These approaches [11,28,42,51] circumvent the process of 3-D reconstruction; instead, multiple images are used to produce an image from a novel viewpoint via *interpolation.* In addition to avoiding the 3-D reconstruction problem, image based rendering also has the advantage of producing novel images independently of the geometric complexity of the scene, with occlusion events being handled through the intelligent combination of multiple images. Image-based rendering systems do not however, generate an editable 3-D representation (such as a CAD model). This can be a disadvantage in many applications, such as when a designer experiments with geometry and/or lighting conditions or applies simulation techniques like collision detection. While some of this flexibility is provided in a hybrid approach [17], it does require significant user input to perform the recovery of shape.

■ 2.1.6 Shape from Silhouette

Also known as Visual Hull methods [48–50], shape from silhouette approaches are very efficient in obtaining models from sequences in which the object to be modeled has been successfully segmented from the rest of the scene. The boundary or silhouette of the object is detected in each image as a series of connected edge lines. These lines are then projected into the 3-D world and intersected with those from other images to reconstruct a volumetric model through the intersection of these visual hulls. The method is both fast and efficient, permitting the reconstruction of objects in real-time. This method does however require prior knowledge of the object or background so that it can be effectively segmented in each image. If both the object and background vary considerably, automatic segmentation becomes difficult and may need to be done manually. There has also been work to extend this method to multiple segmented objects [48].

■ 2.1.7 Photo-consistency Methods

These methods involve representing a 3-D scene as a collection of finite volumetric elements (voxels) [21,36,67]. These voxels are processed in an order which accounts for visibility constraints as determined by the camera locations. Each voxel is then colored or culled depending on the outcome of some consistency criterion. If the projection of the voxel in the cameras is consistent in color over many images then the voxel is deemed photo-consistent and therefore occupied by a mass of the consistent color. If however the projections are inconsistent, the volume occupied by the voxel is thought to be empty and is removed. In this manner, inconsistent voxels are thereby *carved* away leaving the remaining voxels that are consistent with the provided views to describe required 3-D scene. The culling of inconsistent voxels during the carving process can however incorrectly remove occupied volume elements resulting in holes in the final model [65]. Methods based on photo-consistency can often perform reconstruction from cameras with much wider baselines than feature based methods. Background subtraction is often performed on the images segmenting foreground objects from the background thereby reducing the worst case reconstruction to that extracted using shape from silhouette methods. For most real images, it can often be difficult to perform background subtraction without significant human intervention, therefore rendering these methods unusable for fully automatic reconstruction. Some methods also force only foreground reconstruction by requiring that the unwanted background change significantly in each image, therefore failing the photo-consistency check and accordingly removed.

A different approach to volume reconstruction is presented in [18] and is called the Responsibility weighted voxels method or *Roxels* algorithm. This algorithm is based on the cooperative stereo method [45] and describes a unified framework for volumetric reconstruction. The method aims at providing the most general algorithm for scene recovery by allowing for the reconstruction of partially occupied and translucent voxels as opposed to the binary voxel occupancies and opacities required by other methods.

■ 2.2 Modeling Image Formation

In order to facilitate the discussion of volume reconstruction, we begin with a brief overview of the image formation model we are using. Without loss of generality, the appearance of any image is dominated by three main factors:

- scene illumination,
- physical scene content,
- camera properties.

Scene Illumination plays an important role in image formation as even small variations in illumination may lead to images that appear very different. To overcome this problem, many existing techniques presume that the lighting conditions remain fixed

while the scene is imaged. This allows the reconstruction of a scene under specific lighting conditions while ignoring the actual interaction between surface and illumination. Further discussion on this interaction can be found in Chapter 3.

Physical Scene Content is a description of the objects that make up the composition of the scene. The aim of 3-D volume reconstruction is to recover this description such that it can then be interacted with and viewed from novel viewpoints in addition to those from which it was originally imaged. There are many ways in which it is possible to represent these objects, the selection of which is a precursor to deciding which method is best used to reconstruct the scene. It is often convenient to represent the objects as a set of surfaces or alternatively points that lie on the surface that may later be interpolated to obtain a complete representation. The convenience of this representation comes from the face that it is both concise and intuitive. Many computer vision techniques exist for surface extraction from sets of calibrated images [22,23,44,53,59]. For some applications however, the surface representation may not be adequate (such as representing medical data) making volumetric representations more suitable for describing the objects.

Camera Properties include external parameters such as pose information (position and orientation) and internal parameters such as focal length and aspect ratio of the camera. The pin-hole camera model has been widely adopted by the vision community. The model is linear and assumes that light rays passing through the lens converge at a single point known as the optical center or center of projection. This model assumes that no non-linear distortion occurs during the transition of light through the lens. For real cameras, this is not always the case but serves as a good approximation for most problems. In cases where the non-linear distortion is too great to be ignored, techniques allow these distortions to be corrected or at least minimized such that the pin-hole camera model is still usable. Many techniques exist to estimate external parameters (position and orientation) through the use of sensors and/or correspondences across images [22]. Internal parameters can be estimated through the use of a calibration grid and additional correspondence information. Here, for the purposes of volume reconstruction, we assume that both internal and external parameters for the cameras are available, having been estimated in advance, and can therefore be described as producing calibrated imagery.

The interactions between illumination, scene and camera are highly complex and cannot be modeled completely. In practice, simplifications are made to make the problem more tractable. The complexity that can exist in the scene alone is infinite and scale must therefore be chosen to limit the level of detail in the reconstructed scene. In the case of volumetric reconstruction, this is most easily done through the choice of voxel resolution. In order to understand this interaction in more detail, we now present a simple description of image synthesis from volumetric models.

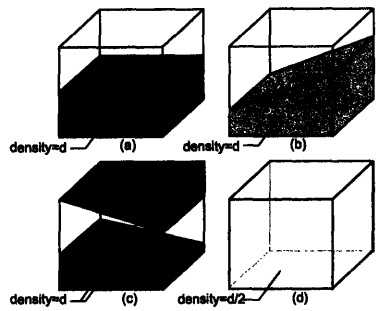


Figure 2.1. Voxel Opacity/Occupancy: All of the above voxels have occupancy values equal to 0.5 since the mass in (d) is half as dense as those in (a), (b) and (c). If the mass is thought to be uniformly distributed, the occupancy indicates how densely a voxel is occupied.

■ 2.2.1 Modeling the Scene with Voxels

Let us start with a view of the world consisting of many (possibly millions) discrete volumetric units called voxels. A voxel at position (x, y, z) has an associated opacity $\alpha(x, y, z)$ and color $c(x, y, z)$. If the color of the incident light on the voxel is c_{in}, the observed color of light c_{obs} after passing through the voxel, as defined in [18], is:

$$c_{obs} = \alpha(x, y, z)c(x, y, z) + (1 - \alpha(x, y, z))c_{in}.$$

From this relation, we see that for a completely transparent voxel ($\alpha(x, y, z) = 0$) the observed color is exactly the incident color. Conversely, for an opaque voxel ($\alpha(x, y, z) = 1$), the color is dependent only on the voxel color $c(x, y, z)$. For translucent (partially opaque) voxels, the color is a combination of the incident color and voxel color, each weighted according to the voxel opacity. As the ray passes through the voxel on its way to the camera, the frequency of the light (observed color) changes. This analysis is still a significant simplification of the true interaction since the effects of reflectance are largely ignored. A typical assumption of most volume reconstruction algorithms is that the volume is either completely transparent or completely opaque. This assumption is violated when the scene contains objects such as colored glass. Even in the absence of translucent objects, there exists a need to account for partially opaque voxels. Since our volume is represented using discrete volumetric units, a binary set of opacities can lead to aliasing artifacts [74]. By allowing all opacities in the range [0..1], we can in effect perform anti-aliasing on the 3-D volume and account for all partly occupied voxels and also translucent voxels such as those shown in Figure 2.1.

Let C_j be the internal camera matrix for the the jth image defined as

$$C_j = \begin{pmatrix} s_x & 0 & 0 \\ tan\theta & s_y & 0 \\ x_0 & y_0 & 1/f \end{pmatrix}$$

where s_x and s_y are the number of pixels in the x and y directions respectively, θ is

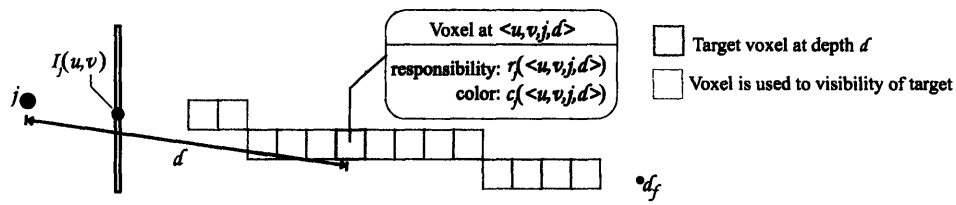


Figure 2.2. The relationship between voxel colors and the observed colors in the image.

the skew angle (the angle between the x and y axes minus $\frac{\pi}{2}$) and the point (x_0, y_0) describes the principal point or the point on the image plane closet to the optical center of the camera. The distance between the principal point and the optical center is the focal length f. If the camera is positioned at the point $\mathcal{T}_j$ in the global coordinate system and orientation described by the rotation matrix $\mathcal{R}_j$, the camera is completely described by the 3×4 projection matrix $\mathcal{P}_j$ such that:

$$\mathcal{P}_j = \mathcal{C}_j[\mathcal{R}_j|\mathcal{T}_j]$$

which maps points in the 3-D world X to the point in the 2D image x. Adopting the notation in [18], the projection operation for a point at coordinates (u, v) in image j, a distance d from the optical center of the camera to the voxel center at (x, y, z) in the 3-D world is:

$$< u, v, j, d >= \mathcal{R}_j^{-1}(\mathcal{C}_j^{-1}(u, v, d)^\top - \mathcal{T}_j) = (x, y, z)$$

Visibility: The visibility $\omega(< u, v, j, d >)$ of a voxel at $< u, v, j, d >$ in camera j depends only the voxels along the same projected ray, at locations $< u, v, j, l >$ where $l < d$ such that:

$$\omega(< u, v, j, d >) = (1 - \prod_{l<d} \alpha(u, v, j, l))$$

where $\alpha(< u, v, j, l >)$ denotes the opacity at location $< u, v, j, l >$.

This visibility can then be used to compute the contribution of voxel at coordinates $< u, v, j, d >$ to the observation in image j. This contribution is defined as the responsibility in [18] and is defined as follows.

Responsibility: The responsibility $r(< u, v, j, d >)$ is a quantity that defines the contribution of a voxel to the observation in an image. The quantity is defined as the product of the voxel opacity and its visibility in an image such that

$$r(< u, v, j, d >) = \alpha(< u, v, j, d >)\omega(< u, v, j, d >).$$

Figure 2.2 shows the relationship between voxel colors and the observed colors in the image. If $c(< u, v, d, j >)$ defines the voxel color, the observed color at coordinates (u, v) in the image can be computed by the voxel colors along the ray weighted by its respective responsibility

$$I_j(u,v) = \prod_{d=0}^{d_f} r(< u, v, j, d >) c(< u, v, j, d >)$$

where d_f is the far distance associated with the rays from the camera. The distance d_f is chosen such that the each camera interacts with the entire reconstruction volume. Alternatively, for large scale reconstruction, the influence of each camera can be limited by setting the far distance. This relation can be rewritten in matrix notation to describe the observations in all images simultaneously using all voxel colors and responsibilities in the volume. The complete set of image observations can be written as a stacked vector $\mathbf{I}$ of length equal to pixels per image times the number of images. The voxel colors can also be written as a vector $\mathbf{C}$, whose length is equal to the number of voxels in the volume. They are related via a responsibility matrix $\mathbf{R}$.

$$\mathbf{I} = \mathbf{RC}$$

The matrix $\mathbf{R}$ can clearly be very large. Even for small inputs of a few low resolution images and a volume that is only 20^3 voxels, the matrix can contain millions of entries. In practice however, $\mathbf{R}$ is typically very sparse.

This description of the imaging process is the unified framework for voxel reconstruction as presented in [18]. Using this notation, we move to recovering the matrix $\mathbf{R}$ and vector $\mathbf{C}$ from the images $\mathbf{I}$.

2.3 Scene Reconstruction

The problem now becomes one of determining the matrix $\mathbf{R}$ and vector $\mathbf{C}$ that satisfy the relation. If we can identify the responsibility matrix, then the problem of computing $\mathbf{C}$ only requires the inversion of $\mathbf{R}$. This is clearly a difficult problem due to the interdependencies within the responsibility matrix itself. These dependencies are directly related to the visibility constraints where the opacity of each voxel may depend on those of others in the volume. The loss of information during the imaging process results in an ill-posed problem since multiple scene shapes may be consistent with a single set of images.

2.3.1 Direct Methods

In theory, it is possible to formulate the problem as one of error minimization and directly estimate R and C. In practice however, this is a hard problem since R is highly non-linear, and due to its size can contain a vast number of parameters to solve.

Direct estimation of the responsibility matrix and voxel color vector is difficult due to the interdependencies of visibility constraints. Many techniques have resorted to simplifying assumptions in order to estimate these quantities. For example, Seitz and Dyer [68] provide an efficient solution by making the assumption that the cameras are configured in a way such that the voxel volume can by traversed in an order according to visibility. To make this possible, no voxel must lie inside the convex hull of the camera locations. A plane can then be swept through the volume and the voxels on that plane are tested one at a time and removed if they appear to be inconsistent in the cameras that observe them. The voxels are also assumed to be either fully opaque or transparent, further simplifying the computation. The algorithm is relatively fast requiring just a single pass through the volume along each axis. The reconstruction volume and resolution are also chosen apriori allowing for a tight bounding box to be created around the target volume. The algorithm also requires either a changing background such that is fails the consistency checks between images therefore preventing its reconstruction or it must be subtracted out from the foreground further aiding the reconstruction process.

Once the entries in the responsibility matrix (voxel opacities) have been computed, its inversion is relatively easy and the voxel color is arrived at by simply taking average of all pixels that directly observe that voxel.

$$\mathbf{C} = \mathbf{R}^{-1}\mathbf{I}$$

In the case of general camera placement however, the problem is more difficult and there is no ordering of the voxels that preserves the visibility constraints. This is due to a mutual dependence of voxel opacities and the only way to overcome this problem is to introduce an iterative approach to estimating the shape of the scene volume and its respective color.

■ 2.3.2 Iterative Estimation

When the cameras are arranged in an arbitrary configuration, the opacity computation of each voxel is mutually dependent on that of other voxels in the volume. Direct approaches are therefore unable to solve the problem analytically and iterative methods must be used instead. One such method described in [36] and is based on the theory of space carving [67]. The coloring of voxels is performed through multiple passes of sweeping planes through the voxel volume, improving the voxel representation at each iteration. This method makes a binary decision at each voxel based on the outcome of a consistency check. This check examines the projection of the voxel into the images and compares image colors in order to make the decision. The underlying assumption is that the surfaces are observed with similar characteristics (brightness and color) in every view. This is true if the illumination is held fixed and the surface is non-specular.

Another iterative algorithm that attempts to tackle the more general reconstruction process is presented in [18]. The Roxels approach provides the ability to reconstruct translucent objects and centers the process around projecting information from the

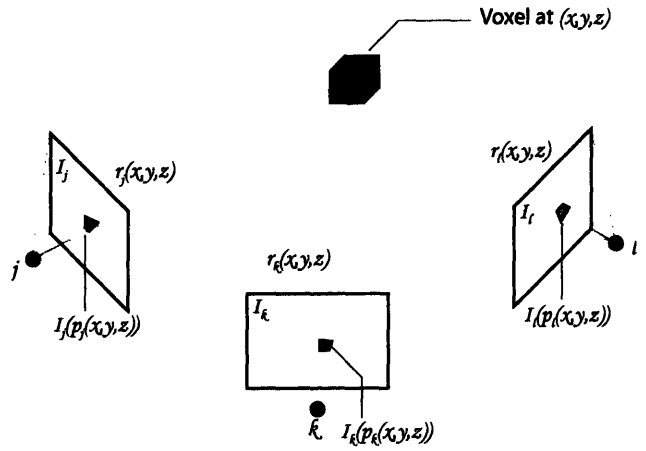


Figure 2.3. Computing the voxel color by weighting the image colors by the view dependent responsibility.

images into the voxel volume. As such, the algorithm can be viewed as a true inversion of the imaging process. The algorithm involves several stages that are executed at each iteration. These stages are described here for convenience:

Step 1: Estimating the Voxel Color Vector C

If the responsibility matrix R is sparse, it is possible to estimate its inverse, and the color $c(x, y, z)$ of voxel at (x, y, z) can be estimated using the following relation:

$$c(x,y,z) = \sum_j r_j(x,y,z) I_j(p_j(x,y,z)) / \sum_j r_j(x,y,z)$$

where the responsibility $r_j(x, y, z)$ for the voxel at 3-D coordinates (x, y, z), in the jth image I_j, therefore weights the color contribution of all observations in $I_j(p_j(x, y, z))$.

The function $p_j(x, y, z)$ projects a voxel at (x, y, z) into the coordinates (u, v) of image I_j via the projection matrix $\mathcal{P}_j$. This projection could be just the voxel center leading to a single pixel in the image. In practice however, the entire volume is projected into the image (via vertex projection or otherwise) and $I_j(p_j(x, y, z))$ is therefore represents by a region containing several pixels. Some statistic (such as the mean) of these pixel values can then be used to determine the color of the voxel. The algorithm is initialized such that $r_j(x, y, z) = 1 \; \forall \; j$, indicating that each voxel is fully responsible for the observation in the image. The voxel color can therefore be computed as the average over all possible observations. Since occlusions are likely to exist in the scene, this average provides only an initial estimate for the color to be improved during subsequent iterations as better estimates for the responsibility $r_j(x, y, z)$ became available. Figure 2.3 depicts the relation between voxel color and the colors in the region of projection in the images.

Step 2: Computation of Agreements

Once an estimate of the voxel color $c(< u, v, j, d >)$ is available, it can be compared to the pixel values in image $I_j(u, v)$ in order to compute a view specific agreement $a_j(u, v, d)$. The agreement, as its name implies, is a measure of similarity and in this case is a function of two color distributions.

$$a_j(u, v, d) = agreement(c(< u, v, d, j >), I_j(u, v))$$

The actually method by which the agreement is computed in discussed in detail in Section 2.3.2, but has the characteristics that $a_j(u, v, d)$ is large when there is a strong correlation between colors and small otherwise. In practice, the agreement is normalized to be in the range [0...1]. These agreement values are then used to compute the opacity estimates for the voxels where the higher the agreement, the more likely the voxel is opaque and therefore is responsible for the observation in the image. Similarities between the observed colors in the image and the voxel color could however be due to false matches and therefore may not directly indicate the opacity of a voxel.

Step 3: Computing View Dependant Responsibilities

In order to determine which voxels are responsible for a particular view, the agreements are normalized along each observation ray in the image. As each ray is projected from the image, the agreements for the voxels that lie on that ray are summed. Voxel colors that highly agree with the observation color are likely to be responsible for that observation. The view dependent responsibility $r_j(< u, v, j, d >)$ can therefore be estimated directly from the agreement values:

$$r_j(< u, v, j, d >) = \frac{a_j(u, v, d)}{\sum_{l=0}^{d_f} a_j(u, v, l)}$$

Step 4: Computing Local Opacities

These responsibility estimates can then be used to directly compute the view dependent opacities $\alpha_j(< u, v, j, d >)$ for each voxel. This is simply a rewrite of the visibility equation given previously and is given by:

$$\alpha_j(< u, v, j, d >) = \frac{r_j(< u, v, j, d >)}{1 - \sum_{l<d} r_j(< u, v, j, l >)}$$

Step 4: Computing Global Opacities

The view dependent opacity estimates can be combined to form a single globally consistent opacity estimate $\alpha(x, y, z)$, and is the weighted average over all views according to the responsibility.

$$\alpha(x,y,z) = \frac{\sum_k r_k(x,y,z)\alpha_k(x,y,z)}{\sum_j r_j(x,y,z)}$$

Step 5: Estimating Global Responsibilities

The final step in each iteration involves using the global opacity estimate to re-compute the responsibility values. These responsibilities will also be globally consistent by combining the information available from every image. The global responsibility for each voxel is computed according to its visibility in a particular view, such that:

$$r'_j(< u,v,j,d >) = \alpha(< u,v,j,d >)(1 - \prod_{l<d} \alpha(< u,v,j,d >))$$

These computations collectively define a single iteration of the algorithm. The computed global responsibilities can then be used in the next iteration and the process repeated until the global opacities converge.

This Roxels algorithm forms the basis for our reconstruction process. The algorithm is a general reconstruction algorithm allowing for the potential reconstruction of translucent objects. The algorithm retains the ability to reconstruct objects despite uncertainty in the input data since it does not differentiate between transparency and uncertainty. Indeed, if the objects in the scene are known to be completely opaque, the iterative algorithm proposed by Kutulakos et al. in [36] could also be used but would produce unreliable results for partially occupied voxels. The effectiveness of all volume reconstruction methods pivot on the agreement computation (sometime called a consistency check) described above which makes use of the underlying assumption that surfaces in the scene are observed through similar color values across images. This is only true when no lighting variation exists between images. We present extensions to the Roxels algorithm by providing improved color agreement computation and the ability to deal with illumination variation in outdoor image sequences.

■ 2.4 Summary

This chapter has provided an overview of related research for 3-D object and scene reconstruction from images including feature based structure from motion and volumetric methods. We have presented a model for image formation in a voxelized world and then methods for inverting the process to recover the scene description from the images. We have discussed the opacity estimation part of the algorithm and showed that despite the loss of information during the imaging process, it is possible to recover scene properties by either making simplifying assumptions or iteratively estimating the shape and color of the volume. In the next chapter, we look at the estimation of illumination conditions from a series of images and later, in Chapter 4, show how we can combine the voxel color estimation and illumination estimates to obtain better color matching across images.

Chapter 3

Illumination Estimation

This chapter explores the automatic estimation of outdoor illumination conditions directly from hemispherically tiled, high dynamic range images. For simplicity, most algorithms [18,36,67] make the assumption that the lighting in the scene remains constant during imaging. This is often true for sequences acquired under controlled conditions. In the real world however, the lighting can vary considerably over time. Figure 3.1 shows images of the same building façade under different lighting conditions. For volumetric reconstruction methods, the estimation of scene shape and color are coupled problems. The changes in observable colors must therefore be suitably accounted for in order to perform robust shape estimation for real world scenes.

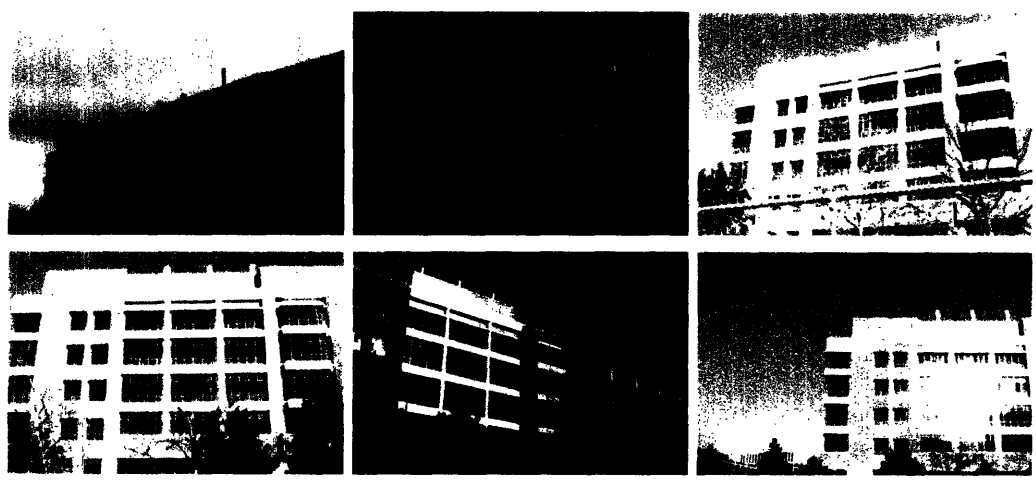


Figure 3.1. The appearance of the same building façade in various images can change considerably according to the outdoor illumination and weather conditions.

Lighting plays a major role in all forms of imagery such as photography and video. Small changes in lighting conditions can dramatically change the way in which objects may appear. These changes can be due to a variety of reasons such as surface properties or shadowing. The ability to estimate the illumination conditions can be invaluable when trying to understand the content of an image. Once the illumination is known,

it can be used to augment the scene with new objects, each with the correct casting of shadows etc. Known illumination conditions are also required when performing shape from shading algorithms. In the case of reconstruction, since these conditions can be so varied and complex, most reconstruction algorithms make the assumption that illumination in the scene remains fixed during imaging. For this assumption to hold, images are usually captured in a controlled indoor environment. Colors and brightness values of objects in the scene then remain consistent across images irrespective of camera position. These consistencies can then used to guide the reconstruction process.

In the absence of these controlled conditions, where the assumption of fixed lighting does not hold, the problem is considerably more difficult. For example, images captured outdoors during the day are predominantly illuminated by a constantly changing light source, the sun. To perform reconstruction under these conditions, several methods have been developed to overcome the changes due to illumination. One popular strategy is to look for properties or features of the scene whose appearance in the image are invariant to changes in illumination and use these features to perform the reconstruction. These features are usually points or lines in the image corresponding to corners and edges in the scene. These features can be detected using any of the various detectors [10,30]. In addition to finding lighting invariant features in images, researchers have also attempted to directly estimate the lighting condition present at the time the image was taken. These conditions can then be used to normalize observed colors in image leading to more robust matching across images.

■ 3.1 Background

There has been much research to estimate the illumination conditions from one or more images. Most images do not contain an image of the light source itself since its brightness would most likely dominate others, leading to poor contrast in the rest of the image. For this reason, either the light source or the camera is placed such that it allows light to be reflected off of the scene and directly enter the camera. Estimating the illumination conditions from such images can be difficult. When estimating the illumination from a single image, information about the illuminant is acquired directly from the observed scene. The scene geometry is therefore assumed to be known [85] and the shapes of shadows can be used to defer the positions of one or more light sources.

If the light source(s) can be successfully captured in the image, then the problem of estimation is simplified. Most illumination estimation algorithms require the placement of a calibration object of known shape in the scene such as a highly specular (mirrored) sphere. The object, known as a light-probe, also reflects other nearby objects into the camera. The problem of the reflected light source saturating the image is solved through the use of high dynamic range imagery. These images allow for the capture of the scene with several orders of magnitude difference between the brightest and darkest part of the scene.

When calibration objects are not available in the scene, panoramas created via

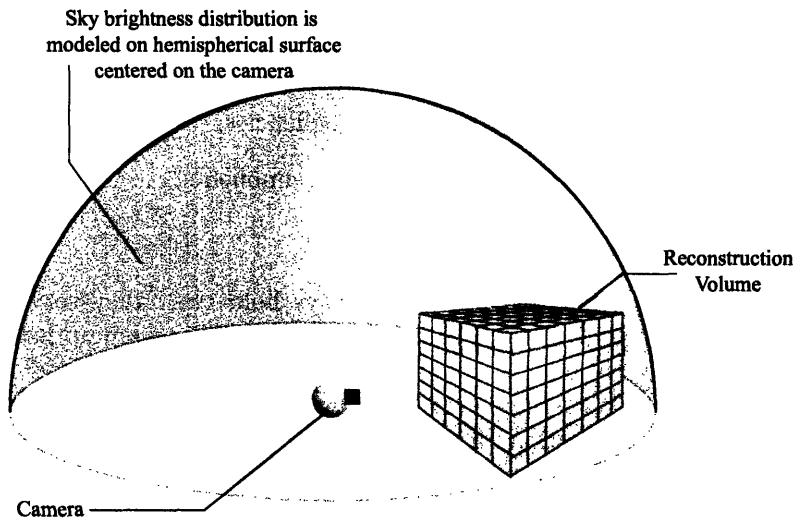


Figure 3.2. Sky Model: The sky is modeled as a hemispherical dome of very large radius which encompasses the entire reconstruction volume.

mosaiced images can be used to the same effect. Images are acquired while the camera is rotated, keeping the optical center fixed. The images can then be tiled together to form a hemispherical panorama. Visible light sources can then be detected directly in the images and used to estimate the lighting. In [54, 61, 85], this approach is adopted for outdoor scenes, where the sun and sky can be considered the dominant illuminants. The regions that correspond to sun and sky are manually selected [85]. The brightness values are then used together with a generalized sky model to estimate the entire sky from these selected regions. Although the results are impressive, the extension to large sets of images is prohibited by the manual selection process. We present an algorithm for the automatic detection and estimation of the illumination provided by the sun and sky in outdoor high dynamic range images.

■ 3.2 Modeling Outdoor Illumination

It is plausible to assume that the most prominent illumination during the day in an outdoor environment is provided by the sun and sky [85]. Since both sun and sky are much further away then the rest of the surroundings, they can be accurately modeled by a brightness distribution on a hemispherical dome (See Figure 3.2). The dome is placed at the optical center of the camera and encompasses the entire scene.

With this physical description of the sky, the brightness distribution can be estimated by projecting intensities from the images onto the hemisphere. Knowing which portions of the images belong to sky and those that correspond to the scene is necessary to perform this projection. This process of segmentation is performed manually in [85].

The automatic detection of portions of sky in the images is possible however through the use of the following observations about the sky:

- Homogeneous, with mostly low spatial frequencies,
- Blue, with particular temporal frequency components,
- Bright, with high luminance values.

Although these conditions do not not always hold true, they serve as an initial conservative estimate as to the appearance of the sky. The assumption that the sky is homogeneous is clearly violated in the presence of clouds and other objects in the sky. However, if the sky is completely cloudy, then it is still observed as being fairly homogeneous. The assumption that the sky is blue is also violated from the time around sunset until sunrise. We will assume that the images are taken during reasonable daytime hours. Since we have already made the assumption that the sky is the dominant illuminant, the last observation of the sky being bright is fair. On a clear day, the sky alone is several magnitudes brighter than objects in the scene.

Using these assumptions, it is possible to obtain an preliminary selection of regions that correspond to sky. Each sample region in the image (pixel or otherwise) is first projected to the surface of a sphere whose center coincides with the optical center of the camera. A series of masks are then applied to each color region on the sphere in order to filter the input before estimating the parameters of the sky model. The first mask removes regions that contain high spatially varying frequency color components. The second mask examines the temporal frequency removing regions that contain low mean ratios of the blue color component to red color component. The final mask removes regions that contain mean luminance values below mean luminance over the entire image. This process is illustrated in Figure 3.3. Figure 3.3(a) shows the images projected onto the sphere. The portions of the sphere highlighted in red in Figure 3.3(b) correspond to regions in the image that contain high spatial frequencies. In Figure 3.3(c), regions that contain the wavelengths of light inconsistent with those typical of sky are highlighted in yellow. Regions in the images that contain low luminance values are highlighted in green in Figure 3.3(d). The complete mask resulting from the union of all three individual masks is shown in Figure 3.3(e). Notice that regions that correspond to windows in the original images are not masked and the highly reflective glass is justifiably misclassified as sky.

Once these portions have been identified, the estimation of the complete sky model can commence. The missing portions of sky are estimated through the use of an analytic model known as the all-weather sky luminance model [57] and is a generalization of the CIE standard clear sky formula. For an element of sky at an angle θ to the zenith, and at an angle ϕ to the position of the sun, the luminance of that element is described by:

$$Ls(\theta, \phi) = Lvz \; f(\theta, \phi)/f(0, z)$$

where Lvz is the luminance at the zenith and z is the angle between the zenith and the sun. Since both Lvz and $f(0,z)$ are constants, we can replace them with the single variable L_z. The function $f(\theta,\phi)$ is defined in [57] as follows:

$$f(\theta,\phi) = [1 + a\ exp(b/cos\theta)][1 + c\ exp(d\phi^h) + e\ cos^2\phi].$$

Here, a, b, c, d, e and h are all constants that can be adjusted to fit the supplied data together with L_z. The first term models the variation of illumination with respect to the zenith. The second term models the illumination component with respect to the sun. The images in our dataset are registered with respect to some absolute coordinate system and each image is also annotated with the date and time at which the image was taken. Using this information the position of the sun can be computed. Alternatively, if absolute registration is not available, or the images are not time-stamped, the position of the sun can also be selected interactively by the user.

The parameters for the sky model are estimated separately for each color channel (red, blue and green) leading to a full luminance and color description of the sun and sky. The parameters are estimated using the Levenberg-Marquadt non-linear optimization [62] method which given some initial parameter estimates, performs gradient descent on the partial derivatives in order to locate the global minimum. Convergence is not guaranteed due to the possible existence of local minima around the initial parameter estimates. In practice however, the algorithm is robust in finding a good minimum. The gradient decent method requires the computation of the following partial derivatives with respect to the parameters:

$$\begin{aligned}
\frac{\partial f}{\partial a} &= L(exp(b/cos\theta))(1 + c\ exp(d\phi^h) + e\ cos^2\phi) \\
\frac{\partial f}{\partial b} &= L(a/cos f(z))(exp(b/cos\theta))(1 + c\ exp(d\phi^h) + e\ cos^2\phi) \\
\frac{\partial f}{\partial c} &= L(1 + a\ exp(b/cos\theta))(exp(d\phi^h)) \\
\frac{\partial f}{\partial d} &= L(c\phi^h)(1 + a\ exp(b/cos\theta))(exp(d\phi^h)) \\
\frac{\partial f}{\partial e} &= L(1 + a\ exp(b/cos\theta))(cos^2\phi) \\
\frac{\partial f}{\partial h} &= L(ch\phi^{h-1})(1 + c\ exp(d\phi^h) + e\ cos^2\phi)
\end{aligned}$$

Once the parameter have been estimated, the brightness (and color) of missing portions of the sky can simply be computed using Equation 3.2. An example of an estimated sky model is shown in Figure 3.3(f). The recovered sky model fits the true sky variations to within an error of 5%. This complete model can also be used to further detect regions initially missed using the masks. The regions can be added to the original input set and the model parameter recomputed to improve the fit. The

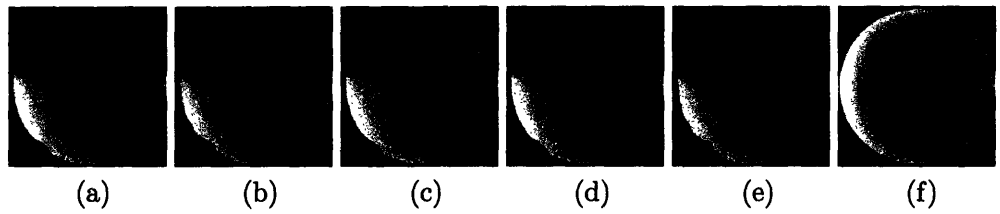
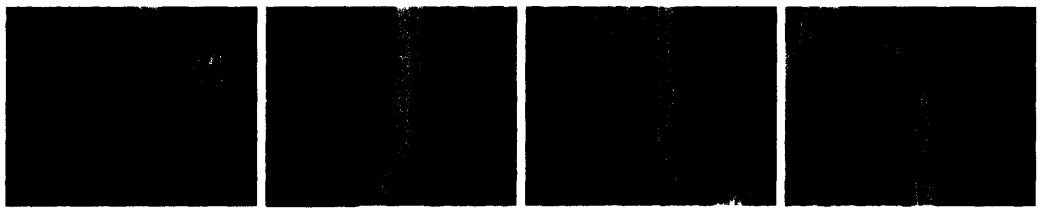


Figure 3.3. Estimation of complete sky illumination from images

Input Images



Background Masks

Figure 3.4. Background masks detected for a number of example input images from the City Scanning Project. The red regions indicate the degree of certainty that the region is background (sky).

sky model parameters quickly converge to correct values, usually within two iterations of the process. Regions in the images that are used to compute the sky model are then classified as background and all other regions as foreground thereby creating a foreground/background mask. This mask however, is not a binary one, the degree of contribution and similarity of each region to the recovered model can be measured and used to create the mask. Uncertainties in regions that could belong to either foreground or background can then be modeled using real values. These image based masks can be used during the reconstruction process in much the same way as for algorithms that require background subtraction. Examples of background masks are shown in Figure 3.4. Four images from the City Scanning dataset are shown above corresponding images with background samples highlighted in red.

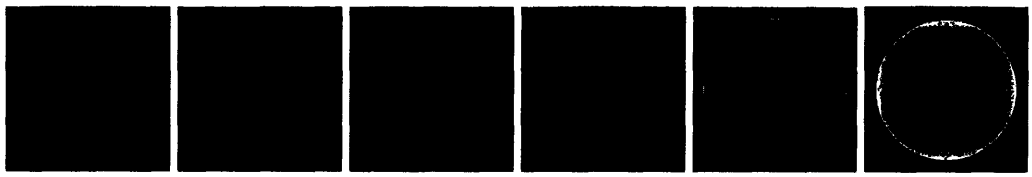


Figure 3.5. Sphere created through successive divisions and point re-mapping of octahedron triangle mesh.

■ 3.3 Implementation and Experiments

The algorithm described above was tested on a number of data nodes from the City Scanning dataset. Each node consisted of between 20 to 40 images captured in High-Dynamic Range format. The images are first projected to a triangulated sphere; the sphere is modeled by successive divisions and point re-mapping to a sphere of a octahedron triangle mesh as shown is Figure 3.5.

Each image is projected onto the spherical mesh and the mean and variance of colors within each facet are computed. Masks are then applied to each triangle, removing those that correspond to regions of low luminance, high spatial frequency or spectrally undesirable regions (unlikely sky colors). Unlikely sky colors are determined by comparison of the red and blue color channels denoted ρ_r and ρ_b respectively. It can be shown [69] that a first approximation for estimating sky colors are made using a classifier where:

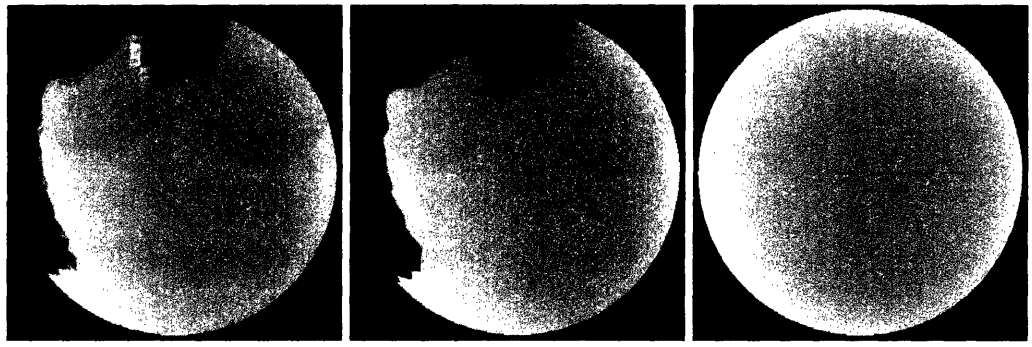
$$\frac{\rho_r}{\rho_b} < 0.95.$$

This RGB ratio filter is derived empirically from data and is shown to reliably return conservative estimates of image regions corresponding to sky. Alternatively, the saturation value in HSV color space can be used [16]; a value above 54% would indicate a region of sky, a value below 47% would indicate cloud, and values in between would be uncertain. This simple discrimination however has not be shown to be particularly accurate and in our implementation, we instead use the RGB ratio filter.

Once each region is classified, all foreground triangles are culled and the remaining regions are used to estimate the sky model parameters for each color channel.

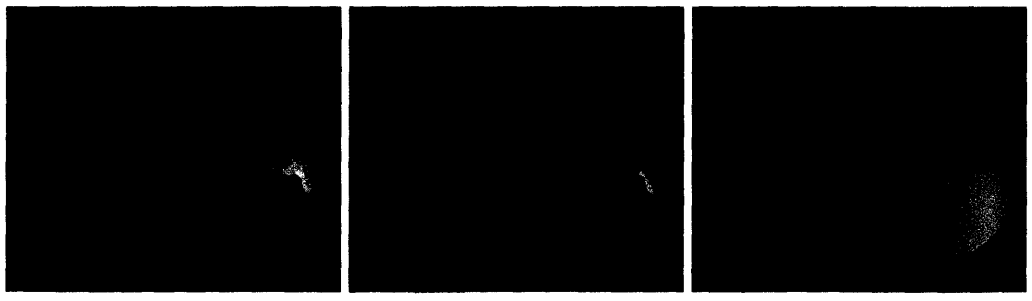
Figure 3.6(a) shows a typical example of input images to the illumination estimator after projection to a sphere. Figure 3.6(b) shows the same input after application of the masks leaving only those regions that conservatively correspond to sun and sky. The complete illumination after parameter estimation is shown in Figure 3.6(c).

The algorithm converges to a visibly correct solution despite a low number of input samples after application of the masks as shown in Figure 3.7(c). The mean error between the estimated sky model and actual intensity values for this node was less than 3%. In cases when the sun is present in one or more of the images, resulting in a region



(a) (b) (c)

Figure 3.6. Typical example: (a) Input data projected to sphere. (b) Data after application of masks. (c) Complete estimated illumination.



(a) (b) (c)

Figure 3.7. (a) Input data projected to sphere. (b) Data after application of masks. Note the low number of remaining samples (c) Complete estimated illumination.

on the sphere that is several magnitudes brighter than the rest, it can be detected and incorporated into the input data overriding the masking decision if necessary. Since the pose of the cameras are in absolute coordinates and also time stamped, the position of the sun can be computed and the regions scanned for high luminance values. Spurious bright spots caused by reflections can therefore be ignored by only considering regions in and around the known sun position. Figure 3.8 shows an estimated illumination model containing the sun. Note that the increased brightness values around the sun have been correctly modeled. The mean error in this case was around 7%, this greater error is due to the larger luminance values around the sun.

As mentioned earlier, the masks are used to provide a conservative estimate of which regions in the image correspond to sun and sky. Figure 3.9 shows a example of correctly estimated parameters despite the windows in the scene being misclassified as sky. The mean error between the input shown in Figure 3.9(b) and the estimated model in Figure 3.9(c) was 4%.

(a) (b) (c)

Figure 3.8. (a) Input data projected to sphere. (b) Data after application of masks. (c) Complete estimated illumination. Note the correctly modeled increased luminance around the sun.

(a) (b) (c)

Figure 3.9. (a) Input data projected to sphere. (b) Data after application of masks. Note that the building windows have been understandably misclassified as sky. (c) Complete estimated illumination.

■ 3.4 Summary

Changing illumination conditions can lead to very different images. When aiming to match objects across images using only color information, it is essential that the illumination is modeled and the observed image colors adjusted to compensate for this change in illumination. In this chapter, we have discussed the problem of outdoor illumination estimation from a set of images with fixed optical center. We have presented a novel method for automatic direct estimation of background illumination from a number high-dynamic range (HDR) images. We have shown that is it possible to fit the CIE all-weather model to a set of filtered image regions and recover the illumination conditions produced by the sun and sky at the time of image capture.

The true object illumination will be far more complex and depend on other factors such as mutual illumination. These initial estimates of illumination conditions are therefore used in conjunction with the voxel opacity and color estimator described in Chapter 2 to improve color matching across images with changing illumination. The following chapter discusses the problem of color matching in the context of color constancy. We also investigate the options of color space that are available and determine

the most suitable one for our application.

Chapter 4

Probabilistic Color Constancy

The extraction of true world color information from images is a valuable ability for machine vision systems. As humans, we base everyday decisions on the colors we observe. As well as decision making, color is also of use as a cue for object recognition. The process of selection on the basis of color would appear to be a simple yet effective way of discriminating between objects, a task that even a machine could be programmed to do. In reality, our own vision systems are very sophisticated, allowing us to recognize the same color under different lighting conditions by automatically compensating for the intensity and color of the light around us. The execution of the same process in computer vision is known as the problem of color constancy.

In this chapter, we examine the use of color as a cue for matching regions across images. We present a framework for color matching in probability in which colors are modeled and matched using Bayesian statistics. We also describe how modeling color as a probabilistic quantity allows partial matching as well as the ability to robustly match colors across images despite illumination variations in the scene.

■ 4.1 Background

The difficult problem of identifying true object color from images has received much attention from vision researchers. Color constancy approaches vary from simple single image techniques to more complex methods involving multiple images. In order to understand the foundation of the problem to be solved, we begin our discussion of previous work in the area with a look at the interaction between illuminant and object surface.

■ 4.1.1 Light Signal

A digital image is a sampling of a light signal $L(\lambda)$ which is a continuous function of wavelength λ and geometric properties of the scene. The light begins its journey to the camera at the light source $S(\lambda)$. The light interacts with the surfaces in the scene on its way to the camera. This interaction is considered linear and the reflectance can be defined as the ratio of reflected light to incident light. In the general case, this reflectance ratio is also a function of the direction of illumination, the direction of the camera, and surface normal. This relation gives rise to the bi-directional reflectance

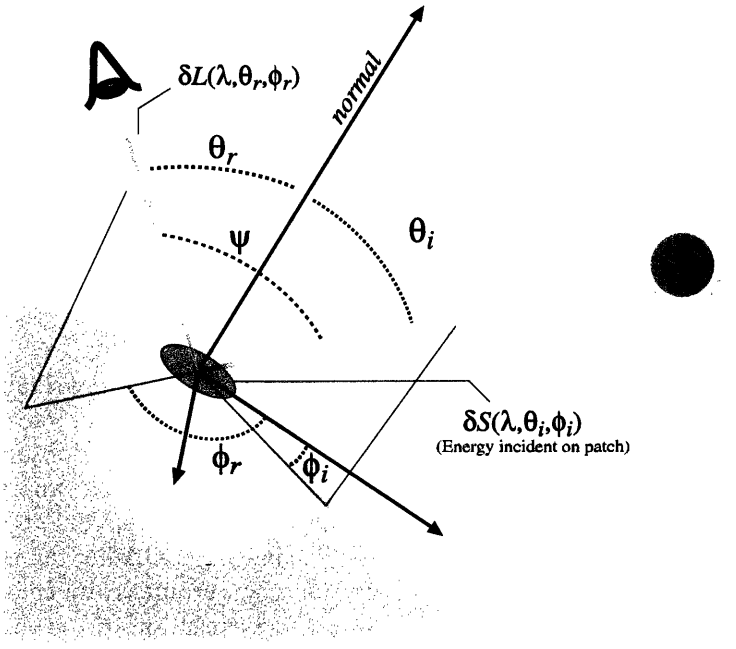


Figure 4.1. Bi-directional Radiance Distribution Function or BRDF is define as the patch size tend to zero.

distribution function or BRDF [32].

■ 4.1.2 Bi-directional Reflectance Distribution Function

The Bi-directional Reflectance Distribution Function is a four dimensional function that describes the interaction of a light signal with a surface. It is defined by the following relation:

$$\rho(\lambda, \theta_i, \phi_i, \theta_r, \phi_r) = \frac{\delta L(\lambda, \theta_r, \phi_r)}{\delta S(\lambda, \theta_i, \phi_i)}$$

where $\delta L(\lambda, \theta_r, \phi_r)$ is the differential image radiance (reflected light) with wavelength λ, at zenith angle θ_r and azimuthal angle ϕ_r. The light source is defined by the term $\delta S(\lambda, \theta_i, \phi_i)$ and is the differential incident surface irradiance at an angle θ_i to the zenith and azimuthal angle ϕ_i.

This model is sufficient for describing most surfaces except those that fluoresce. Fluorescent surfaces absorb light at one frequency and emit light at a different frequency. Since these surface are rare, especially in outdoor urban environments, and generally difficult to handle, they are not considered here. Analytic approximations to surface BRDF exist in forms such as the Phong [58], Cook-Torrence [12] and Lafortune models [37]. These models are often unable to capture subtleties in the reflectance properties of real materials while also being driven by parameters that are unintuitive making them

difficult to tune in practice. Recently, researchers have persisted with the difficult and tedious task of accurately measuring the BRDF for a variety of surfaces [46,47].

Isotropic Surfaces

The reflectance for most real world objects does not change when the surface is rotated about the surface normal. These surfaces are known as isotropic and lead to a simplified BRDF where only the angle ψ between the incident and reflected ray needs to be considered, such that:

$$\rho_{iso} = \rho(\lambda, \phi_i, \phi_r, \psi)$$

Lambertian Surfaces

Further approximations can be made with regards to the interaction between light and surface. If we assume that incident light is reflected equally, in all directions, we arrive at a simple surface model known as a Lambertian reflector that does not depend on the incident or reflected light directions. It can be shown that the reflectance in this case is defined as:

$$\rho_L = \frac{\rho_0}{\pi}$$

where π is a normalizing factor and ρ_0 is a constant. Unfortunately, most real world materials do not follow Lambert's law, but due to its simplicity, it can lead to a reasonable first approximation for describing the interaction of light and surface. The independence of the reflectance function from the directions of light and camera have made it an attractive approximation to use in many vision algorithms such as shape from shading.

Given this notion of light-surface interaction, many vision algorithms aim to identify the surfaces in the scene that share some property, such as color, despite changes in illumination (direction or otherwise) or viewing direction. Since we do not directly observe reflectance but instead only its interaction with light, reflectance properties of a surface may not be immediately intuitive independent of a light source. Reflectance is therefore recovered in relation to some known illumination referred to as a *canonical* illuminant.

Mutual Illumination

In the real world, light reflected from a surface interacts with several other surfaces on its way to the camera. In this way, surfaces mutually illuminate each other. Estimating the illumination for images where this interaction dominates can be very difficult. In this thesis, we assume that the illumination is mainly provided through incident light directly from the light source and the process of mutual illumination can be ignored.

4.1.3 Varying Illumination and Noise

Many factors can alter the observed intensities in the image including changes in scene illumination, and imaging noise. The lighting can vary both in time and space leading to different effects on the resulting images.

Spatially varying illumination

The illumination in the scene is often a function of position in the 3-D world. The position and orientation of the objects relative to the light sources in the scene therefore affect the way in which they appear in the image.

Time varying illumination

Images of a scene acquired simultaneously are illuminated identically, but allowing images of the same scene to be taken over time while also allowing the light source to change provides access to more information about the illumination and scene. A simple method for modeling the illumination change is via a linear transformation. Each pixel in the image p_i, with three color channels is mapped to a pixel value under known canonical illumination p_c by a 3 by 3 matrix $\mathbf{M}$ such that $p_c = \mathbf{M}p_i$. If the matrix $\mathbf{M}$ is restricted to be diagonal such that each color channel is independent, then the illumination model is further simplified. This is known as the diagonal model [24] and has been used extensively in the color constancy research.

Imaging noise

Since the process of image capture is not perfect, noise can be introduced at various stages. Quantization noise is one such unavoidable source of imaging error and is due to the finite number of bits available to store each pixel. The effect of imaging noise is such that it is practically impossible to take two images that exhibit identical responses at every pixel.

4.1.4 Previous Work

In the presence of these effects, researchers have aimed to tackle the problem of color constancy and recover information on the true scene color from images. We give a brief overview of some of the many methods currently used to perform color constancy.

White Patch

This method presumes that a perfectly white patch is present in the scene and can therefore be considered a region of maximal reflectance for each color channel. Spectral normalization can then be performed where each channel is independently normalized to maximize the color of this region to "white". In practice, variations on this approach are used such as the 1% white patch, which considers the brightest 1% of pixels in an effort to reduce the effects of noise. This method is appealing in its simplicity and only

requires a single image as input. The assumptions for this method are untrue however, since the existence of a maximally reflective patch in the scene is unlikely. The method also fails in the presence of highly specular regions that can be mistakenly identified as white patches in the scene.

Grey World Algorithm

This simple approach operates by computing a single statistic of the reflectance in the scene, namely the mean. The algorithm assumes that the average reflectance of all objects in the scene is approximately constant and equal to some known reflectance (grey). Each color channel is then normalized to equalize the observed and known reflectance. The choice of grey largely determines the effectiveness of the algorithm and has led to reasonable results from synthetic data. For real images however, the true value for real world grey reflectance is not readily available, making a valid assessment of the technique difficult.

Retinex Method

The Retinex methods [38–40] are aimed at computationally modeling the human vision system. Although various versions of the Retinex algorithm exist, the central idea is to estimate the surface lightness in each channel by comparing it against some statistic of neighboring pixels. Estimates are made of the brightness of every observed surface in each channel. These estimates rely on the assumption that small spatial changes in the responses (such as those in neighboring pixel values) are due to illumination variations and larger changes are due to changes in surface properties. Retinex methods are considered to be robust when dealing with images with slowly spatially varying illumination but have only been tested on images with fairly uniform illumination distributions.

Neural Network Color Constancy

Neural Network approaches [26] to color constancy require training on a set of synthetic images generated from a database of known illuminants and reflectance. The method works by dividing chromacity space into a series of bins. During the learning stage, the input to the network is defined by some function of the image (usually a chromacity histogram together with the true illuminant). The outcome of the network is then compared to the actual illuminant and the difference back-propagated [5] to update the weights of the network, thus learning to estimate the illuminant based on the input.

K-L Divergence

The K-L or Kullback-Leibler divergence [66] is a measure of the similarity of two distributions. For problems in color constancy, K-L divergence can therefore be used to measure the difference between a true color distributions $C = c(x)$ and some approximate color distribution, $C' = c'(x)$. For a discrete distribution of n classes, the K-L divergence is defined as:

$$KL(C||C') = -\sum_{x=1}^{n} c(x) log \frac{c'(x)}{c(x)}$$

The K-L divergence is smaller for distributions that are more similar. The method relies on a set of illumination parameters, together with a maximum likelihood formulation to estimate the most probably lighting given a set of observed colors. This also requires statistics of real world surface color independent of lighting in the form of histograms. Once the most likely global illumination conditions have been estimated, they can be factored out to identify the true world color under canonical lighting. This method requires at least two images of a scene taken from the same position but under differing illumination and cannot be performed on a single image.

Bayesian Color Constancy

Similar to K-L divergence, Bayesian color constancy [8, 25, 80] estimates true world color by undoing the most probable lighting according to available priors of real world illumination. The method begins with prior knowledge of probabilities of illuminations and surface reflectance occurrence. Using this information, we can simulate a sensor response given a combination of a illuminant and surface reflectance. Let the observed sensor response or image be described by the variable $\mathbf{y}$, and let the parameters of combinations of illumination and surface reflectance be $\mathbf{x}$. We can then use Bayes' rule to estimate the probability distribution over the parameters given the image, $P(\mathbf{x}|\mathbf{y})$ as follows:

$$P(\mathbf{x}|\mathbf{y}) = \frac{P(\mathbf{y}|\mathbf{x})P(\mathbf{x})}{P(\mathbf{y})}$$

The value of $\mathbf{x}$ that corresponds the maximum $P(\mathbf{x}|\mathbf{y})$ is chosen since it represents the most probably surface reflectance given the possible illumination. This method can prove to be computationally expensive since the number of parameters is a function of the possible number of surfaces in the scene (i.e $\mathbf{y}$ could be very large). The method also makes the assumption that surfaces are defined by reflectance information that is independent, but for most real images the source surfaces for neighboring image pixels are identical thus resulting in a fundamental dependency. It is therefore necessary to either perform image segmentation to separate regions in the image corresponding to the same surface or sample a few points sparsely before estimating the reflectance. Finally, the method also requires distributions of real world illuminations which may not be well known in practice.

Color Constancy from Multiple Views

When multiple images of the same scene are available under different illumination, more information regarding the reflectance in the scene can be deferred. Research aimed at utilizing this fact has emerged but without tremendous success in the most

general cases. Improved results are made possible however by restricting the type of illumination considered (e.g CIE daylight).

For outdoor images, such as our chosen data set, we can assume the dominant illuminant is the sun and sky. Using the CIE daylight model to describe the illumination is a natural choice. In our approach, we estimate the position of the sun and sky conditions at the time of capture using the methods described in Chapter 3. These estimates are then used to obtain a better estimate of surface reflectance under canonical illumination. We use the diagonal model [24] to model the changes due to the varying illumination described above with the additional assumption that the surfaces are Lambertian. This provides a simple method by which to compute the response of a surface under some canonical illuminant; the intensity of the estimated daylight illuminant in each color channel can be factored out of the observed pixel values leading to an estimate of surface reflectance under some canonical illuminant. This computation comes directly from the diagonal model discussed earlier and although it is over zealous in its assumptions, it does lead to an improved normalization of observed object colors across images. Volume reconstruction algorithms can then use these normalized color estimates for typical consistency checks rather than the pixel values directly. This proves to be an effective way of loosening the fixed lighting constraint placed on most reconstruction algorithms.

■ 4.1.5 Color Constancy for Volume Reconstruction

In the case of volume reconstruction algorithms, many simple color matching techniques have been proposed. Most, if not all, of these algorithms presume fixed illumination and therefore circumvent the color constancy problem. Colors in the images can be directly compared to one another and the reconstruction directed according to the consistency between these colors.

The consistency of colors across images is typically measured using a direct distance measure in a chosen color space (usually RGB). The distance is computed between the average observed color over all images and the estimate based on a single image. In general, the distance $d_{i,j}$ between two colors C_i and C_j, represented as vectors, is defined as:

$$d_{i,j} = \sqrt{||C_i - C_j||^2}.$$

This distance measure can then be normalized by the maximum distance between two colors in color space d_{max}[1]. An agreement $a_{i,j}$ between C_i and C_j can then be computed using the following linear relation:

$$a_{i,j} = 1 - \frac{d_{i,j}}{d_{max}}. \tag{4.1}$$

An agreement of $a_{i,j} = 1.0$ implies a perfect match between the colors, where as an

[1]In RGB space, this is the length of the cube diagonal, $d_{max} = \sqrt{3}$

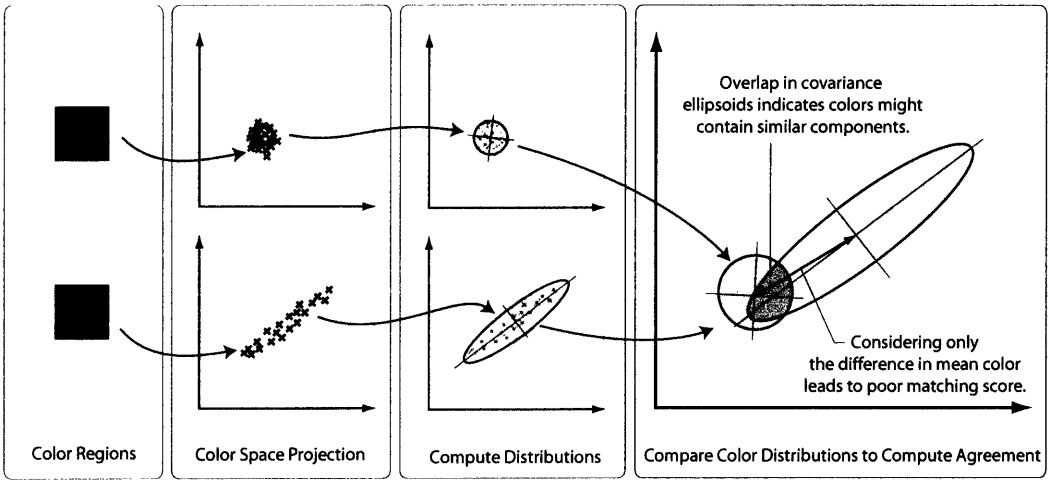


Figure 4.2. Computing agreements between regions of color using only the region mean results in poor matching despite common color components in each region. By also using second order statistics, the color overlap can be used to give a partial agreement score measuring the probability that distributions originally came from the same variable.

agreement of $a_{i,j} = 0$ would indicates that the colors are maximally different. The agreement is then thresholded to decide if the colors are sufficiently close together to be considered the same.

In the voxel reconstruction algorithm described by DeBonet and Viola [18], an exponential relation is used to compute agreement:

$$a_{i,j} = e^{\frac{-d_{i,j}^2}{\sigma^2}}, \tag{4.2}$$

where σ is a free parameter expressing a measure of noise in the observed colors. This approach is a step towards modeling colors with a direct way of influencing the degree of matching according to the level of noise in the measurements.

Consider now the case in which a voxel in 3-D space projects onto a region of an image consisting of multiple colors (each color must now be represented as a distribution of color, or at least some statistic of that distribution). Most reconstruction algorithms use only the mean color of the projected region, this works if the region is small or completely homogeneous. If however, the region contains multiple colors, the underlying distribution is lost by considering only the mean. This deficiency can give rise to false negative match results as shown in Figure 4.2. A homogeneous region of color in the source patch is incorrectly identified as matching poorly with the region made up of colors including that of the source. In this situation, an agreement function that allows for partial matching would be preferable. Therefore the underlying variation color must also be captured to perform robust color matching.

We propose the addition of second order statistics in the matching of colors across images in an effort to enable partial matching of color mixtures even when the means

differ considerably. By using probability to represent, combine and match colors, we show that superior matching can be achieved. We also show that by moving from the commonly used RGB color space to the perceptually uniform CIE-*Luv* space, matching can be performed successfully even in the presence of minor illumination variation.

■ 4.2 Probabilistic Color

The representation of colors in a probabilistic framework leads itself well to the idea of partial matching. We utilize Gaussian distributions to describe colors due to its mathematical simplicity. The conversion of colors to normal distributions allows several benefits:

- Includes all the functionality of matching color means,
- Allows degree of matching rather than threshold matching,
- Homogeneous color regions in an image can now effectively be differentiated from a region consisting of a mixture of colors with the same mean,
- Noise can be modeled directly,
- Use of well understood mathematics to match distributions.

■ 4.2.1 Modeling

We define color as a probabilistic quantity in a chosen space. Each color C_i is described by a multi-dimensional normal distribution $C_i(x)$:

$$C_i(x) = \frac{1}{(2\pi)^{1.5}|\Sigma_i|^{0.5}} e^{-\frac{1}{2}(x-\mu_i)^\top \Sigma_i^{-1}(x-\mu_i)},$$

where μ_i is the mean color, Σ_i is the $n \times n$ color covariance matrix, and n is the number of color components. Each dimension of the distribution represents a separate component of the color space.

Combining

Defining color as a normal distribution allows use of sequential (on-line) update equations for both mean and covariance in order to combine colors. This becomes useful when estimating the color of a region (2-D or 3-D) from multiple color samples that may be acquired in any order. In order to combine two colors, C_i and C_j with weights w_i and w_j respectively, the covariances Σ_i and Σ_j are first combined to form a new covariance Σ':

$$\Sigma' = \frac{w_i\Sigma_i + w_j\Sigma_j}{w_i + w_j} + \frac{w_i w_j}{(w_i + w_j)^2}(\mu_i - \mu_j)^2, \tag{4.3}$$

and the means μ_i and μ_j are combined to form a new mean μ':

$$\mu' = \frac{w_i\mu_i + w_j\mu_j}{w_i + w_j}. \tag{4.4}$$

In our reconstruction algorithm, this will allow the combination of colors at every voxel. As each image is processed, its color contribution will be combined with the color already at each voxel. The weights used for the combination will depend on the responsibility of the voxel in each image.

■ 4.2.2 Matching and Agreement Computation

Once the color has been converted to a statistical quantity, we define a metric, called the *agreement* as in [18], to measure the similarity between any two colors. The agreement values are normalized to lie in the interval $[0 - 1]$ such that:

- Colors with similar means and small covariance ellipsoids result in high agreement values such that $a_{i,j} \approx 1$
- Colors with differing means and small covariance ellipsoids result in an low agreement values, $a_{i,j} \approx 0$
- Colors with similar means but large overlapping covariance ellipsoids result in a partial matching based on the overlap $0 < a_{i,j} < 1$.

The similarity between color distributions could be measured using the K-L divergence method discussed earlier but would required modification to satisfy the limiting conditions. We instead opt for a different way method of computing the agreement $a_{i,j}$ using Bayesian statistics:

$$a_{i,j} = \int_{-\infty}^{\infty} C_i(x)C_j(x)\delta x$$

$$a_{i,j} = \frac{1}{(2\pi)^{1.5}|\Sigma_{i,j}|^{0.5}} e^{-\frac{1}{2}(\mu_i-\mu_j)^\top \Sigma_{i,j}^{-1}(\mu_i-\mu_j)}, \tag{4.5}$$

where $\Sigma_{i,j} = \Sigma_i + \Sigma_j$. In order to produce an agreement whose values lie in the correct range, $a_{i,j}$ must be normalized according to some maximum value. To achieve this, we define a covariance Σ_{min} that describes the minimum variation in noise we expect to observe in the images (due to the imaging process). This covariance is then used to compute the maximum agreement value (the best match between two colors) a_{max} such that:

$$a_{max} = \frac{1}{(2\pi)^{1.5}|\Sigma_{min}|^{0.5}}.$$

The normalized agreement $\hat{a}_{i,j}$ is then simply:

$$\hat{a}_{i,j} = \frac{a_{i,j}}{a_{max}}.$$

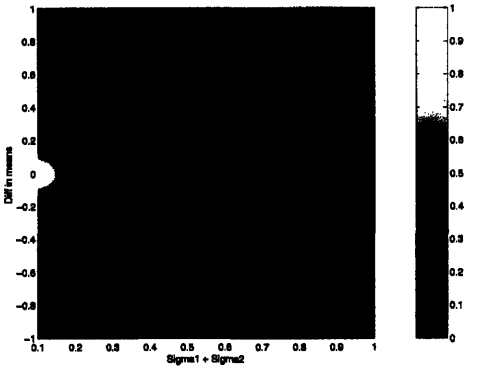


Figure 4.3. Agreement as a function of difference in means $(\mu_i - \mu_j)$ and $\Sigma_{i,j}$ computed using Equation 4.5.

The normalized agreement $\hat{a}_{i,j}$ is shown as a function of difference of means $(\mu_i - \mu_j)$ and $\Sigma_{i,j}$ in Figure 4.3 for the one dimensional case. The reader should note that this definition satisfies all the requirements outlined earlier by measuring the degree of overlap between the two color distributions. When the difference in the means is small and and the combined covariance is also small, implying that the colors are very similar, the agreement is high. If however the combined variance is large, it indicates a large variation in one or both of the colors and the agreement reflects this through a lower match. Very low agreements result for tight distributions with different means as required.

Although the benefits of matching colors in a probabilistic framework are evident, we now show that further gains are achievable through appropriate selection of a color space.

■ 4.3 Choosing a Color-space

Our vision systems are based on the responses of light sensitive cones to short, medium and large wavelength. These are sometimes referred to incorrectly as red, green and blue cones, since the wavelengths do not correspond directly to those colors that correspond to single colors. Colors can be represented in many forms, and depending on the application, each brings its own advantages and disadvantages. In this section, we explore the possibility of using various color-spaces for application in volume reconstruction algorithms.

Perceptually Uniform

A color space is considered perceptually uniform if small perturbations to a color component value result in approximately equal changes in the perceived color across the entire range of that value. Researchers have studied the way in which we as humans perceive colors in an effort to create a color space that is perceptually uniform. Colors

that differ equally according to our visual system are represented by equally distant points in the color-space. Although this is only true for small changes in color, it permits the same level of discrimination throughout the range of colors and is therefore of particular use in comparison measures for color based volume reconstruction.

■ 4.3.1 RGB

In digital imagery, colors are usually encoded using three color channels. The most common image coding system consists of red, green and blue components and is known as the *RGB tristimulus* [60]. This space relies on the theory of trichromacity and the additive nature of color. Colors are represented as a coordinate triplet and many computer systems, images, and file formats make use of this representation. The advantage of using RGB color is that it is based on the human visual system, and is well understood. Each channel can be viewed, and readily understood, independently. As a result, most image processing techniques are also designed around this color-space.

One problem with the RGB space is an inability to reproduce all visible colors irrespective of precision using only positive values. Since luminance information is also tied into each channel, changing the brightness requires a change in each of the three components. Another problem is that RGB color is not perceptually uniform and is also device dependent such that the same color could be perceived differently on different devices due to voltage variations. The introduction of the Standard RGB or sRGB [1] is aimed at resolving the device dependence although it still fails to address the other issues.

■ 4.3.2 HSV

Another popular color space is that of HSV which defines color in terms of Hue (tint), Saturation (shade), and Value (brightness). This color space is often depicted as a hex-cone but the coordinate system is cylindrical in which the Hue value ranges from $[0 - 2\pi]$. The saturation S can be seen as the purity of the color and lies within the $[0 - 1]$ range, where $S = 1$ is the purest color. The brightness or tone is captured in the value V component and ranges from $[0 - 1]$ where $V = 0$ is black.

The HSV color space offers many advantages in that it is easy to control, intuitive, and therefore useful for design purposes where the separation of brightness from chromacity (value from hue and saturation) makes it easier to manipulate and select colors. Like RGB space however, HSV is not perceptually uniform. Additional, its cylindrical coordinate system does not lend itself well to the matching functions discussed in Section 4.2 since the wrapping discontinuity of the hue variable causes problems.

■ 4.3.3 CIE-XYZ Primaries

Defined by CIE in 1931 and called X, Y, and Z, the standard primaries can be used to define all visible colors using only positive values. The separation of luminance from chromacity is also used to define the XYZ color space. These primaries can be computed

via a linear transform from the RGB space. Since luminance is a linear combination of the RGB components, the Y component intentionally matches the luminous-efficiency function of the human eye. The chromacity components x and y are defined as follows:

$$x = \frac{X}{X+Y+Z}$$

$$y = \frac{Y}{X+Y+Z}.$$

The inverse conversion matrix (from XYZ to RGB) contains negative coefficients (see Appendix A), and it is therefore possible to convert XYZ values to RGB values that are negative or greater than one. As a result, there are visible colors that exist in XYZ space that map to points outside the RGB cube and therefore cannot be represented using typical RGB values. The coefficients of the conversion matrices depend on the light source under which it is calibrated. An example of conversion matrices calibrated for D65 white-point is given in Appendix A.

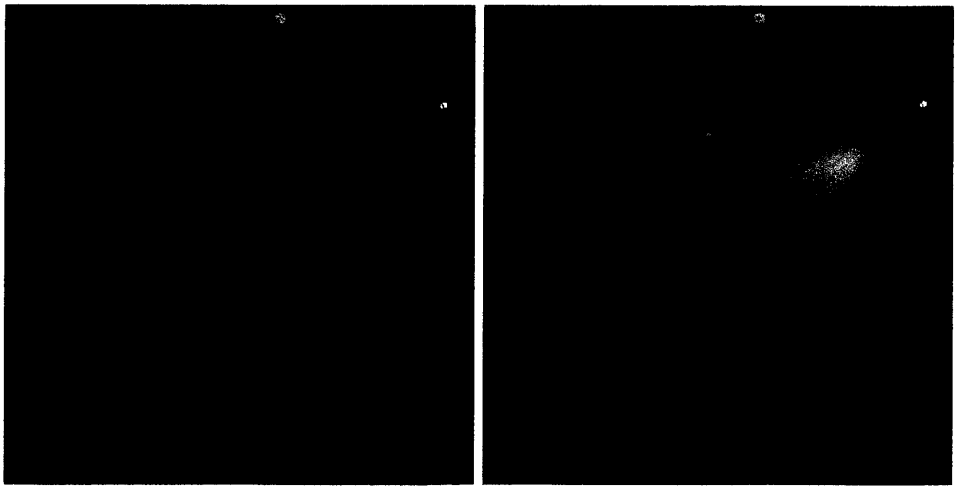
As mentioned earlier, the Y component is directly related to the luminance. The chromacity coordinates can also be computed from the ratio of each of a set of tristimulus values to their sum. This system can encode any visible color as a set of positive XYZ values. The only disadvantage of using this color system is that is also is not perceptually uniform.

■ 4.3.4 CIE-*Luv*

Based directly on the standard primaries XYZ, CIE-*Luv* is an adaptation to directly address the problem of perceptually uniformity. Small differences between perceived colors in *Luv* are approximately equalized through appropriate scaling and non-linear conversion of XYZ in accordance to a reference white point. CIE-*Luv* color space therefore inherits all the benefits from all the advantages of the XYZ space while also being perceptually uniform. The L component also measures the luminance, while u and v encode the chromacity. An indirect advantage of perceptual uniformity is that it naturally leads to less sensitivity towards changes in luminance as with our own vision system. One disadvantage of using CIE-*Luv* however, is that is is not as intuitive to navigate through in comparison to RGB and HSV color spaces. Conversion formulae between *Luv* and XYZ space is given in Appendix A.

■ 4.3.5 Other Color Spaces

The are a vast number of different color spaces available to work in, each developed for a specific purpose and system in mind. Other color spaces that separate chromacity and luminance information (such as YIQ) could also be used to model and match colors providing suitably lower weighting is given to color change information over illumination effects. The advantage of using a perceptually uniform space such as CIE-*Luv* is that it provides a non-linear mapping with adjustments made with these effects in mind.



(a) (b)

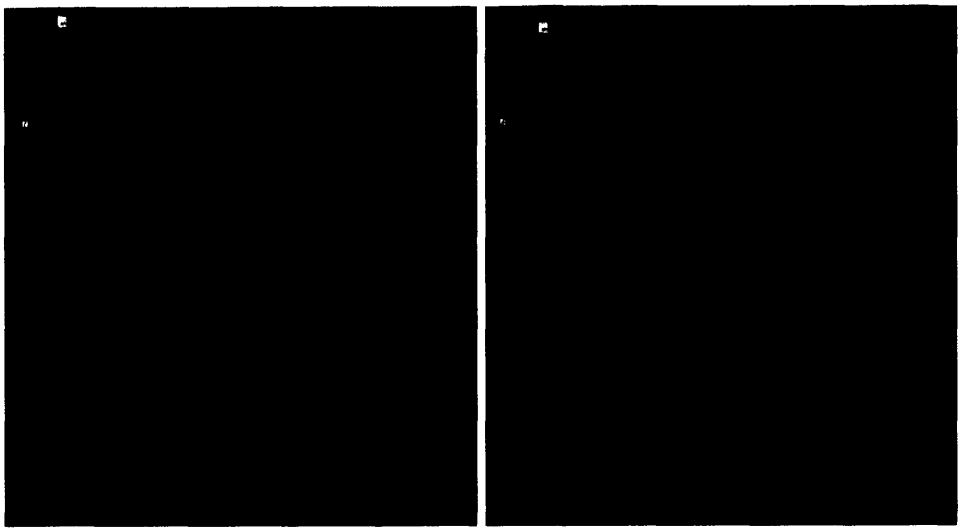
Figure 4.4. Result of applying the matching function to the color at the center of the RGB cube and other colors within the cube. For a matching functions defined in RGB color space (a), the result is a Gaussian sphere. Matching in *Luv* space (b), we obtain an ellipsoid with high variance in the illumination direction.

Changes due to illumination have less of an effect on the perception of a particular color resulting in smaller distances between colors related by a change in illumination. These smaller distances allow the use of uniform Gaussian distributions in CIE-*Luv* space to represent small changes in color but slightly more variation in illumination.

■ 4.4 Color Space Comparisons

In order to evaluate the utility of using CIE-*Luv* color space, we tested the probabilistic matching function in both RGB and *Luv* space by examining the agreement values between a preselected sample color and those in the RGB cube. The first test measures the agreement between a grey color sample $rgb = (0.5, 0.5, 0.5)$ at the center of the RGB cube and all other colors in the space. The results for the RGB case for a diagonal covariance matrix is predictably a Gaussian sphere. In the case of *Luv* matching, colors are mapped from RGB to *Luv* before measuring the agreement. Here, we see that high matching values resemble a ellipsoid tilted in the luminance direction. This indicates that the variance is higher and thereby less sensitive in the luminance direction which is in agreement why psychophysical data (Figure 4.4).

In the second experiment, a red color sample $rgb = (1.0, 0.0, 0.0)$ is chosen and the agreement measured with points throughout the RGB cube. The results are again predictable for the RGB matching case as a quadrant of a Gaussian sphere. When matching in *Luv* space however, the matching function is more discriminatory, producing high agreement values for only colors that are perceptually close to red (lower



(a) (b)

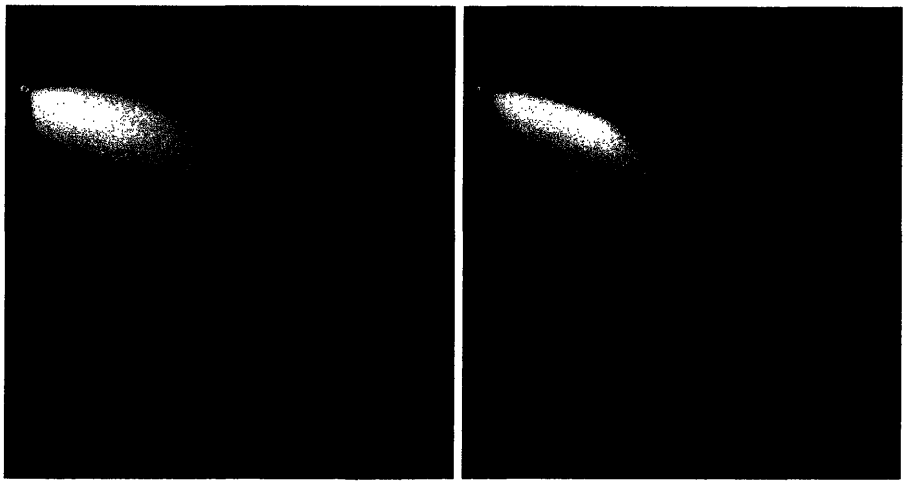
Figure 4.5. Matching function applied to pure red color in RGB space and other colors within the RGB cube. For a matching functions defined in RGB color space (a), the result is the quadrant of a Gaussian sphere. Matching in *Luv* space (b), we obtain only high agreement between those that appear perceptually red with greater discrimination than matching in RGB.

agreement values are produced than in RGB between red and colors with greater pink and orange hues (Figure 4.5). Agreement comparisons for other colors show similar signs of higher discrimination in *luv* over RGB.

In the third test, a color sample is chosen to contain equal portions of both blue and yellow. The agreement is measured between this color mixture and values that lie within the RGB cube, the results are shown in Figures 4.6. When the agreement is measured in RGB space, the result is the Gaussian ellipsoid encompassing both blue and yellow 4.6(a). It also matches well with colors containing green, magenta and orange. Moving to *Luv* however, we only obtain high agreement between those that appear perceptually blue, yellow, or a mixture of both with greater discrimination than matching in RGB 4.6(b).

■ 4.5 Comparison of Agreement Computation Methods

In order to test the accuracy of the agreement computation when colors are modeled as normal random variables, a number of tests were conducted on synthetically generated color patches. Agreements are computed between a chosen reference patch and a selection of target patches. The target patches are generated from the reference patch by either simulating a direct change in color (indicating a change in surface) or illumination. The aim of these experiments was to show that agreement matching in *Luv* is able to provide an improved matching score for changes in illumination while maintaining



(a) (b)

Figure 4.6. Matching function applied to mixture of blue and yellow colors in RGB space and other colors within the RGB cube. For a matching functions defined in RGB color space (a), the result is the Gaussian ellipsoid encompassing both blue and yellow. It also matches well with colors containing green, magenta and orange. Matching in *Luv* space (b), we obtain only high agreement between those that appear perceptually blue, yellow, or a mixture of both with greater discrimination than matching in RGB.

the ability to detect changes in surface. The effect of illumination changes produced by shadows are also investigated.

Simulated Color Change

To test the various methods of computing agreement for direct changes in color, we create a target patch whose color is equal to the reference patch with the addition of some corrupting color. Low agreement values are expected between the reference patch and target patches with greater proportions of the corrupting color thus indicating the ability to correctly detect differences in image color that actually originate from different surfaces in the scene. The graph shown in Figure 4.7(a) shown the change in agreement between the reference patch and target patch with increasing degrees of color corruption. The x-axis effectively represents the change in surface. The mean and covariance for each color patch is computed using Equations 4.4 and 4.3, followed by the computation of the color agreement (Equation 4.5). The agreement values are plotted together with those computed using thresholded linear distance (Equation 4.1) and exponential matching (Equation 4.2). Note that the probabilistic color agreement score for RGB and CIE-*Luv* space quickly falls to zero as the level color difference increases.

Figure 4.7(b) similarly graphs the probabilistic agreement in CIE-*Luv* space as a function of changing color. Here, the reference color and corrupting color are chosen at

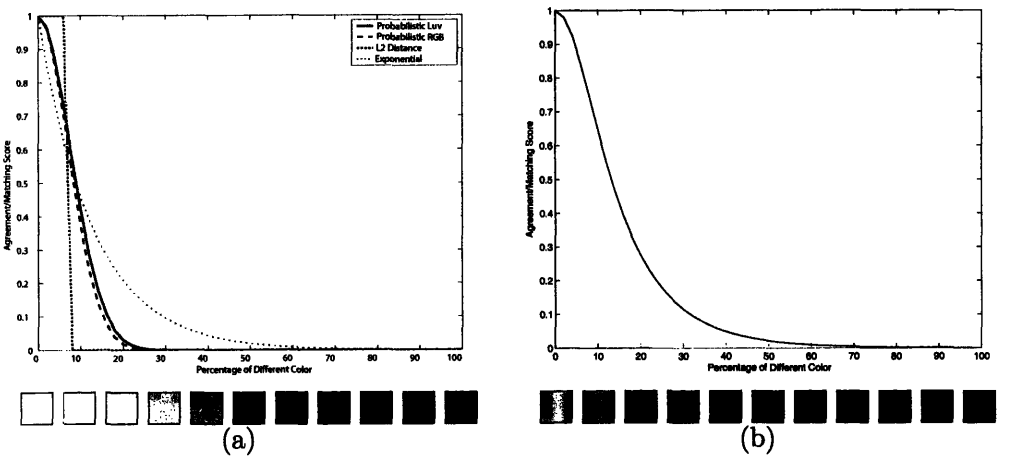


Figure 4.7. Graph of agreement versus change in color: A reference color is corrupted with various levels of a different color. Note that the probabilistic color agreement scores (Left) for RGB and CIE-*Luv* space quickly fall to zero as the level color difference increases. The graph to the right shows the change in agreement versus change in color between color pairs selected at random, the solid black line indicating the average change over all color pairs.

random. Agreement graphs are shown for fifty random pairs in yellow and the mean variation is shown in black. On average, the agreement between random color pairs falls as an exponential, as is expected from limiting theorems.

Simulated Matching Across Color Boundaries

Color matching techniques used in volume reconstruction algorithm often require the matching of regions of color. For all but the simplest of scenes, it is likely that some of these regions will fall on the boundaries of two or more surfaces, each with its own color. Consider the scenario of measuring the agreement of a source patch containing a reference color in one image, and a target patch containing both the reference color and some other color in a different image. We hope that our agreement computation will return some indication that portions of both source and target patches contain the same color. It is clear why using only the mean color of each patch would result in unreliable agreement values.

The various color agreement metrics presented earlier were tested on synthetically generated color patches. A series of target patches were created with varying proportions of a reference color and a different color. The agreement was computed between each pair and the results are shown in Figure 4.8(a). The reader should note that only the probabilistic matching techniques are able to provide reliable partial matching score when the target patch contains less than 50% of the reference color. Agreement computation using the probabilistic matching in CIE-*Luv* color space was tested separately on a series of random colors and the results are presented in Figure 4.8(b). The

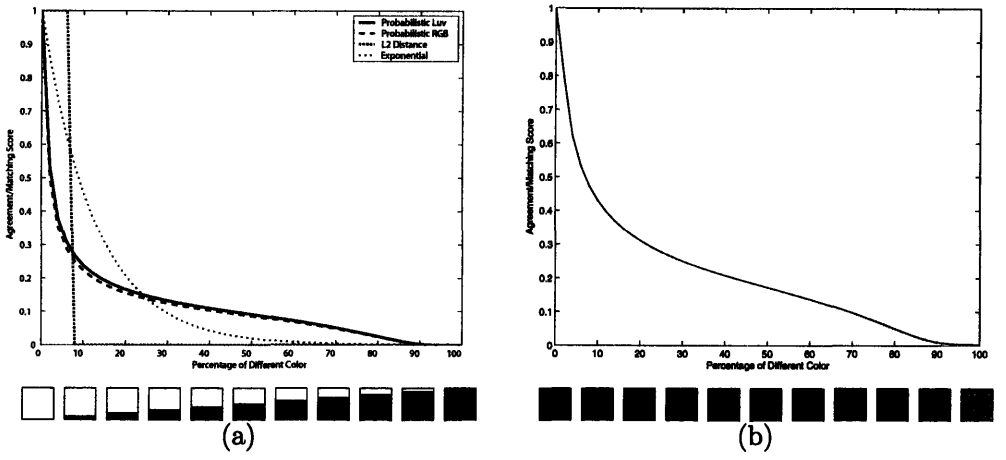


Figure 4.8. Measuring the degree of partial agreement score versus change in color. Note that the Probabilistic approaches maintain a partial match even when more the half the region is occupied by a different color. The graph to the right shows the change in agreement versus change in color between color pairs selected at random, the solid black line indicating the average change over all color pairs.

increase in agreement values can be attributed to the modeling of color variation using the covariance rather than just the mean color. The agreement effectively measures the overlap in the covariances and therefore enables matching between patches that contain even small regions of the same color.

Simulated Illumination Change

In order to test the variation in agreement during illumination change, the illumination of the source reference patch color is adjusted while holding the chromacity constant to form several target patches. The adjustment in illumination in this case is represented simply by a change in the overall brightness of the color although this is true in general. The relation between agreement and change in illumination is shown in Figure 4.9(a). Since color matching in CIE-*Luv* space is less sensitive to changes in illumination, the computed agreements are consistently higher than those in RGB space. This enables the correct agreement to be computed between colors even in the presence of small lighting variations on the surface of objects between images. Figure 4.9(b) shows the probabilistic agreement computation on synthetically generated color patches. The lines in yellow are for a single random color pair and the line in black indicates the average agreement over 50 random pairs.

Simulated Matching Across Shadow Boundaries

Consider now the case where our target patch falls on a shadow boundary; a region of the patch will directly resemble the reference color in the source patch. The remaining

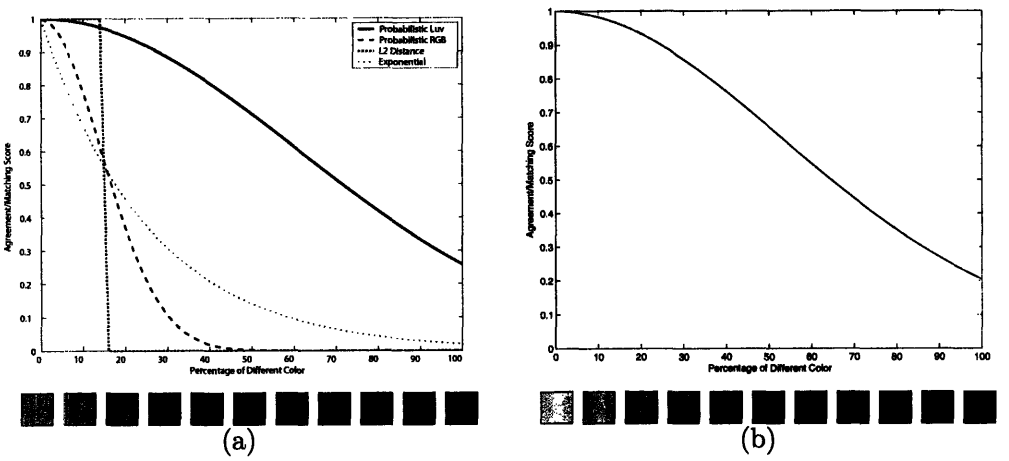


Figure 4.9. Simulating illumination change: The probabilistic agreement measured in CIE-*Luv* space maintains a higher level of agreement as the patch undergoes a simulated change in illumination. The graph to the right shows the change in agreement versus change in color between color pairs selected at random, the solid black line indicating the average change over all color pairs.

region will contain the reference color with a different (lower) overall intensity. By simulating the matching of colors, even in the presence of shadows, we ensure a more robust color matching scheme for application in volume reconstruction.

This case differs from the one described above in that the regions containing the shadow boundary will be represented by a distribution with greater variation along the illumination direction since they would contain portions of both colors rather than a single homogeneous color region.

Figure 4.10(a) shows the agreement between color patches in the presence of shadow boundaries. Again, the decreased sensitivity of the CIE-*Luv* color space to changes in illumination results in higher agreement scores than matching in RGB space when most of the target patch is in shadow. The change in agreement computed in CIE-*Luv* space for varying amounts of shadow in the target patch is shown in Figure 4.10(b).

Changes in Color and Illumination

Further experiments were conducted to test the computed agreement as functions of change in color and illumination. An image containing color variation (horizontally) and illumination variation (vertically) was generated and is shown in Figure 4.11(a). A reference color was chosen as the the center of the generated image and is shown in Figure 4.11(b). The agreement was then computed between this reference patch and every other color in the image. An ideal matching function would return a response in the form of high agreement as a vertical band in the center indicating low sensitivity to lighting change but good discriminatory ability between differing colors. Note that use of the probabilistic *Luv* color model (Figure 4.11(f)) results in higher agreement values

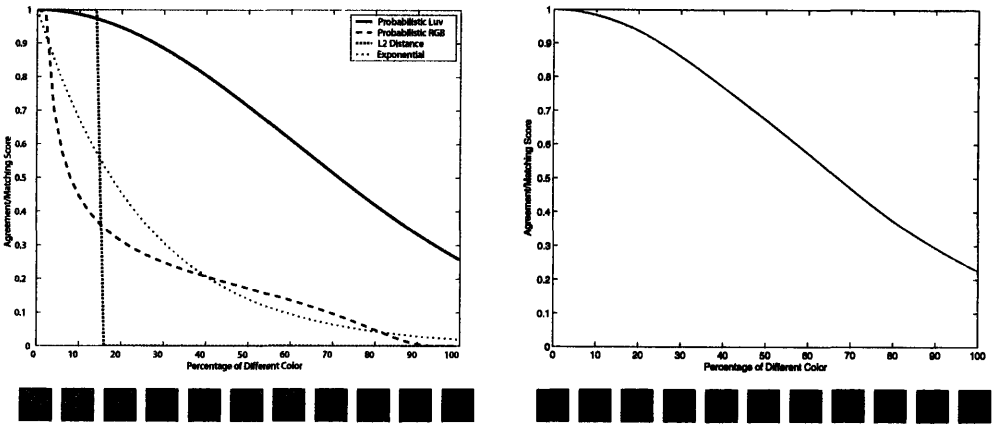


Figure 4.10. Simulating shadow boundaries: The probabilistic agreement measured in CIE-*Luv* space maintains a higher level of agreement as a shadow boundary moves across the target patch. The graph to the right shows the change in agreement versus change in color between color pairs selected at random, the solid black line indicating the average change over all color pairs.

between colors undergoing a lighting change and best approximates the desired vertical band when compared to the other tested agreement functions. The other methods tested showed only high levels of agreement in and around the reference color irrespective of whether the change is due to a change in color or illumination.

Testing Real Data

A series of real images containing the same building façade under differing illumination conditions were used to test the various methods of computing color agreement. An approximately homogeneous region of color is extracted from each image (Figure 4.12) and the agreement is computed between each color region and a reference color. The reference color is computed as the mean of all color regions in order to remove bias towards any one patch. The resulting agreements are graphed in Figure 4.13 together with values computed using Equations 4.1 and 4.2. For the tested patches, our method proves the most consistent in correctly returning high levels of agreement. A series of outliers were also tested and all methods correctly returned low agreement values for these patches.

■ 4.6 Summary

In this chapter, we have introduced a probabilistic color model that can be used to match colors in images in the presence of varying illumination and noise. The matching of colors across images provides the foundation of all consistency checks used in volume reconstruction algorithm and since these checks determine whether a particular voxel should be kept of discarded, the accuracy of color matching has a direct effect on the

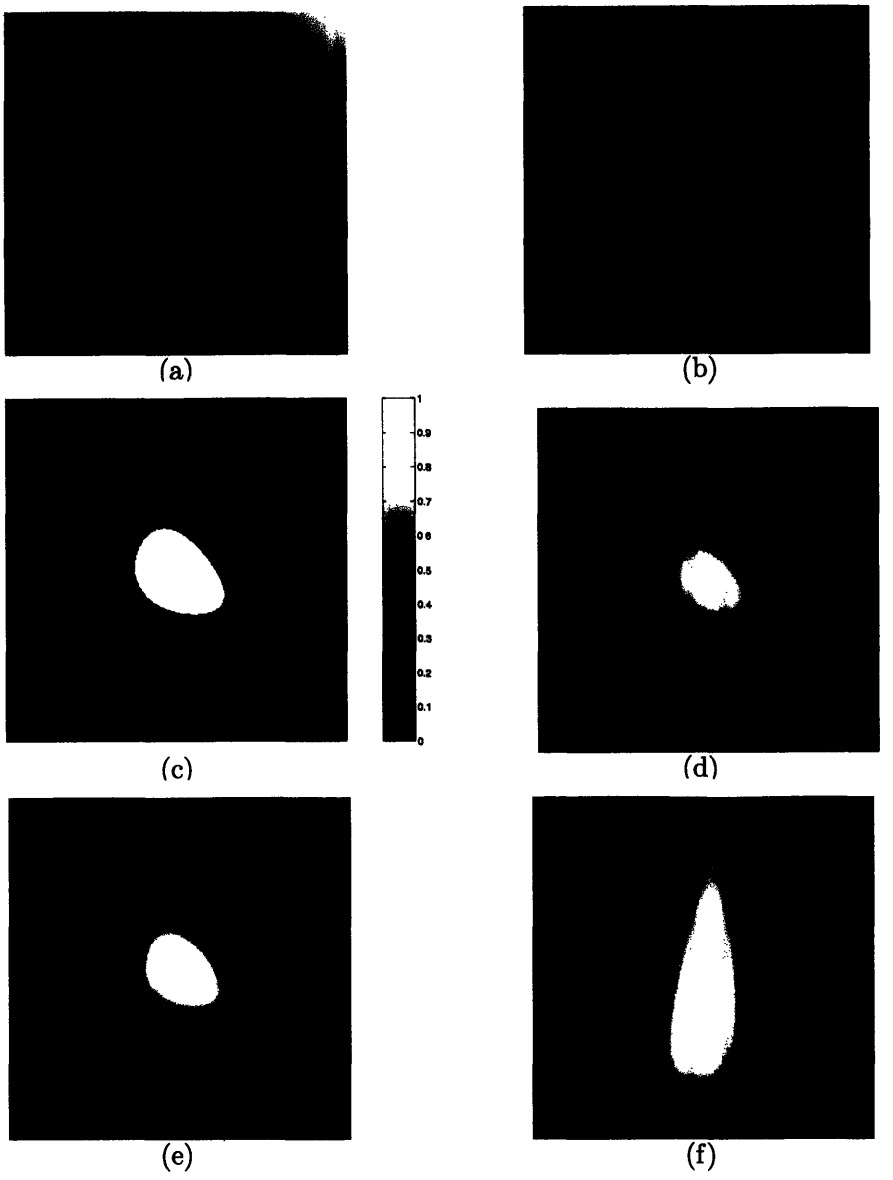


Figure 4.11. Synthetic Matching:(a) The Test image. (b) Enlarged view reference color (test image center). We match pixel colors in the test image against the reference color. (c) Linear thresholded match. (d) Exponential matching function used in [18]. Probabilistic RGB matching is shown in (e) and probabilistic CIE-*Luv* matching in (f).

quality of the final reconstruction.

When an estimate of the illumination (as described in Chapter 3) is available, a

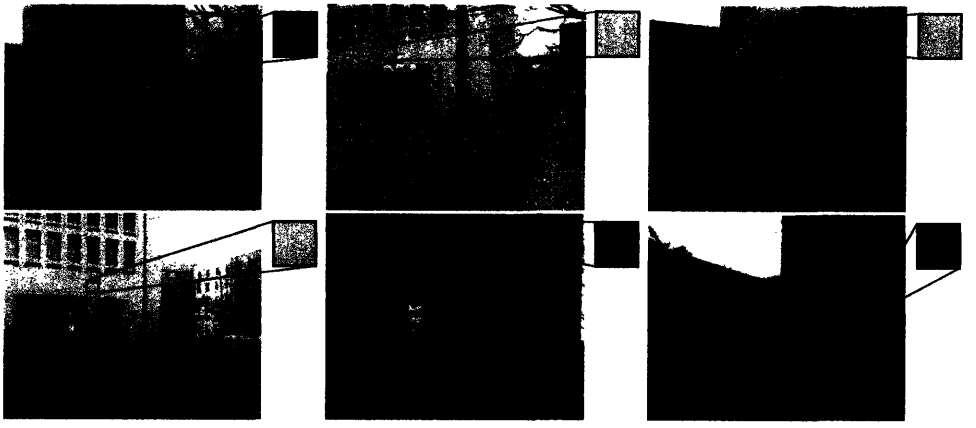


Figure 4.12. A region of color is extracted from a building surface imaged under various illumination conditions.

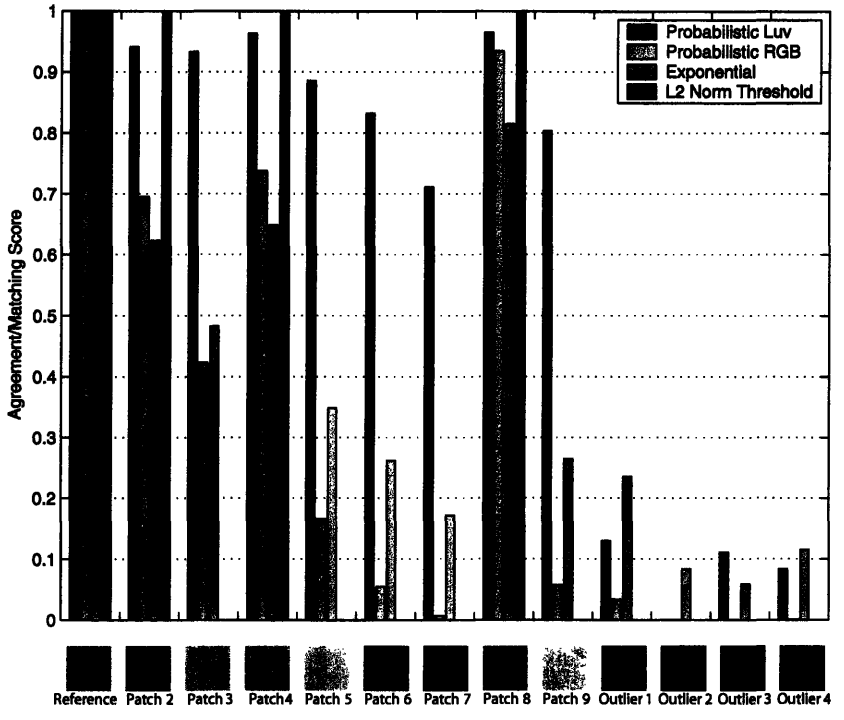


Figure 4.13. Comparison of agreement values between a reference patch and a series of other color patches corresponding to the same surface under different illumination together with some outliers. Agreements are shown for Probabilistic Matching in *Luv* Space, RGB Space, together with exponential (Equation 4.2) and Thresholded L_2 Norm (Equation 4.1) agreements.

first approximation to the surface reflectance under canonical illumination can be made by normalizing each color channel independently. These estimates are then placed in a probabilistic framework defined in CIE-*Luv* color space for comparison. These colors can be combined and manipulated in the same way as conventional color models computed as a simple mean, while also maintaining information regarding the distribution of colors from which they are derived. Matching colors statistically also allows us to compute partial matches between regions containing regions of common color. Results are given to highlight this new matching technique over traditional methods yielding improved results in the presence of changing illumination for both real and synthetic data.

This novel color matching algorithm is used in conjunction with the outdoor illumination estimator described in Chapter 3 as the basis for the overall volume reconstruction algorithm. In the following chapter, we look at optimization that can be made to the reconstruction algorithm before describing the complete implementation of the system in Chapter 7.

Chapter 5

Optimizations

The basic volume reconstruction algorithm as described in Chapter 2 can be enhanced using a series of practical optimizations to improve the overall efficiency. In this chapter, an analysis of the approximate complexity is given and compared to a more naive method. The optimizations are presented categorized as reductions in time (clock cycles) or space (memory requirements).

■ 5.1 Reductions in Time

With millions of voxels, and in our case possibly thousands of images, the time to process this information can quickly become significant. The problem is further compounded by the inter-dependencies between voxels. As described in Chapter 2, an iterative method is required to solve the problem. In this section, we examine the complexity of a naive voxel centric approach of processing the voxels and comparing it to our more efficient image centric approach.

Let us consider a simple algorithm for processing the voxels in arbitrary order. The pseudo code for the algorithm is given in Table 5.1. This approach proves to be computationally expensive and therefore time consuming. To illustrate this, let us consider a reconstruction from N views in a voxel grid containing a total of V voxels. Each opacity estimate requires the opacities of a further αV voxels for each node, where α is a constant that indicates the average number of occluders $O(i,j)$ for a voxel v_i visible in view j such that:

$$1 \gg \alpha = \frac{1}{VN} \sum_{j=0}^{N} \sum_{i=0}^{V} O(i,j)$$

The total number of voxel opacities required during each iteration is therefore $O(\alpha V^2 N^2)$. In this analysis, we see that every voxel must be processed or checked multiple times leading to wasted computation. Commonly used volume reconstruction algorithms avoid this problem by using a specific ordering of the voxels. If no voxels exist inside the convex hull of the camera centers, the voxels can be processed sequentially in an order that maintains the visibility constraints [67]. This means that the inner loop of the process over the occluding voxels is now unnecessary and the complexity is reduced

```
for each voxel v_i
   for each image I_j
      project v_i into I_j to find observation o_{i,j}
      find set of occluders V_{i,j}
      visibility[j] = 1.0
      for each occluding voxel v_p ∈ V_{i,j}
         visibility[j] = visibility[j] × (1 − occupancy(v_p))
      end
   end
   compute opacity(v_j) using visibility[j] and o_{i,j} ∀j
end
```

Table 5.1. Naive Approach: Pseudo code highlighting the algorithm for each iteration.

```
for each image I_j
   for each sample point s
      project s from I_j as ray r_{s,j}
      done=false
      visibility=1
      while(¬done)
         extend ray r_{s,j} to next voxel v_{s,j}
         if v_{s,j} is outside volume
            done=true
         else
            update occupancy(v_{s,j}) using visibility
            visibility = visibility × (1 − occupancy(v_{s,j}))
         endif
      end
   end
end
```

Table 5.2. Proposed Algorithm: Pseudo code highlighting the algorithm for each iteration

to $O(VN)$. To achieve the same complexity in time for arbitrary camera configurations, we adopt a camera-centric approach.

This method focuses on the fact that images are the source of information. This information can be efficiently packaged and propagated out into the discrete world. The voxels now act as placeholders of information combined from every available view. The

packaging of information is achieved by reducing each image to a set of samples that capture information about sets of neighboring pixels, each sample is then projected out into the voxel grid where this information is suitably combined and checked for consistency. This approach results in a reordering of the process loops from Table 5.1 such that the outermost loop is now over the images as shown in Table 5.2. This eliminates the need to compute the visibility product since each projected ray would carry its current state of visibility and observation responsibility as it propagates through the voxel grid. The number of samples per image can be considered to be constant since each image is likely to be of similar size and shape. The complexity of this new algorithm can be seen to be $O(VN)$.

■ 5.1.1 Image Representation and Sampling

Traditional voxel coloring algorithms work by projecting voxels into the images to assess their consistency. This can be considered a voxel centric approach and ensures that every voxel will be visited at least once. When consistency information is projected from the images, as with our image centric approach, there is no such guarantee. The chosen sampling in the image determines whether voxels will be visited once, many times, or in some case, not at all. Choose a sampling too sparse and voxels will be neglected (Figure 5.1(a)); increasing the density of samples fixes this problem but results in some voxels being processed multiple times by adjacent sample rays. This is especially true closer to the optical center where rays are packed tightly together (Figure 5.1(b)). One way of solving this problem is to use adaptive sampling that varies with distance away from the cameras optical center.

Adaptive Sampling

A fixed number of rays in the image leads to either under sampling of voxels far from the camera's optical center or an over-sampling close to it. We require a sampling that changes as it propagates through the volume. As each ray extends further from the optical center, it divides into multiple rays which together preserve the information of the parent ray. Each ray continues to divide and propagate through the voxel grid. In this way, by selecting when the ray division occurs, it is possible to ensure that every voxel in the volume is processed at least once while reducing multiple processing of the same voxel as with dense sampling (Figure 5.1(c)).

The division of rays results in the commonly used image pyramid [9,82] where as the depth into the volume increases, so does the resolution and level of detail as depicted in Figure 5.2. The effect is that of projecting the image pyramid into the volume.

The reconstruction volume is discrete in its make up and therefore the rays projected through the volume will be subject to aliasing. Some voxels (in particular those close to the camera) will be updated many times by projected rays from the camera. Each ray will enter and leave each voxel at distinct points on the voxel surface, thus resulting in a unique line segment for each ray-voxel pair. Each voxel represents a sampling of the volume at the position of the voxel center. The distance of each line segment from

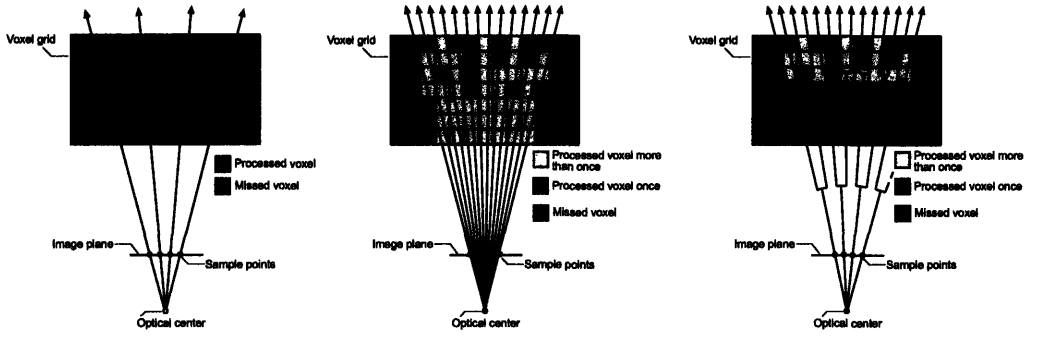


Figure 5.1. Reducing Complexity:(a) A sparse set of samples means that some voxels will be missed by the projected rays. (b) A dense set of samples will results in many voxels being processed several times. (c) Adaptive sampling enables total coverage while avoiding multiple processing of voxels.

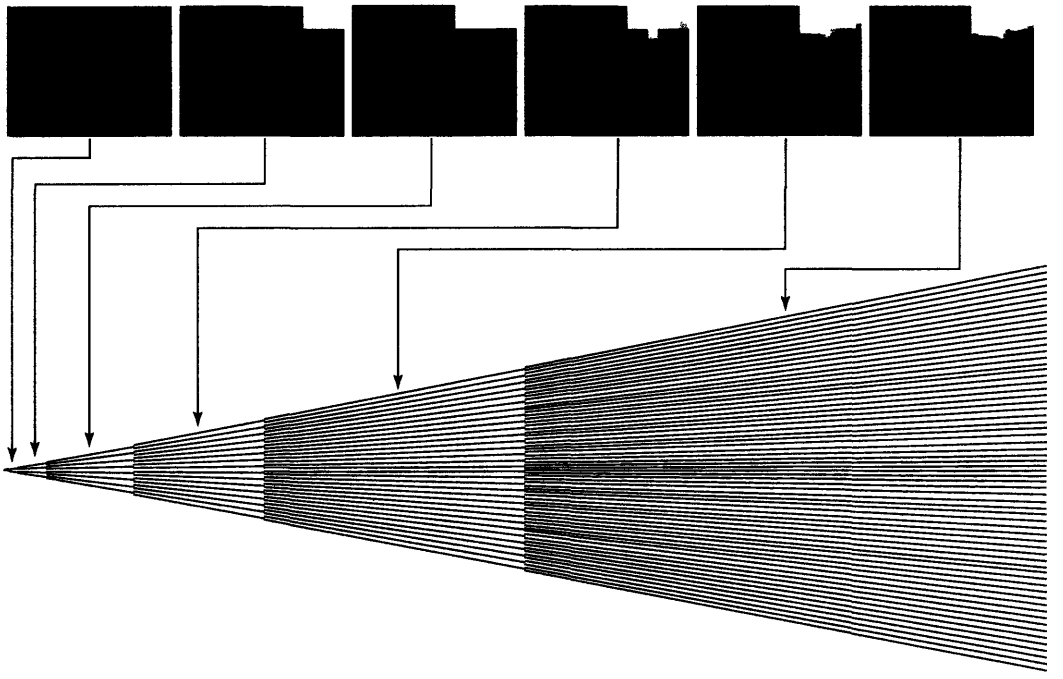


Figure 5.2. Example images from various layers in the image pyramid. The further the projection into the volume, the higher the image resolution.

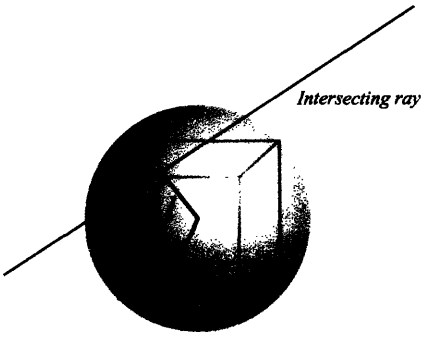


Figure 5.3. Computing the contribution weight of each ray to a particular voxel by examining the shortest distance from the ray to the voxel center.

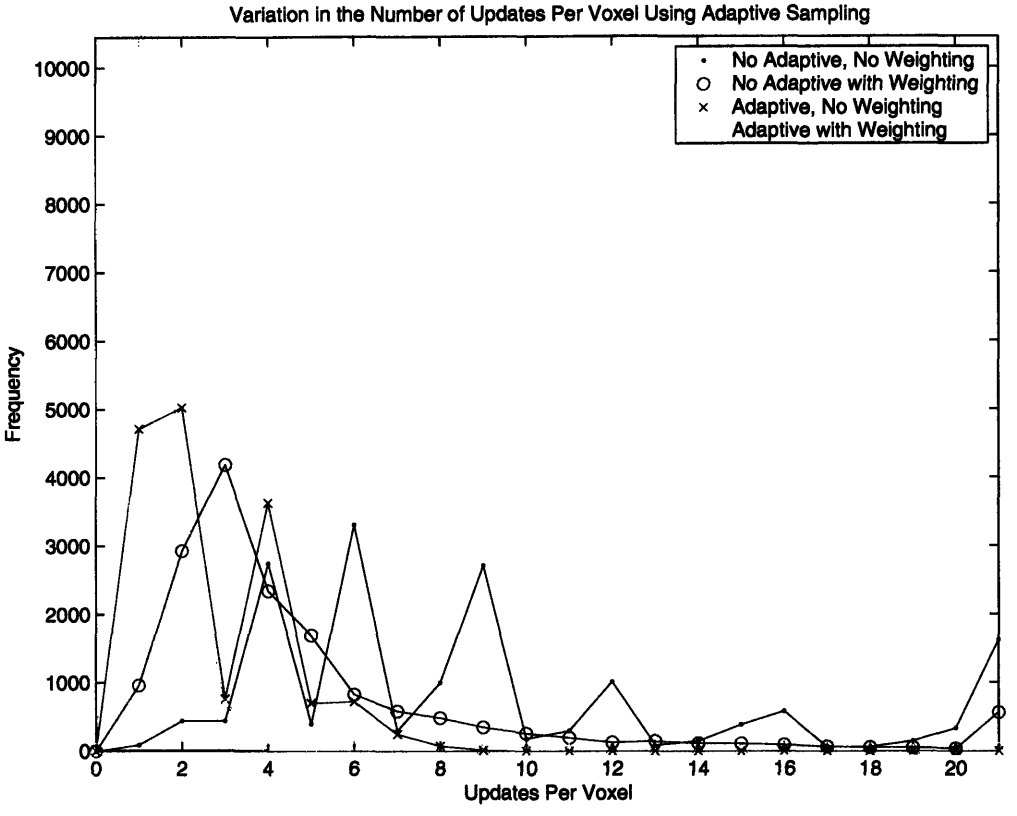


Figure 5.4. Graph of the frequency of the average number of times a voxel is updated by rays projected from the image. With adaptive sampling, most voxels can be seen to be updated only once.

the voxel center can therefore be used to weight the contribution of each ray to the voxel and by doing so, avoid the problem of aliasing. The weight is computed such

that rays passing through the center of the voxel fully contribute to it color and opacity computation, whereas those that pass through only a single vertex do not contribute at all. This weighting is achieved simply and directly by examining the distance of closest approach of the line segment and the voxel center as illustrated in Figure 5.3.

The effects of using adaptive sample are illustrated in Figure 5.4. When adaptive sampling is not used, the number of updates per voxel varies almost uniformly between once to over 20 times. The oscillations in the graph are caused by aliasing artifacts. When the contribution of each rays is appropriately weighted, the frequency peaks at around 3 updates per voxel, although some voxels are still process up to 20 times or more as indicated by the long tail of the graph. When adaptive sampling is used, the number of updates per voxel is limited to no more than 9 in this example. Again, the oscillations in the graph are due to aliasing. Once the updates are weighted, we see that the majority of voxels are updated only once and no more than 4 times for a particular image. This overall reduction in the number of updates directly translates into savings in processing time.

The decision of when to divide each ray is made according to the camera's internal parameters and the distance from the camera. Images taken by cameras with a narrow field of view can be used to avoid the problems associated with projecting information from the images into the volume. These cameras can be assumed to be almost orthographic and the projected rays are therefore approximately parallel. In these instances, ray division is unnecessary and dealt with automatically using adaptive sampling.

Adaptive sampling however does add complexity to the agreement and opacity computation discussed in Section 2.3.2. Visibility constraints must be maintained along each path through the volume. Agreements are computed and summed along each ray segment. The agreement of each voxel is then normalized to compute the responsibility according the maximum summed agreement over all combined rays through that voxel. This ensures that the summed responsibility along any valid sequence of rays is less that or equal to unity.

■ 5.1.2 Parallel Computation

The reordering of the processing loops in Table 5.2 also has another major advantage. The information from each view can be considered to be independent and therefore all views can be processed in parallel. The projection of each view can be handled by a different processor leading to an increase in speed that is directly related to the number of processors used. At the end of each iteration, the results from each processor are combined into a single consistent estimate for every voxel. These estimates can then be used in the next iteration. This implementation can therefore benefit directly from the ever increasing availability of computational power. A schematic of the parallelism of the algorithm is shown in Figure 5.5. The master processor handles the distribution of work to a set of node processor. Each node processor is able to process a single set of images. Once it has completed an image set or node, it informs the master processor, which will either respond with a new set of images or a wait signal indicating that no

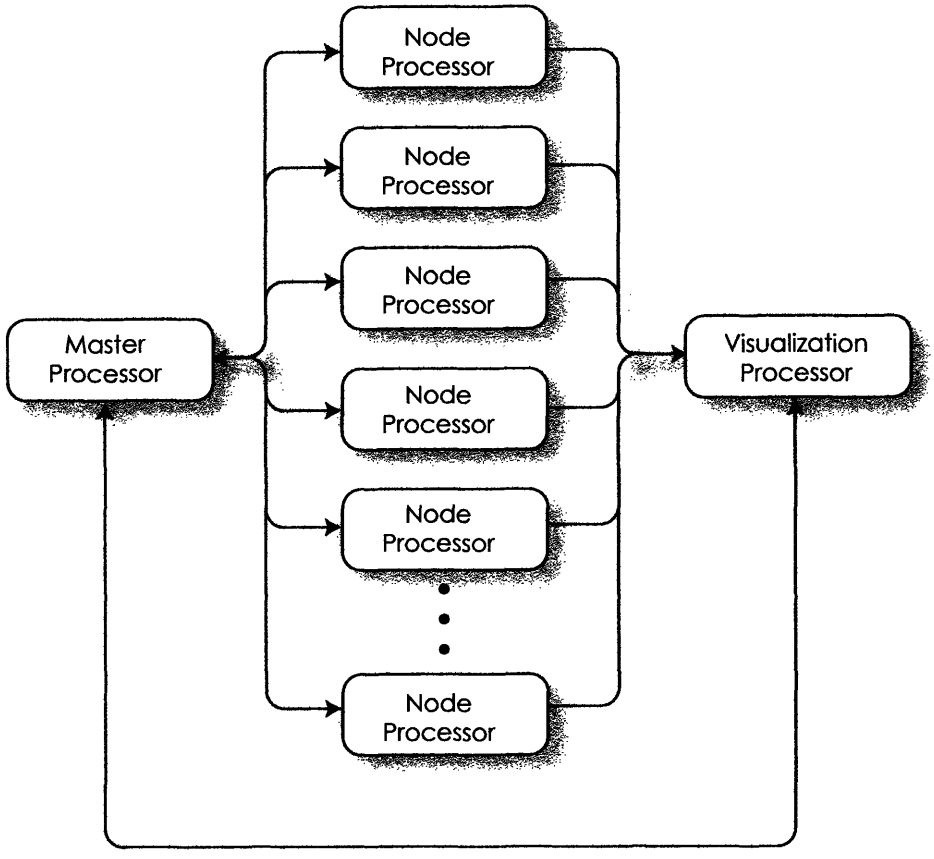


Figure 5.5. Schematic of Algorithm Parallelism: The master distributes work to the node processors, while the current status is displayed by the visualization processor.

more nodes require processing in this iteration. Both master and node processors communicate will the visualization processor which displays the current status of progress to the user along with the current processors in use.

■ 5.2 Reductions in Space

One of the main disadvantages of voxel based volume reconstruction algorithms is the extensive demand on memory. Memory requirement increase as $O(n^3)$ in the level of resolution such that a doubling in the reconstruction resolution leads to a increase by a factor of eight on the required memory. Despite the abundance of memory in high end computers, this demand can quickly become prohibitively expensive. In addition to the space allocation for voxels, extensive memory is also required for the possibly thousands of images that provide the input to the algorithm. In order to alleviate these problems, several space saving optimizations can be made and are discussed here.

5.2.1 Lazy Voxel Allocation

In our implementation, each voxel contains a variety of information such as color and opacity. Although the requirements of a single voxel is fairly small, storing this information in memory for even a low resolution model containing roughly one million voxels can quickly lead to problems. Others has avoided this problem by manually defining a tight bounding box around the object to be reconstructed thus minimizing storage of unused voxels. A fully automatic algorithm would have to sidestep this additional step.

One observation that directly leads to a space saving strategy is that we need only store information for voxels that are valid and that lie within the viewing volume. This is accomplished using a lazy allocation strategy. Since information is projected from the images and through the volume, voxels are only created if they are explicitly required. All voxels need only be defined during the first iteration of the algorithm since all subsequent iterations propagate information along identical rays. Although lazy voxel allocation avoids the unnecessary usage of memory, high resolution reconstructions are still significantly limited by the memory requirement.

5.2.2 Multi-Resolution Reconstruction

Multi-resolution methods have become increasingly popular as they can provide increased stability and speed of convergence for many algorithms. Most volume reconstruction algorithms however, have avoided the use of multi-resolution techniques due to problems associated with aliasing [65] and have instead opted to perform reconstruction only at the highest resolution. The aliasing arises in low resolution reconstruction due to increased relative size of each voxel to features within the image. These large voxels project to regions in the image that are much larger than a single pixel and therefore possibly multiple colors or surfaces. Since these algorithms use simple consistency checks based on color means, they are ill-equipped to deal with partial matching and therefore fail to work well at lower resolutions.

Partial matches can be achieved using a more sophisticated technique such as the probabilistic color matching described in the previous chapter, thus enabling multi-resolution reconstruction. The algorithm begins at the lowest resolution and continues until the volume converges. Voxels that converge to low opacity values are removed. The remaining voxels are then sub-divided and the iterated to convergence again. This cycle is continued until the desired overall resolution is achieved.

A two dimensional example of the multi-resolution reconstruction is shown in Figure 5.6. Figure 5.6(a-d) shows the evolving shape recovered at increasing resolutions. The large scale structure is identified during the first iteration (Figure 5.6(a)). Figure 5.6(b) shows the reconstruction at the double the original resolution. Notice that only the voxels on the boundary are reconstructed since they are responsible for the image observations. The highest resolution reconstruction is shown in Figure 5.6(d) which is a factor of 8 times higher than the original. For comparison, the ground truth for the synthetic example is shown in Figure 5.6(e).

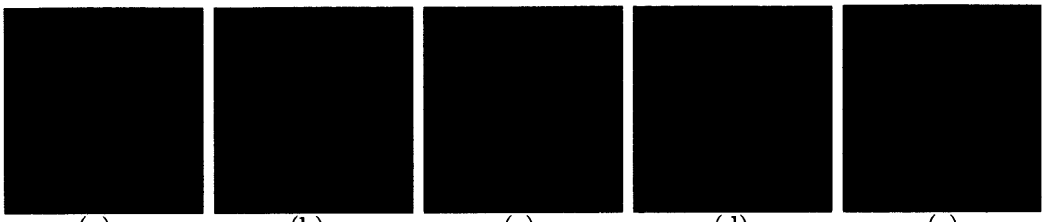


Figure 5.6. Multi-resolution 2D reconstruction: The evolving reconstruction is shown in (a-d) together with ground-truth (e)

■ 5.3 Summary

In Chapter 2, we presented a basic volume reconstruction based on the Roxels algorithm [18]. This approach was extended first in Chapter 3 where we described a method for automatically estimating outdoor (daylight) illumination from hemispherically tiled images. These illumination estimates were then used in Chapter 4, in a probabilistic color modeling and matching system, forming the basis for a consistency check commonly used in volume reconstruction algorithms. In this chapter, we examined various optimizations aimed at improving the overall efficiency of the reconstruction algorithm in term of memory usage and computational speed. Optimizations such as adaptive sampling in the images and parallel computation are used to reduce the computational time for the algorithm. The extensive memory requirements that are typically associated with voxel based reconstruction methods are curbed through the use of a multi-resolution method and lazy voxel allocation which prevents unnecessary voxels from ever being created. These enhancements are integrated into the volume, illumination and color estimators defined in the previous chapters to form a complete reconstruction algorithm. The following chapter describes how the estimated volumetric model can be used to directly to create a depth map per image and therefore, a surface representation of the scene. In Chapter 7, we outline implementation details of how these various modules are combined to achieve the goals set out at the beginning of this thesis.

Chapter 6

Depth and Surface Estimation

In the previous chapter, we described the various optimizations to the algorithm aimed at recovering a volumetric representation of a scene from calibrated imagery. The quality of the final reconstructed model is often subjectively based on simply how well this corresponds to the actual object. In this respect, voxel based models suffer from poor visual quality due to the limitation of using cubes as a building block for all possible models. The visual appearance of reconstructed models are often superficially improved through an appropriate use of texturing, where sections of the images are 'pasted' onto the surface to give the illusion of more surface detail and to hide imperfections. The techniques are usually successful in providing a greater sense of realism to the model which would otherwise be difficult to obtain. Unfortunately, voxel representations do not lend themselves very well to being textured. This is due to the discontinuities around the surface of each voxel. A further disadvantage of voxel based models is the demands they place on memory. A more concise, and more commonly used, representation for computer models describes the surfaces in the scene using triangulated meshes. These meshes can also be trivially textured for improving the asthetics of the model.

In this chapter, we present a method for obtaining a surface representation from the probabilistic voxel based scene produced using the methods already outlined in this thesis. Per image depth maps are obtained during voxel reconstruction in which the depth is modeled as a Gaussian random variable. Inconsistencies in the depth estimates due to texture-less or occluded regions are corrected using Bayesian Belief Propagation in the image. These depth maps can then combined to form a single, globally consistent surface of the scene although the actual implementation of this stage is left as future work.

6.1 Background

The imaging process can essentially be described as a projection of the 3-D world to a 2-D image. The depth of all visible objects in the scene are mapped to a single fixed distance in the camera known as the focal length. The recovery of the true depth for each point in the image naturally leads to a description of the original scene. Unless the dimension of an object in the scene is known, the depth can only be recovered up to a scale factor. This makes sense when we consider that images of objects twice the

size but imaged from twice the distance can appear identical to the original. Depth can be acquired via both active and passive sensors. We now give a brief review of both methods.

■ 6.1.1 Depth from Active Sensors

Recent advances in technology have seen an increase in the use of active sensors to obtain depth estimates.

Structured Light

A laser beam of known geometry is projected onto the surface of the object to be reconstructed. Images of the reflected light can then be used in conjunction with knowledge of the position and shape of the light emitter to triangulate the depth of the surface from the camera. The configuration of light emitter and camera means that structured light algorithms can practically only be used to compute depth for small objects and are not suitable for outdoor scenes.

Laser Range Scanners

Laser Range scanners emit pulses of light and measure characteristics of the returned signal to estimate depth. Although time of flight can be used to compute depth for object at large distance, objects at closer ranges examine properties such as phase or frequency shifts of lasers modulated with a low frequency signal for more accurate depth estimates. Early research with laser range sensors [55] produced an acquisition rate of 500ms per pixel. Today, it is possible to capture an entire high-resolution range images (500x500) at a similar rate. The sensors can often be large and expensive encouraging research to obtain depth from from smaller, cheaper, and therefore more prevalent sensors such as images.

■ 6.1.2 Depth from Passive Sensors

Stereo vision is the process of acquiring 3-D range information about a scene from two or more images taken from different viewpoints. This is similar to the human visual system where the different perspectives of our two eyes result in a slight displacement of the scene in each of the two monocular views, and permits us to estimate depth. Computer stereo vision is a passive sensing method which is based on triangulation between the pixels that corresponds to the same scene structure being projected onto each of the images.

Optical flow

This method computes the motion of points in the scene using the simple assumption that the image intensities $I(x, y, t)$ remain constant over time. The velocity of a point in the image can then be estimated using the brightness change constraint equation [32].

Optical flow methods can therefore be used to obtain per pixel point correspondences between images which in turn used to compute depth depth from two or more images in a sequence. This assumption is only true for small displacements and fixed scene illumination. The small displacements in the camera however result in greater errors in depth estimate from triangulation.

Dense Correlation

The availability of two or more views of a scene enables the computation of depth via triangulation. These views can be obtained using multiple cameras distributed throughout the scene, allowing images to be acquired from several viewpoints simultaneously. Alternatively, for static scenes, a single camera can be moved through the world, taking images as it moves, to the same effect. In [59], they opt for the later, using an uncalibrated hand-held camera to estimate depth. They use epipolar geometry and random sampling consensus (RANSAC) techniques to assist in feature correspondence. These correspondences identify 2-D points in the images that are derived from a single 3-D point in the world. These points are then used to recover the relative positions and orientations for every camera in the sequence. The images are then rectified, where all image planes are mapped to a single reference plane to facilitate depth estimation. Once rectified, regions in any one image can be correlated against all possible corresponding regions in the next image in the sequence. Candidates for possible correspondences are identified through an epipolar search. Once the correct correspondence is found, the depth can be estimated via triangulation. These estimates can then be smoothed and combined with other views to construct a piecewise surface model of the scene.

The short base-lines between images in a video sequence enables accurate correspondences to be found automatically since adjacent frames appear very similar. Short base-lines have the disadvantage however that all else being equal, they lead to inaccurate depth estimates when compared to images from wider baselines. Panoramic images [35] can be used to overcome this problem by exploiting their wide field of view to benefit from accurate correspondence along the baselines while also maintaining accurate depth estimates perpendicular to them.

Multiple synchronized cameras arranged on the inside of a dome are used in [34] do obtain dynamic depth maps from multi-baseline stereo. The results are impressive but limited to objects that can be placed inside the five meter diameter dome.

Robust dense correlation methods that make full use of multiple images can be computationally expensive. Matching regions simultaneously in n images has an order of growth $O(k^n)$, where k is a constant, and therefore can quickly become impractical.

In this chapter, we present a method which utilizes the volumetric reconstruction algorithm described previously to compute dense depth maps for the scene. We exploit the opacity and view specific responsibility information in the voxels to obtain an estimate of the depth as a normal random variable for every sample in the image. These initial depth estimates are then placed within a pairwise Markov Random Field and improved through the use of Bayesian Belief Propagation. The estimates are fi-

nally projected out into the 3-D world to give a surface representation for the surface. Finally, the surface can be textured to improve the visual impact of the results.

■ 6.2 Depth Estimation

In order to compute the depth per image sample, let us make the assumption that every sample in the image has a single corresponding depth which is true for a scene containing only opaque objects.

Returning to the notation used in Chapter 2, let $< u, v, j, d >$ be the 3-D world point obtained by projecting a 2-D image point with coordinates (u, v) in image j a depth d from the optical center. Recall that the view dependent responsibility $r_j(< u, v, j, d >)$ describes the contribution of a voxel at $< u, v, j, d >$ to image j. In addition to using the responsibility to compute the voxel opacity and color, we can also make use of it to compute the mean depth $D_j(u, v)$ associated with the point (u, v) in the image:

$$D_j(u,v) = \frac{\sum_{d=0}^{d_f} w_j(< u, v, j, d >)d}{\sum_{l=0}^{d_f} w_j(< u, v, j, l >)}$$

where $w_j(< u, v, j, d >) = r_j(< u, v, j, d >)\alpha_j(< u, v, j, d >)$. In words, this relation weights each depth along the ray from (u, v) according to the product of responsibility and opacity of the voxel at that depth. Similarly, the variance $\sigma_j(u, v)$ in the depth can be computed

$$\sigma_j(u,v) = \frac{\sum_{d=0}^{d_f} w_j(< u, v, j, d >)d^2}{\sum_{l=0}^{d_f} w_j(< u, v, j, l >)} - (D_j(u,v))^2.$$

These depth estimates can be computed for every sample in the image once responsibility and opacity estimates in the image have converged. These initial estimates of depth will no doubt contain errors. Uncertainties in the volume reconstruction result in corresponding uncertainties in depth. The non-uniformity of these uncertainties will mean that we have accurate depth information for some areas of the image while other regions will suffer from ambiguities. This non-uniformity can be exploited to compute more accurate depth estimates throughout the image. By making simple assumptions about neighboring estimates, reliable depth can be propagated through the image using Bayesian Belief Propagation.

6.3 Bayesian Belief Propagation

As was shown by Weiss [81], reliable estimation of local scene properties such as depth can be achieved by propagating measurements across images. The central idea behind Bayesian Belief Propagation (BBP) is that an image consists of many interconnected hidden units and observations of each of these units. Each of the hidden units can transmit and combine information according to probability calculus. The objective is to maximize the posterior probability of the hidden unit values given the observations.

To this end, with our depth modeled as an observable Gaussian random variable, we can propagate these estimates throughout the image to obtain the true underlying depth. We begin with a brief overview of Bayesian modeling before presenting details of computing reliable depth from our estimates using BBP.

6.3.1 Basics of Bayesian Modeling

A Bayesian model, as described in [72], is the statistical description of an estimation problem. The description is the combination of two components, the *prior* model and the *likelihood* or sensor model to form a third component, the *posterior*. The *prior* model $p(\mathbf{z})$ captures known information about the surroundings before any data is acquired. The *sensor* model $p(\mathbf{D}|\mathbf{z})$ describes the probabilistic relationship between the sensed measurement $\mathbf{D}$ and the desired hidden data $\mathbf{z}$. The *posterior* then combines the *prior* and *sensor* models using Bayes' rule

$$p(\mathbf{z}|\mathbf{D}) = \frac{p(\mathbf{D}|\mathbf{z})p(\mathbf{z})}{p(\mathbf{D})} \propto p(\mathbf{D}|\mathbf{z})p(\mathbf{z})$$

The *posterior* can then be used to infer the most probable estimate of $\mathbf{z}$ given the sensed measurement data $\mathbf{D}$. In reference to low level vision problems, the sensed data $\mathbf{D}$ usually corresponds to a computable function of the image intensities. Given an image (or sequence of images), the goal is to infer some hidden information about the imaged scene such as object motion or scene depth at each pixel. The prior model is used to describe knowledge about the problem before the image is examined. In our case, we make the regularization assumption that neighboring depth values are likely to be similar. The process by which each hidden variables (e.g depth) gives rise to corresponding observable quantities in the image is captured by the sensor model. Our objective is to combine these models to form the posterior, and then find the values of the hidden variables (true depth) $\mathbf{z}$ which maximizes it.

For the problem discussed in this chapter, an image is defined as a network of observable nodes $\mathbf{D}$ which correspond to our initial estimates of depth. These nodes are related to a set of hidden scene nodes $\mathbf{x}$. We assume that each observable measurement node D_i (where the index i corresponds to a particular (u, v) coordinate in the image) is a function of the corresponding hidden node z_i. In order to impose some structure on the network, each hidden node z_i is dependent on one or more of its neighbors which we shall call the neighborhood N_i. This dependency can be described using the marginal

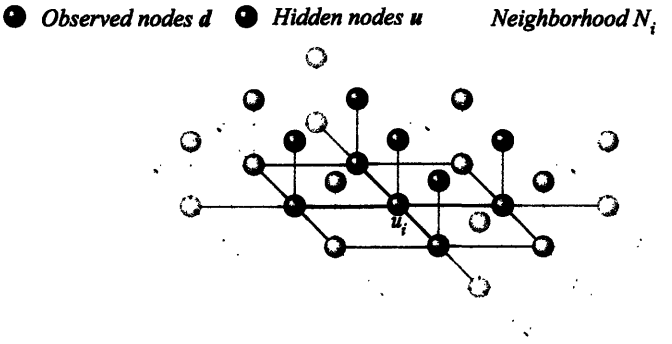


Figure 6.1. 2D Square lattice pairwise Markov Random Field (MRF) showing the relation between observed (sensed) nodes and hidden nodes. The neighborhood N_i of node u_i using the Ising model is also shown

probability

$$p(z_i|\mathbf{z}) = p(z_i|\{z_k\}), z_k \in N_i$$

The model just described is known as a pairwise Markov Random Field and is a well established model in the field of computer vision [27,43]. Pairwise in this case refers to fact that the network is made of pairs of observation and hidden nodes.

■ 6.3.2 Pairwise Markov Random Fields

Pairwise Markov Random Field (MRF) theory provides a convenient way of modeling context-dependent entities within a discrete field such as an image. A common MRF used to represent images is referred to as the Ising model shown in Figure 6.1 in which the neighborhood of node z_i consists of its nearest neighbors N_i.

The interaction between neighboring hidden nodes is defined by the prior. The sensor model describes the relationship between image intensities (or function thereof) representing observable node data D_i and each hidden node z_i. Markov Random Fields can therefore be used to estimate the distribution of the posterior using these nodal interactions.

■ 6.3.3 Prior Model

Knowing which solutions to a problem are more probable *a priori* can often guide the result. Prior models are used to indicate these more probable solutions and in doing so provide essential constraints when solving ill-posed problems. As mentioned earlier, images in low level vision problems can be represented as an MRF in which the prior describes the relationship between neighboring hidden nodes in the network. This relationship is often one of similarity such that hidden nodes in close proximity to one another are assumed to take on approximately the same value.

In general, a pairwise MRF network can be considered an undirected graph and the prior can therefore be defined using a compatibility matrix [84] $\psi(x(u,v), x(u',v'))$. Making the assumption that neighboring units will likely represent similar depths in the scene, we can use a Gaussian prior where the compatibility matrix is given by

$$\psi_{ij}(z_i, z_j) = exp(-\frac{[z_i - z_j]^2}{2\sigma_p^2}), \quad if\ z_j \in N_i$$

$$\psi_{ij}(z_i, z_j) = 0, \ otherwise$$

The variance σ_p^2 can be considered a parameter that controls the degree of fitting; small values of σ_p^2 indicates strong dependence between neighboring nodes resulting in a flattened (smoothed) output. Conversely, for larger values of σ_p^2 the ties between nodes are loosened, resulting in over-fitting to the observed data.

■ 6.3.4 Sensor Model

The fact that sensed data is seldom perfect has forced the necessity to model sensor errors stochastically. The knowledge that a sensor is consistently biased may be corrected once each measurement has been taken. Similarly, knowing in which measurements one should have more or less confidence can be of great use. A sensor model relates local evidence from sensor data at node D_i to corresponding hidden node x_i being estimated. The most commonly used sensor model, primarily due to its simplicity and well understood nature, is the Gaussian. Noisy measurements defined by a Gaussian can be completely characterized by its mean and covariance. A Gaussian sensor model relates the underlying hidden node x_i to the observed sensor node D_i using the conditional probability distribution

$$\phi_i(z_i) = p(D_i|z_i) \propto exp((D_i - z_i)^2/2\sigma_i^2).$$

The variance σ_i^2 defines the noise in the sensor and is dependent on the type of sensor being used. The computed variance in our depth estimate $\sigma(u,v)^2$ is the error in our 'sensor' and can therefore be used directly.

Having presented the basics of BBP, we now examine how it can be used to estimate the marginal posterior distribution for each hidden node in the MRF. In the BBP framework, each hidden node in the network is thought to transmit messages neighboring nodes. Each message carries statistical information including the relation between source and destination nodes of the message. Incoming messages to a node can then be combined using probability calculus to estimate the marginal posterior probability known as the belief. Outgoing messages from a node summarize the statistical information in incoming message. In this way, messages propagate local information throughout the network.

Using the notation of compatibility matrices $\psi_{ij}(z_i, z_j)$ and $\phi_i(z_i)$ defined in Sections 6.3.3 and 6.3.4, each message $m(z_i, z_j)$ from node z_i to node z_j is defined as follows:

$$m_{ij} \leftarrow \beta \sum_{z_i} \psi_{ij}(z_i, z_j)\phi_i(z_i) \prod_{k \forall z_k \in N_i, k \neq j} m_{ki}$$

where β is a normalization constant. The marginal posterior probability (belief) in node z_i is obtained by taking the product of all incoming messages to that node and the local evidence $\phi_i(z_i)$ i.e.

$$b_i(z_i) \leftarrow \alpha \phi_i(z_i) \prod_{k \forall z_k \in N_i} m_{ki}.$$

Both compatibility matrices are Gaussian, and therefore the messages and beliefs must also be Gaussian. This Gaussian belief $b_i(z_i)$ is maximized, when z_i is equal to the mean of the distribution. Denoting the message $m_{ij} \sim \mathcal{N}(\mu_{ij}, \sigma_{ij})$ leads to the following maximum a posteriori (MAP) estimate for each node:

$$z_i \leftarrow \frac{\frac{D_i}{\sigma_i^2} + \sum_{j \forall z_j \in N_i} \frac{\mu_{ji}}{\sigma_{ji}^2}}{\frac{1}{\sigma_i^2} + \sum_{j \forall z_j \in N_i} \frac{1}{\sigma_{ji}^2}}. \tag{6.1}$$

The mean and variance can be computed through a series of updates, the update rules for incoming messages to node z_i can be shown to equal

$$\mu_{ji} \leftarrow (\frac{D_j}{\sigma_j^2} + \sum_{k \forall z_k \in N_j, k \neq i} \frac{\mu_{kj}}{\sigma_{kj}^2})(\frac{1}{\sigma_j^2} + \sum_{k \forall z_k \in N_j, k \neq i} \frac{1}{\sigma_{kj}^2})^{-1}$$

$$\sigma_{ji}^2 \leftarrow \sigma_p^2 + (\frac{1}{\sigma_j^2} + \sum_{k \forall z_k \in N_j, k \neq i} \frac{1}{\sigma_{kj}^2})^{-1}$$

The messages for each node in the network can be updated in parallel or asynchronously using the BBP update rules described. With each update of all the nodes, the messages carry an increased amount of pooled statistical information and therefore provide more global information. In this way the hidden nodes z_i are iterated to converge at the the MAP estimate of depth from the observed measurements.

The results of computing depth from the volumetric models and the subsequent effect of perform BBP on these estimates is presented in chapter 8.

■ 6.4 Summary

The basic volume reconstruction algorithm described in Chapter 2 in conjunction with the extensions for handling illumination variation (Chapter 3) and probabilistic color matching strategy (Chapter 4) can be used to recover a volumetric model of an imaged scene. In the previous chapter, we described a number of optimization to the algorithm to improve the computational complexity in both time and space.

In this chapter, we described a method for extraction of depth estimates from the recovered volumetric model. From the standpoint of an end user of the algorithm, this provides greater flexibility in terms of the usability of the final model. The volumetric model contains uncertainties and associated depth is therefore modeled as a Gaussian random variable to capture these uncertainties. We have described a method of using Bayesian Belief Propagation to reduce the errors in depth by iteratively combining local information. The depth estimation is the final stage of the described reconstruction algorithm. In the next chapter, the complete implementation of the algorithm is discussed from input images to final surface model. In chapter 8, we present the results of performing the algorithm on a variety of real and synthetic datasets including those from the City Scanning project.

Chapter 7

Implementation

In Chapter 2, we presented a basic volume reconstruction based on DeBonet and Viola's Roxels algorithm [18]. Chapter 3 described a method for automatically estimating outdoor (daylight) illumination from spherically tiled images. These estimates were then used in Chapter 4, in conjunction with a probabilistic color modeling system, as the basis for an improved color matching strategy to be used in the volume reconstruction algorithm. In Chapter 5, we described optimizations to improve the algorithmic orders of growth in both time and space. The previous chapter described how, given the volumetric model, it was possible to estimate surface depth for every sample in the image. In this chapter, we proceed with a more detailed look at the implementation used to combine these various pieces and achieve the desired reconstruction of outdoor urban environments.

■ 7.1 Initialization

Our system is initialized based on the input images. The pose and orientation information, provided as part of the calibrated images, are first used to define the reconstruction volume. Each image is reduced to a collection of samples ready for projection. We then proceed with the first iteration of the algorithm by projecting these samples into the volume. The voxels in this volume are created and initialized at this time, ready for subsequent iterations. The sample rays end their journey through the discretized volume on the surface of a sphere whose radius is equal to the far distance. This far distance is currently a variable defined by the user but could also be computed through consideration of camera placement and relative baseline lengths. The samples on this sphere are then used to estimate the initial illumination conditions. These initializations are described in more detail below.

Reconstruction Volume

The position of each camera can be used directly to compute the size and scale of the reconstruction volume. Each camera is assigned a far distance, beyond which it will have no effect on the volume. This effectively defines a sphere of influence for each camera. Further refinement to this initial volume can be made by also considering the camera orientation and internal parameters. Both internal and external parameters are

used to form the camera's view frustum. The convex hull of all view frustum volumes is then used to define the initial reconstruction volume. This volume can then be divided coarsely into an initial set of potential voxels.

Image Sampling

The image sampling also begins at the coarsest level, with the top level sample comprising the entire image. The image is then divided into four sub-regions. The divisions continue until the size of the sample region reaches a limited case. This limiting sample size L is computed using the far-distance D and focal length f for the camera and the voxel size s. Using projective geometry we have:

$$L = \frac{f}{D}s.$$

This limit on the sample size defines the maximum sampling density in each image such that the ray spacing at the far distance is equal to the voxel size. This ensures at least one sample ray will interact with each voxel within the region of influence of the image. Each sample encoded the color statistics (mean and variance) of the pixels within its region, and the sample ray direction computed by projecting the point at center of the region out into the world.

Samples are then projected through the volume starting with the top level sample. The next set of samples are used at depth z, such that:

$$z = \frac{s}{l}f$$

where l is length of the sample region diagonal. This choice of the depth z, ensures that the sample ray division occurs before they become too sparse and fail to hit every voxel in the region of influence. The density of rays therefore adapts to the distance from the camera.

Voxels

During the first iteration of the process, each sample propagates and divides through the volume, passing through regions of possible voxels. If it encounters a region that has not be initialized as a voxel, it does so and then deposits its respective color at this voxel. For voxels encountered that have already been created, the current voxel color is combined with the sample ray color as defined in Section 4.2.1. Once all images have been processed, every 'visible' voxel in the volume is initialized and contains the color of all views that might possibly observe it. This color serves as our initial estimate for the global color.

Background Illumination

Once the samples have been propagated through the voxel volume, the sample color, together with its corresponding ray direction, is used to update a region on a trian-

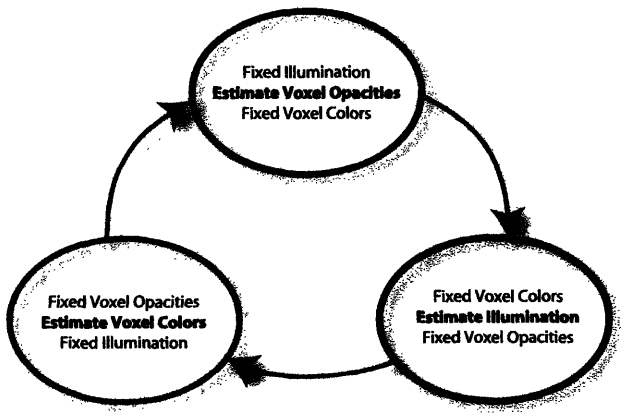


Figure 7.1. Schematic of Estimation Algorithm: The voxel opacities, illumination and voxel colors are estimated in turn. While each variable is estimated, the other two are held constant.

gulated sphere; the same sphere described in Section 3.3 to model the illumination. These colors provide the input for the illumination estimation described in Chapter 3. The various masks are applied to remove regions that correspond to high variance, low luminance or unlikely sky colors. The remaining region colors are used to estimate the entire sky model through non-linear optimization. This estimate of the illumination conditions, together with the foreground/background mask created, is later iterated as more information becomes available regarding the shape of the scene.

With these initial estimates for the voxels and illumination conditions, we can proceed with the iterations to recover the shape and color of the scene from the images.

■ 7.2 Iterative Scene Volume Estimation

The shape and color of the scene is estimated using an iterative scheme along with the illumination for each node (image set). The algorithm is shown schematically in Figure 7.1. Each of the scene variable is estimated while keeping the others fixed.

■ 7.2.1 Opacity Estimation

Given the initial color estimates for every voxel, together with the sky illumination model parameters for every node, the opacity of each voxel is estimated using the outlined in Chapter 2. An opacity estimate is made with respect to each image that observes it. Since there is no dependency between the images, they can be processed in any order or even at the same time as with the parallel implementation described in Chapter 5. These estimates are then combined to form a global consensus estimate of the opacity ready for the next iteration.

7.2.2 Illumination Estimation

The initial estimate for the illumination is updated during each iteration of the algorithm. As the estimates for the volume opacity (and therefore shape) improve, samples propagated through the volume that do not interact/intersect the volume must effectively be *looking* at background or sky, and can therefore be used to update the background model. The separation between samples used to update the voxel opacities and those used to update the sky illumination model lead to an image based segmentation of foreground and background. This segmentation could then be used in much the same way as with algorithms that require background subtraction, restricting the worst case reconstruction to that produced by shape from silhouette techniques.

7.2.3 Color Estimation

Once an estimate of the illumination intensity is available, it can be used to approximate the surface reflectance under canonical illumination using the diagonal model described in Section 4.1.3. If the mean color of sample is given by $S = (S_r, S_g, S_b)$ and the mean illumination intensity is $E = (S_r, S_g, S_b)$, the surface color ρ can be estimated as follows[1]:

$$\rho = (\rho_r, \rho_g, \rho_b) = (\frac{S_r}{E_r}, \frac{S_g}{E_g}, \frac{S_b}{E_b})$$

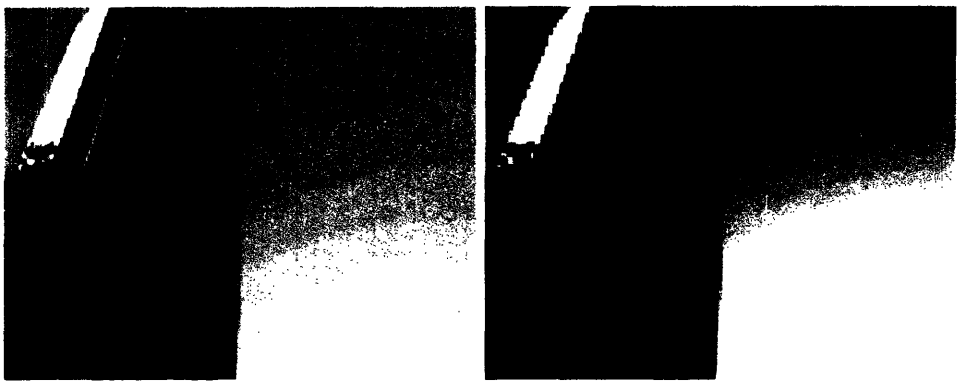
The illumination model for each node is integrated over the hemisphere and mean intensity is factored out of the samples of all images in the corresponding node. An example of the sample colors before and after this normalization is shown in Figure 7.2.

Since no information is directly available for surface within the scene, this forms the best guess at the color of the surface under canonical illumination. The color corrected samples are then used in the next iteration. This cycle of estimating some parameters while others are held constant is continued until convergence is attained.

7.2.4 Multi-Resolution Reconstruction

The voxels that converge to low opacity values are removed. The remaining voxels are subdivided and the process in repeated again. In practice, low resolution reconstructions can sometimes lead to a few voxels being incorrectly removed at the end of each stage in the multi-resolution process. This effect is be attributed to the imperfection of the partial matching system. Since voxels are only created during the first iteration, and at the coarsest voxel size, the false removal of voxels can lead to recovered models that contain holes or missing regions. This problem of *lost* voxels can be alleviated by convolving the reconstruction volume after removing low opacity voxels [65]. This

[1]The reader will note that this normalization is computed in RGB color space since the diagonal model does not directly translate to CIE-*Luv* color space



Before Normalization After Normalization

Figure 7.2. Color under canonical illumination: Image sample colors before and after normalization by the estimated illumination.

expansion of the reconstruction volume is performed before subdividing, and ensures the completeness of the high resolution models.

■ 7.3 Depth Estimation

Once the volumetric model has converged at the highest required resolution, it is used as input to the depth estimator as described in Chapter 6. The initial estimates for the depth mean and variance are computed by projecting samples from the image and examining the opacity and responsibility of voxels along each sample ray. These estimates are then used directly as the values associated with the observable nodes of a Pairwise Markov Random Field. Messages carrying nodal information are propagated and updated until convergence is achieved in the belief estimates of the hidden nodes. The belief distribution at each node now forms the new estimate for the depth at each sample. The propagation of information results in the observation of adjusted depth means and a reduction in the variance of each sample depth indicating more reliable estimates. These final depth values can be projected out to define surfaces in the scene. These surfaces are finally appropriately textured with portions of the image to form the end result of the algorithm.

■ 7.4 Summary

In this chapter, we have described the overall implementation of the algorithm by piecing together the various topics described throughout this thesis. The result is a voxel reconstruction algorithm that can recover the shape, color and illumination of urban scenes. The voxel representation of the scene can then be used to produce dense depth maps and surfaces. In the following chapter, the described algorithm is tested on a variety of synthetic and real data set to illustrate its effectiveness.

Chapter 8

Reconstruction Results

In Chapter 7, we described the implementation of the algorithm to automatically recover scene shape and color under canonical illumination from calibrated imagery. In this chapter, we present the results of using the described system on a variety of inputs from synthetic data to the target real data of urban environments from the City Scanning dataset. All CPU times are for a single threaded version of the algorithm running on a 250MHz Silicon Graphics Octane with 1GB of RAM.

■ 8.1 Synthetic Data

The algorithm was initially tested on a series of synthetic datasets for verification via the availability of ground truth. Scenes were created and rendered using the Inventor modeling language. The internal and external parameters of the virtual cameras are known and the synthetic images are of size 400×300 pixels.

■ 8.1.1 Textured Plane

The first test dataset for reconstruction consisted of 15 synthetic images of a textured planar object in front of a uniform black background. The top row of Figure 8.1 shows three example images from the dataset. The lighting in the scene was simulated using a single fixed point light source. Since the lighting is consistent across all views, the illumination does not need to be estimated and the pixel color can be used directly for matching purposes. The reconstructed scene is shown in Figures 8.1(bottom row) from the same viewpoints as the virtual cameras. The reconstruction is visibly faithful to the original input images. The reconstruction is also shown from a novel viewpoint is Figure 8.2. Using the multi-resolution approach, the reconstruction occurs at three different resolutions from coarsest to finest. The voxel opacities and corresponding colors are found to converge in six iterations of the algorithm at the highest resolution. This complete reconstruction takes approximately 3 CPU minutes to complete.

Image space validation

The accuracy of the reconstruction in image space is highlighted in Figure 8.4. The difference between the sample colors projected from the image and the reprojected

Example Images

Reconstruction results

Figure 8.1. Input images (top row) and views of reconstructions from the same camera viewpoints (bottom row).

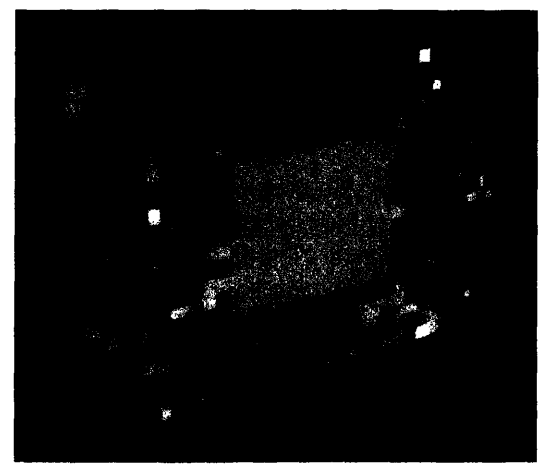
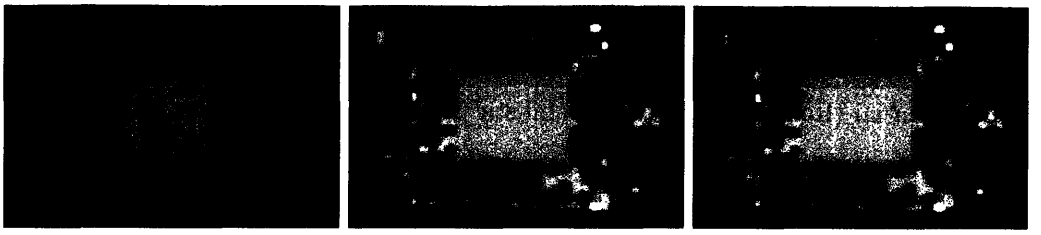


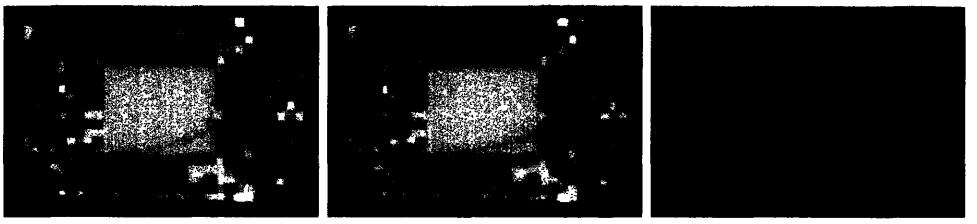
Figure 8.2. The reconstructed textured plane from a novel viewpoint.

reconstruction is shown in Figure 8.4. The root-mean-squared error in color values over the entire is 7%. This relatively large error can be attributed to the high frequency colored texture on the planar surface where the voxels that lie on the boundary between the two highly contrasting colors exhibit the effect of interpolation. The error can be



Exponential (Roxels) Probabilistic RGB Probabilistic CIE-*Luv*

Figure 8.3. Comparison of textured plane reconstructions for various agreement computations.



Sampled Input Reconstruction Error

Figure 8.4. Accuracy of the textured plane model: The difference of the sampled image and the reprojected reconstruction provides an error metric in image space. For this image. the RMS error over all color channels was 7%.

reduced through further voxel divisions.

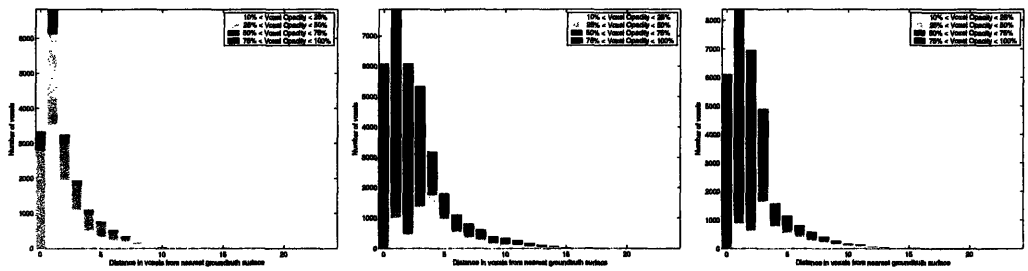
World space validation

With the availability of groundtruth, the reconstructed model can also be validated in world space. The cross-section of the reconstructed plane is shown in Figure 8.5. The red line indicates the true position of the plane and the voxels can be seen to closely fit the true model. In order to quantitatively measure the accuracy of the reconstruction, the known geometry is first used to locate voxels on the surface of the model, thus providing a best case volumetric model of the scene. The positions of voxels in the reconstructed model can then be compared to this ideal model in order to evaluate its accuracy. The distance from each reconstructed voxel to closest true surface voxel is measured and histogrammed. The histograms in Figure 8.6 highlight the advantages of using probabilistic methods to compute color agreement over the exponential method used in the Roxels approach. The histograms for the probabilistic methods show higher voxel opacity for voxels close to the surface. In comparison, using the exponential agreement method results in fewer voxels close to the surface and lower voxel opacities overall. Since no illumination variation exists between input images, we do not expect to see much difference between the reconstructions when agreements are computed in RGB and CIE-*Luv* color spaces.

Further comparisons between the agreement computation methods are shown in

Exponential (Roxels) Probabilistic RGB Probabilistic CIE-*Luv*

Figure 8.5. Cross-sections of the reconstructed textured plane for various agreement computations. The voxels are false colored to enhance detail. The red line is derived from the original model indicating the true surface position.



Exponential (Roxels) Probabilistic RGB Probabilistic CIE-*Luv*

Figure 8.6. Histograms of reconstructed voxel distance from the groundtruth surface for the textured plane model using the exponential color matching (left) from the Roxels approach, probabilistic RGB matching (center), and probabilistic CIE-*Luv* matching (right).

Table 8.1. Voxels that are located within four voxel lengths of the groundtruth surface are first labeled as being close to the true model. Numerical comparisons between the various agreement methods can then be made by examining the percentage of all voxels that are labeled as close to the surface. For those voxels that are on or near the surface, we also expect a convergence to higher opacity values. This is measured through calculating a distance weighted mean opacity for all voxels close to the surface. Again, the probabilistic methods can be seen to provide vast improvements over the exponential approach.

Agreement Computation	Total Number of Voxels	Voxels close to groundtruth (%)	Distance weighted mean opacity for voxels close to groundtruth (%)
Exponential (Roxels)	21928	73.2	27.1
Probabilistic RGB	35281	81.4	76.3
Probabilistic CIE-*Luv*	32223	83.3	77.0

Table 8.1. Comparison of textured plane reconstructions using different color agreement methods. Voxels are defined as being close to the surface if they are located within four voxels of the groundtruth model.

■ 8.1.2 Textured Head

A textured head model is used as the second synthetic dataset. The geometry of the object is more complex than the simple textured plane described in Section 8.1.1. The model also contains multiple homogeneous regions of color which are typically more

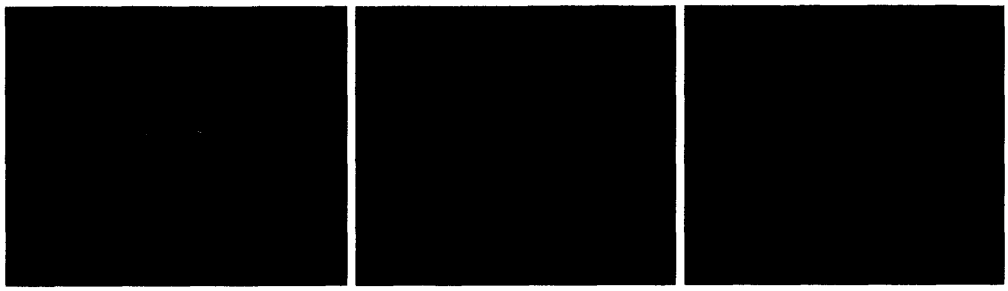


Figure 8.7. Textured Head Sequence: Sample input images from the 20 images in the sequence.

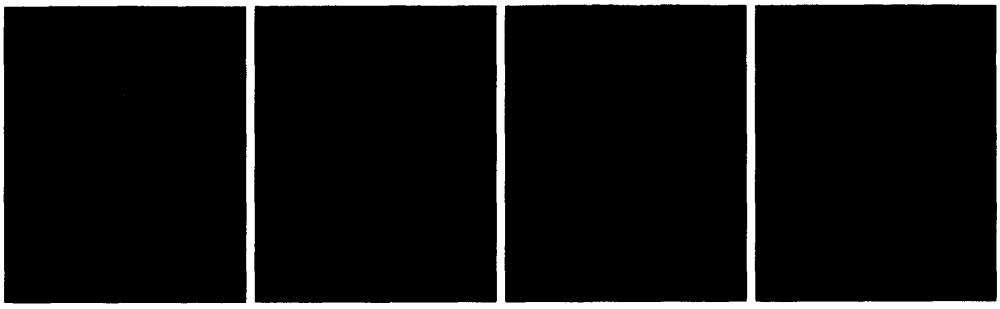


Figure 8.8. Views of the reconstructed model head from novel viewpoints.

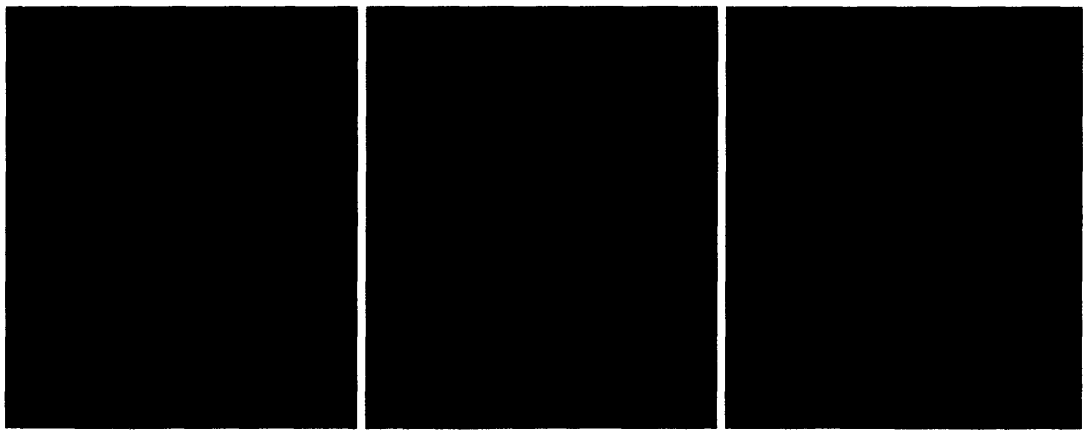
difficult to reconstruct due to the ambiguities in matching that they introduce. The dataset is made up of 20 images, taken in a turntable style sequence around the model head. Three examples of the input images are shown in Figure 8.7. Figures 8.8 show the reconstructed model from novel viewpoints and again can be seen to be faithful to the original input. The reconstruction is also conducted at three different resolutions, starting at the coarsest and then making refinements to the finest. The complete reconstruction of the head takes approximately 5 CPU minutes.

Image space validation

The accuracy of the reconstruction can again be measured in image space in the same way as the previous textured plane example. The difference between the sample colors projected from the image and the reprojected reconstruction is shown in Figure 8.10. The root-mean-squared error in color values over the entire is less than 3%.

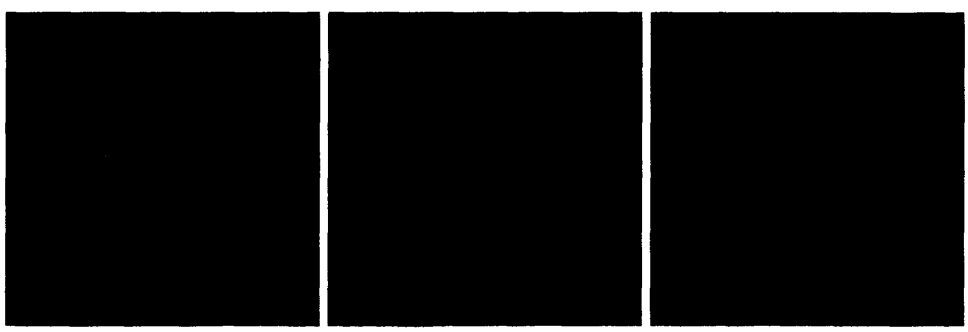
World space validation

A simple validation of the reconstructed model in world space can be made visually by comparison to the groundtruth model. Cross-sections of the reconstructed volumetric model are shown in Figure 8.11. The reconstructed model is hollow since once voxels at the surface converge to high opacity values, the responsibility (and subsequently the



Exponential (Roxels) Probabilistic RGB Probabilistic CIE-*Luv*

Figure 8.9. Comparison of textured head reconstructions for various agreement computations.

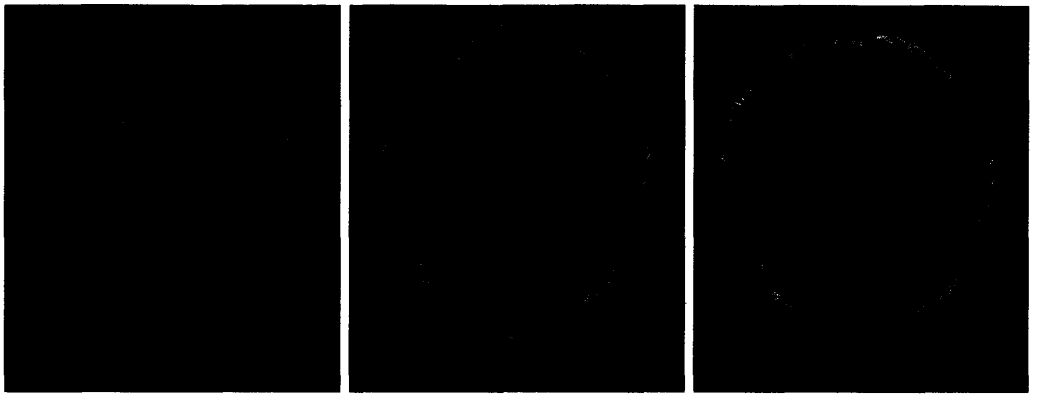


Sampled Image Reconstruction Error

Figure 8.10. Accuracy of the textured head model: The squared difference of the sampled image and the reprojected reconstruction is shown as the error image. The RMS error in the estimated colors is less than 3% over the entire image.

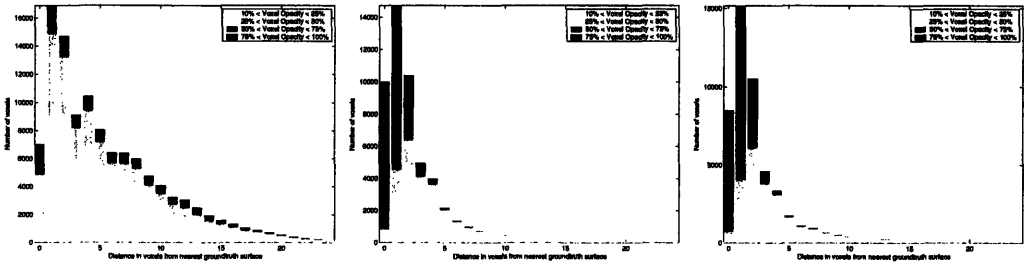
opacity) of voxels inside the volume are suppressed. Due to a lack of texture over the majority of the model, the recovered voxels are unable to accurately describe surfaces of high curvature and are instead roughly approximated. Histograms of distances from the recovered voxels to the closet voxel in best case volumetric model using groundtruth are shown in Figure 8.12. The advantages of using the probabilistic methods can again be seen to provide more voxels, with higher opacities, closer to the true surface locations over the exponential agreement method. These gains can also be seen in Table 8.2 with higher mean opacity values for voxels close to the surface. The total number of voxels reconstructed using the probabilistic methods is also lower since voxels inside the model are automatically suppressed.

Synthetic data provides a good foundation on which to test the underlying principles



Exponential (Roxels) Probabilistic RGB Probabilistic CIE-*Luv*

Figure 8.11. Cross-sections of the reconstructed textured head model for various methods of agreement computation. The voxels are false colored to enhance detail. The red lines are derived from the original model indicated the true surface position. Notice that using the probabilistic methods only voxels on, or near the surface are reconstructed. Due to a lack of texture over the majority of the model, regions of high curvature are roughly approximated.



Exponential (Roxels) Probabilistic RGB Probabilistic CIE-*Luv*

Figure 8.12. Histograms of reconstructed voxel distance from the groundtruth surface for the textured head model using the exponential color matching (left) from the Roxels approach, probabilistic RGB matching (center), and probabilistic CIE-*Luv* matching (right).

of the algorithm. This can be attributed to perfect camera information and fixed simulated lighting conditions. To truly test the usefulness of the algorithm, real images must be used as input.

■ 8.2 Real Data

Reconstructions of scenes from real images present many challenges over synthetic data. In addition to the more complex information available in the images, the positions and orientations of the cameras are unlikely to be known precisely and therefore limit the resolution to which the scene description can be recovered. Another challenge is presented by the possibility of changing illumination that may exist between images in

Agreement Computation	Total Number of Voxels	Voxels close to groundtruth (%)	Distance weighted mean opacity for voxels close to groundtruth (%)
Exponential (Roxels)	127245	40.2	29.3
Probabilistic RGB	65890	63.4	57.7
Probabilistic CIE-*Luv*	62996	64.2	59.9

Table 8.2. Comparison of textured head reconstructions using different color agreement functions. Voxels are defined as being close to the surface if they are located within four voxels of the groundtruth model.

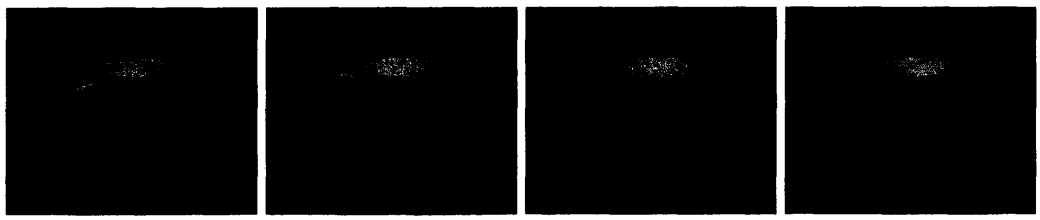


Figure 8.13. Example images from the coffee mug sequence. Images are courtesy of Peter Eisert, Laboratorium für Nachrichtentechnik, Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

the sequence.

We attempt to introduce these difficulties one at a time by first considering a dataset that contains typical errors in camera positions, but was acquired in a controlled environment away from noticeable illumination variation. We then move to data from the City Scanning Project that contain both errors in the camera pose and illumination effects.

■ 8.2.1 Coffee Mug

The first real dataset to be tested is of a coffee mug imaged as it rotates on a turntable. The data consists of 14, 352 × 288 images and are provided courtesy of Peter Eisert at Laboratorium für Nachrichtentechnik, Friedrich-Alexander-Universität Erlangen-Nürnberg in Germany. Four sample images from the dataset are shown in Figure 8.13. The reconstruction was performed at three increasing resolutions from initial grid dimensions of $50 \times 50 \times 20$ to a final resolution of $200 \times 200 \times 80$ after sub-divisions. Images of the final model from various viewpoints are shown in Figure 8.14. Two cross-sections of the mug are shown in Figure 8.16 highlighting that the cylindrical nature of the object is recovered. Only voxels on the surface converge to high opacity levels while those inside the object converge to zero. The complete reconstruction took approximately 24 CPU minutes.

Image space validation

The accuracy of the model is demonstrated in Figure 8.15 by the difference between the sample colors projected from the image and the reprojected reconstruction. The root-mean-squared error in color values over the entire is less than 6%.

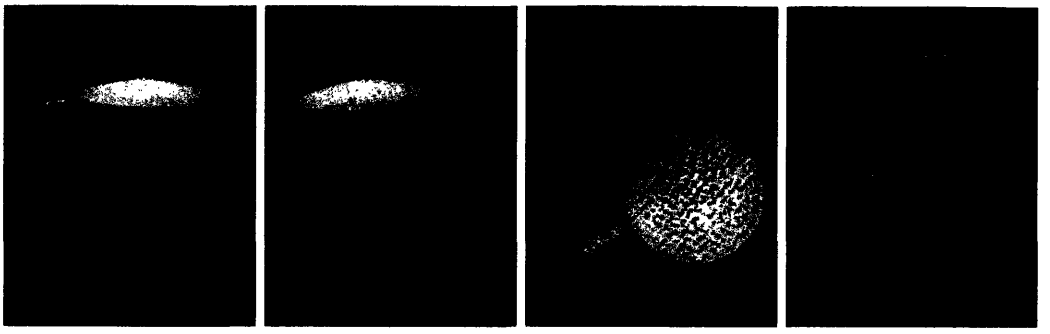
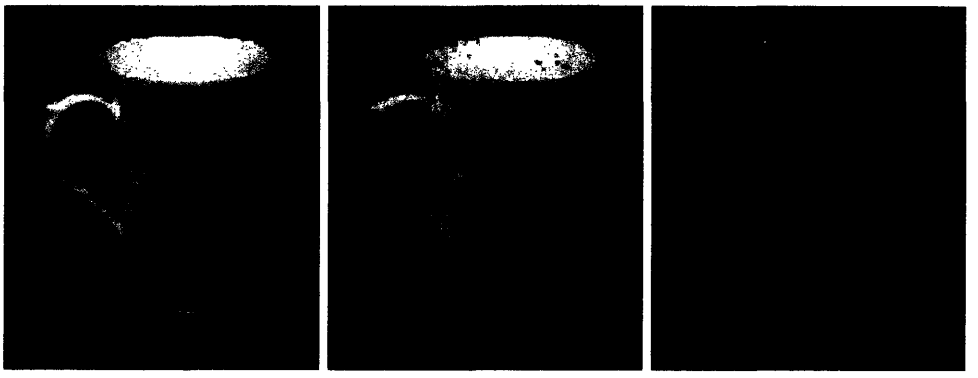


Figure 8.14. Reconstruction results from novel viewpoints for the coffee mug sequence.



Sampled Image Reconstruction Error

Figure 8.15. Accuracy of the coffee mug model: The squared difference of the sampled image and the reprojected reconstruction results in the error image. The RMS error in the estimated colors is less than 6% over the entire image.

World space validation

Since no groundtruth was available for the coffee mug model, the results could only be validated qualitatively. Horizontal and vertical cross-sections of the reconstructed model in Figure 8.16 show that voxels inside the mug have converged to low opacities leaving only voxels on the surface. The voxels are false colored in order to enhance contrast. The cylindrical shape of the mug and the handle are also clearly visible.

Having successfully performed a reconstruction using real data acquired in a controlled environment, we now present results for the target data set from the City Scanning project at MIT.

■ 8.2.2 Media Lab

The next test input sequence is from the City Scanning dataset at MIT. The images are taken outdoors in an uncontrolled environment subject to camera calibration errors

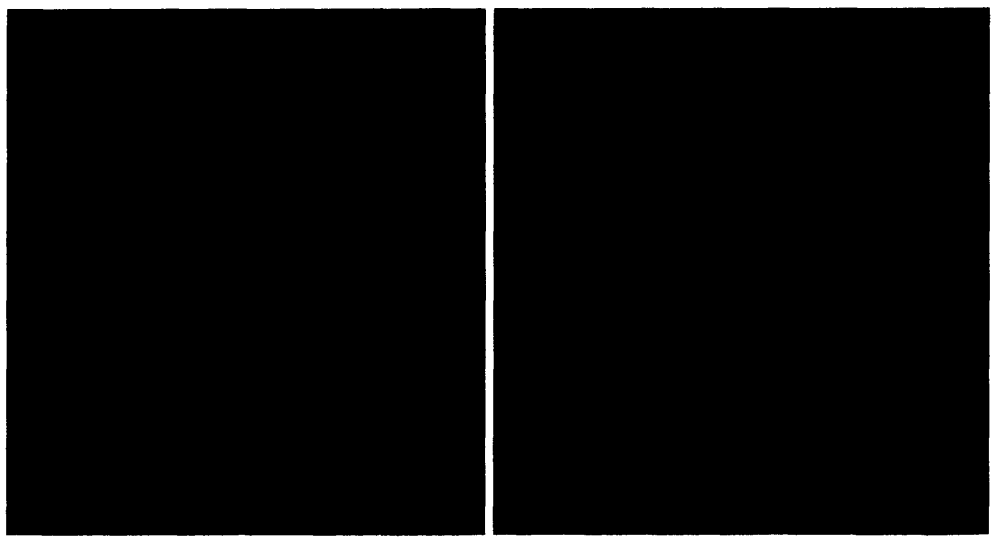


Figure 8.16. A cross-sections of the reconstructed coffee mug. The voxels are false colored to enhance detail. The voxels are false colored in order to enhance contrast. Note that the reconstruction does have the desired circular cross-section with voxels inside the mug converging to zero.

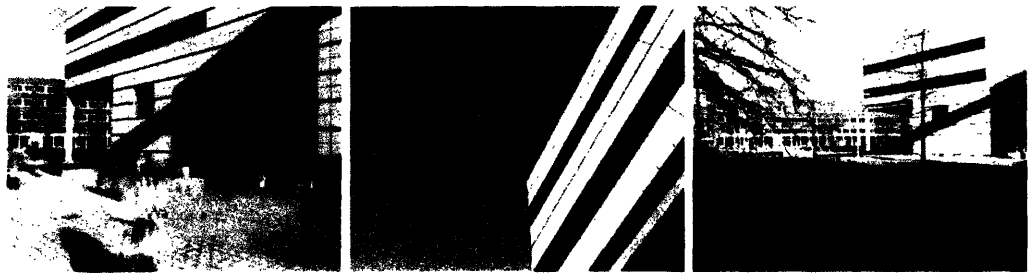


Figure 8.17. Example input images from the Media Lab sequence, part of the City Scanning dataset

and changes in illumination. The input consists of 13 nodes, each of which is made up of 20 individual images as described in Section 1.1. The nodes are acquired in the vicinity of the Media Lab building on the MIT campus which forms the target for our reconstruction algorithm. Each image has a resolution of 324×256 pixels and is in HDR (High Dynamic Range) format. Examples of the input images are shown in Figure 8.17. Examples of the estimated illumination for the images are shown in Figure 8.18. Results for the reconstruction are shown in Figure 8.19. Although the shape and color of the main façade of the Media Lab can be seen to be successfully recovered, the reconstruction contains a number of noisy voxels which can be attributed to the homogeneity (lack of texture) of the target surface.

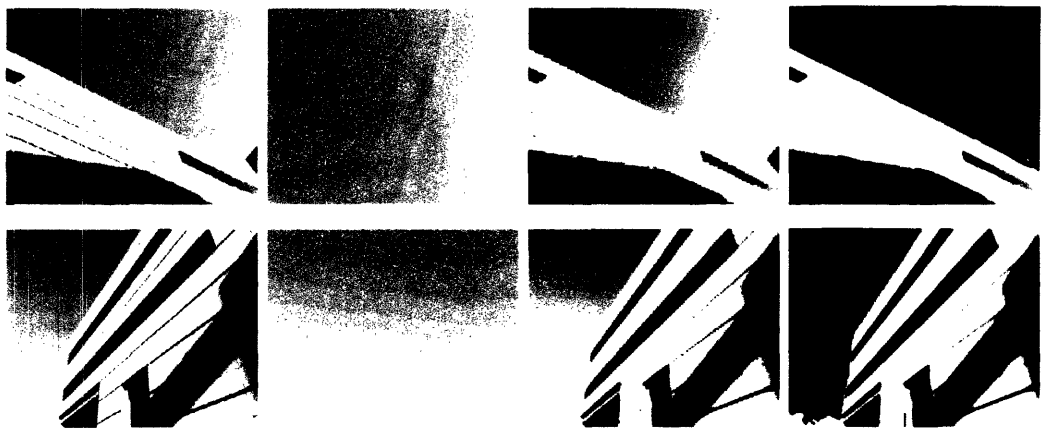


Figure 8.18. From left to right: Examples of original input from the Media Lab sequence, illumination estimates for the images, image colors adjusted by the illumination and the segmented foreground from the image.

Image space validation

A image space comparison can again be made for reconstructions produced using each of the color agreement measures. Since the Media Lab dataset also contains changes in illumination conditions between images, the effects of illumination adjustment on the reconstruction can also be validated. Figure 8.20 shows identical views of the reconstructed Media Lab for the various color agreement strategies, both with and without adjustment for illumination. The results produced using the adjustment appear visibly sharper with improved contrast. The difference in the reconstructions using the different color matching methods is less obvious. The boundaries between high contrast regions however, such as around the tinted windows, appear sharper using the probabilistic method in CIE-*Luv* color space.

The accuracy is again measured numerical by comparing the sampled image where colors are adjusted for illumination, with the reprojected reconstruction from the same view. The difference in the images is shown in Figure 8.21. The RMS error in the estimated colors is less than 10% in all color channels over the entire image.

World space validation

With no groundtruth model available for the Media Lab dataset, only qualitative comparisons could be made between reconstructions acquired using the different color agreement methods, both with and without illumination adjustment. Cross-sections of the models are shown in Figure 8.22 with the voxels false colored to enhance contrast. With this visualization, the flat shape of the main façade model is still visible despite noise in the model. Although visual comparisons between the methods show noticeable gains in using probabilistic matching methods over a simple exponential matching strategy,

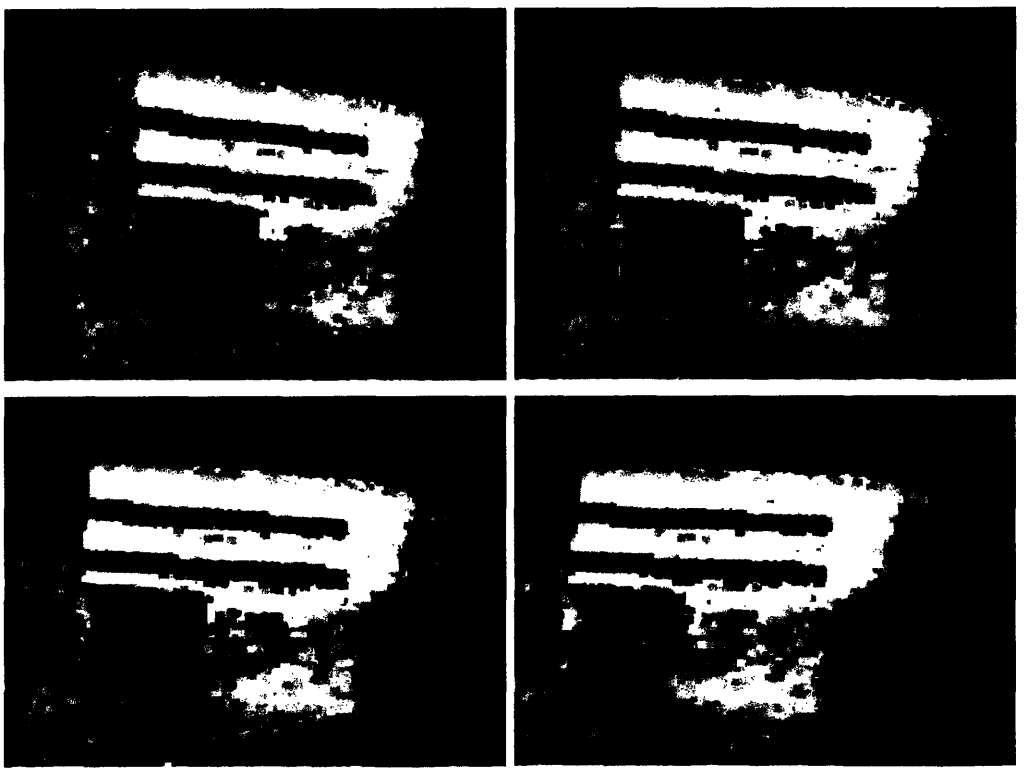


Figure 8.19. Reconstruction from the Media Lab sequence. The result, although noisy, does represent the main façade of the building.

the effects of using the perceptually uniform CIE-*Luv* color space over RGB appear less obvious.

■ 8.2.3 Green Building

The final test input is also from the City Scanning dataset at MIT. The input consists of just 7 nodes, each of which is made up of between 20 and 40 individual images as described in Section 1.1. The nodes are acquired in the vicinity of the Green building on the MIT campus which forms the target for this reconstruction. Each image has a resolution of 324×256 pixels again in HDR (High Dynamic Range) format. Examples of the input images are shown in Figure 8.23. Examples of the estimated illumination for the images are shown in Figure 8.24. The evolving reconstruction is shown in Figure 8.25 highlighting how the multi-resolution optimization is used to recover large scale features first. Final results for the reconstruction are shown in Figure 8.26. The result is a crude model, but does highlight the shape and color of the actual building.

Exponential (Roxels) Probabilistic RGB Probabilistic CIE-*Luv*

Without Illumination Adjustment

Exponential (Roxels) Probabilistic RGB Probabilistic CIE-*Luv*

With Illumination Adjustment

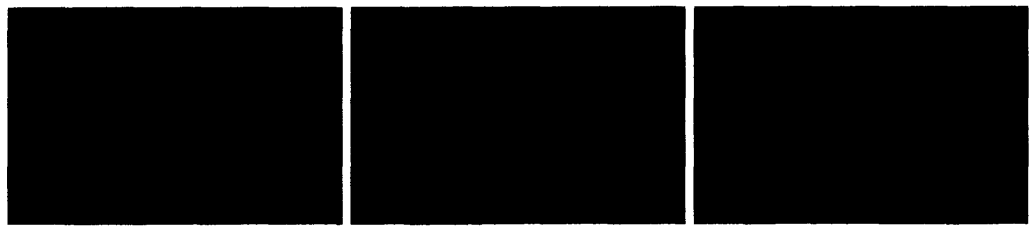
Figure 8.20. Identical views of the reconstructed Media Lab building at MIT for various color agreement functions and illumination adjustments.

Original Sampled Image Reconstruction Error

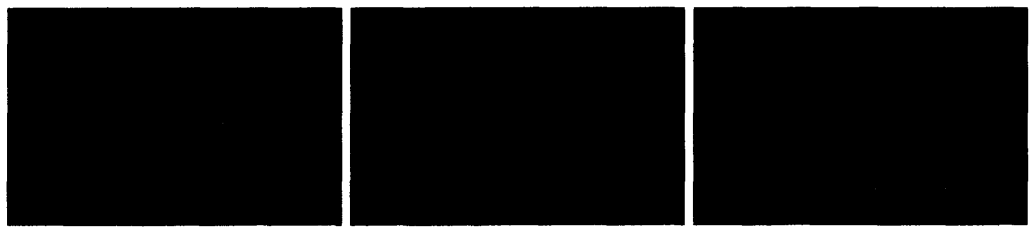
Figure 8.21. Accuracy of the Media Lab model: The difference of the sampled image and the reprojected reconstruction is shown as the error image on the right. The RMS error in the estimated colors is less than 10% in all color channels over the entire image.

Image space validation

The algorithm was tested on the Green building data set with each of the color agreement functions both with and without large scale illumination adjustment. The reconstruction without illumination adjustment appear grey, unlike the true appearance of the façade. The results with illumination adjustment better highlight the colors in the scene with reconstructions using the CIE-*Luv* space agreement measure providing the



Exponential (Roxels) Probabilistic RGB Probabilistic CIE-*Luv*
Without Illumination Adjustment



Exponential (Roxels) Probabilistic RGB Probabilistic CIE-*Luv*
With Illumination Adjustment

Figure 8.22. Cross-sections of the reconstructed Media Lab at MIT using the different color agreement methods, both with and without illumination adjustment. The voxels are false colored to enhance detail. the flat shape of the main façade model is still visible towards the bottom of the image.

Figure 8.23. Three sample images from the Green building sequence from the City Scanning dataset.

image with the least noise. The resulting reconstructions are shown in Figure 8.27. An image space comparison of the original input image, sampled image with colors adjusted for illumination effect, and the reprojected reconstruction is shown in Figure 8.28. The RMS error in each color channel over the entire image is less than 5%.

The effects of illumination adjustment can be more clearly seen in Figure 8.29. The model is imaged in both cases with the estimated background placed behind the model. The color in the model without illumination estimation appears grey and washed-out. This effect can be attributed to a larger error in the mean color estimates that are obtained.

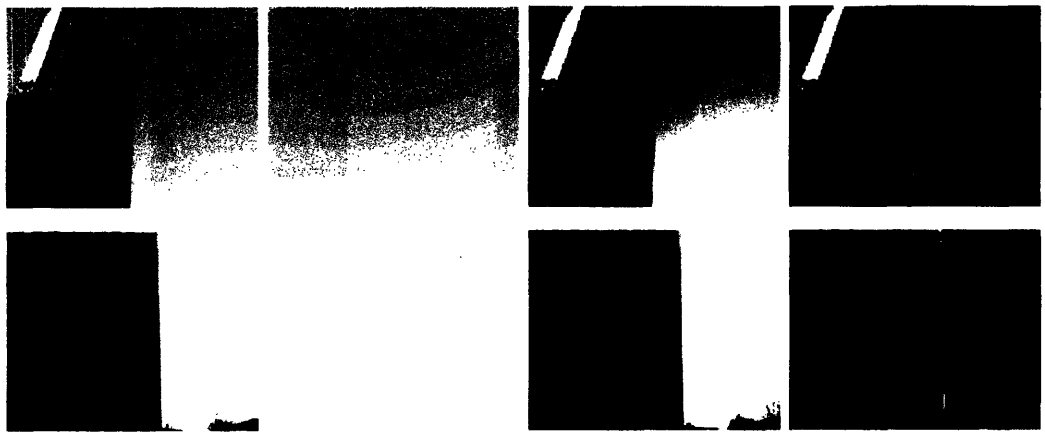


Figure 8.24. From left to right: Examples of original input from the Green building sequence, illumination estimates for the images, image colors adjusted by the illumination and the segmented foreground from the image.

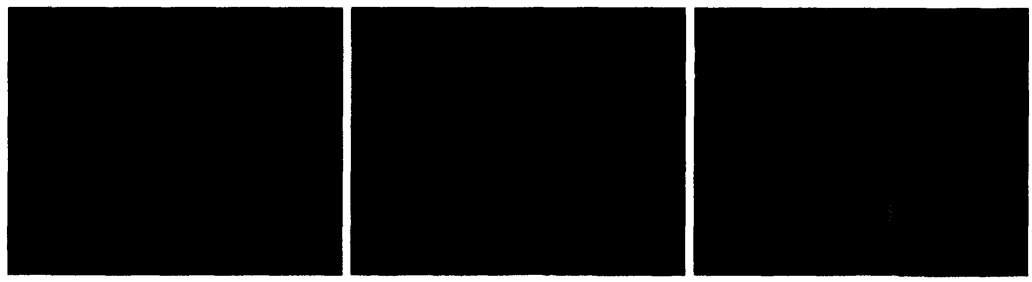
World space validation

A piecewise planar model approximately resembling the shape and position of the Green building was created by hand and used as groundtruth in order to validate the reconstruction results in world space. Cross-sections of the model are shown in Figure 8.30 for different color agreement strategies and adjustment for illumination effects with the voxels false colored to enhance contrast and the approximate groundtruth overlayed as a red line. The effects are again quantitatively measured using distance histograms of voxel position and opacity from the closest true surface voxel position. These histograms are shown in Figures 8.31 and 8.32. The color agreement methods are first compared without adjustment for illumination as shown in Figure 8.31. The probabilistic agreement measure in CIE-*Luv* space can again be seen to produce higher opacity voxels closer to the true object surface with fewer voxels for from the surface. Table 8.3 gives numerical comparisons between the methods showing the using our color matching method produces on average 13% more opaque voxels close to the surface.

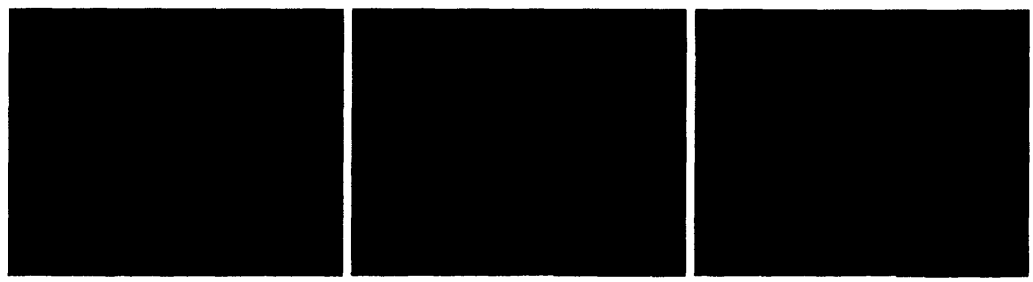
Agreement Computation	Total Number of Voxels	Voxels close to groundtruth (%)	Distance weighted mean opacity for voxels close to groundtruth (%)
Exponential (Roxels)	223721	46.1	25.3
Probabilistic RGB	173755	42.3	26.2
Probabilistic CIE-*Luv*	114488	47.8	39.5

Table 8.3. Comparison of Green building reconstructions using different color agreement functions without illumination adjustment. Voxels are defined as being close to the surface if they lie within four voxels of the groundtruth model.

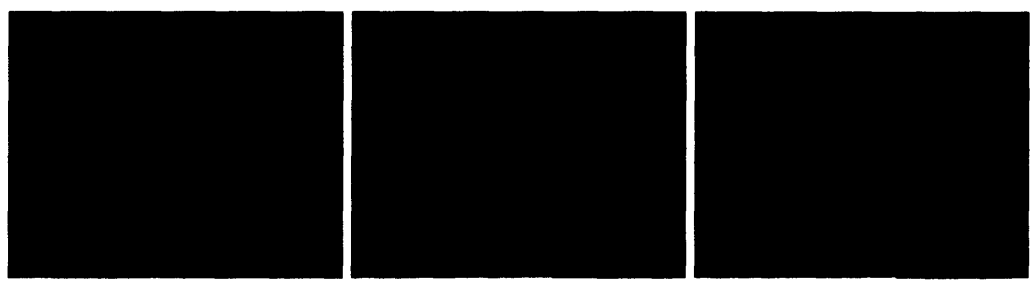
The same validations test are repeated, this time also performing the adjustment



Low resolution



Intermediate resolution



Highest resolution

Figure 8.25. Reconstruction results at various stages in the process with color adjustment for illumination effects: The top row shows the reconstruction evolving (left to right) at the lowest resolution. The middle row shows the reconstruction at an intermediate resolution and the bottom row shows the same stages at the highest resolution.

for large scale illumination. The resulting histograms are shown in Figure 8.32. The numerical comparisons of the methods in Table 8.4 show only marginal improvement in accuracy of the resulting models acquired without illumination adjustment. For voxels close to the surface, the mean opacity increases by over 4% using probabilistic CIE-*Luv* color matching through adjustment for large scale illumination changes in the images.

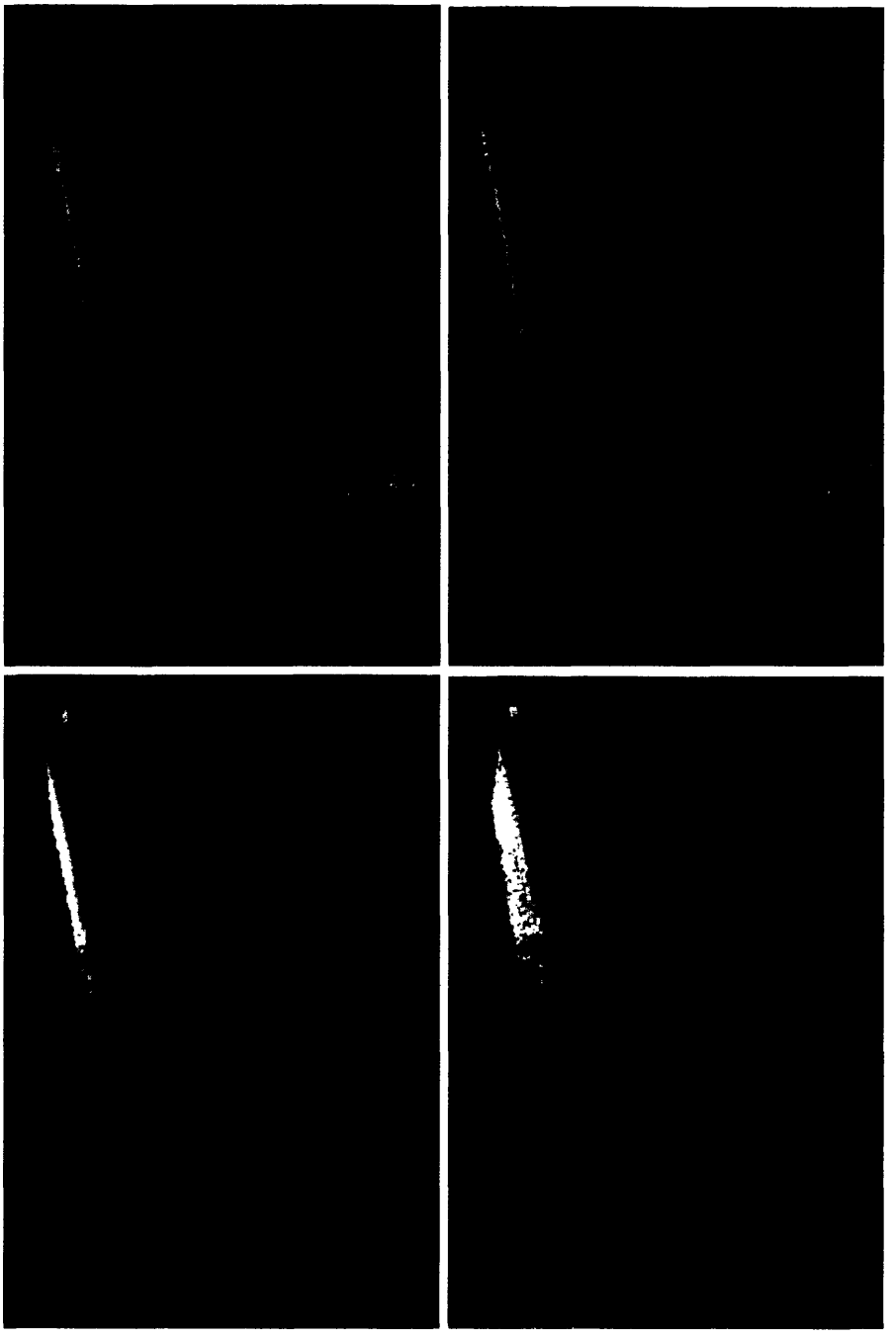
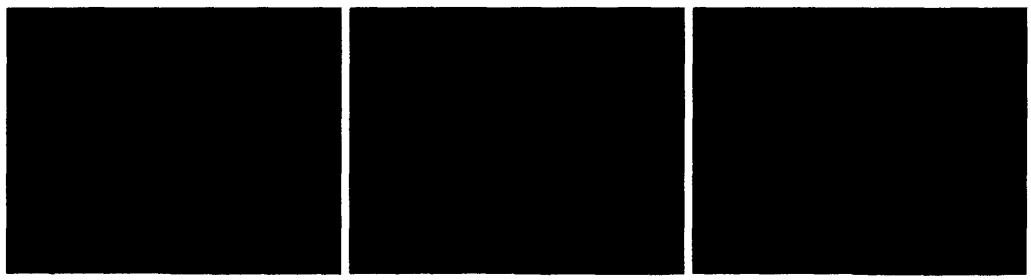


Figure 8.26. Several views of the green building reconstruction from novel viewpoints.



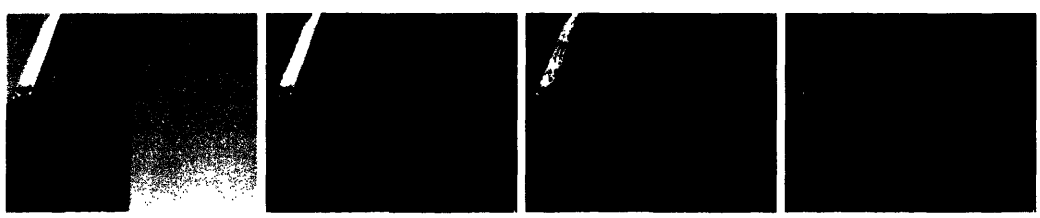
Exponential (Roxels) Probabilistic RGB Probabilistic CIE-*Luv*

Without Illumination Adjustment

Exponential (Roxels) Probabilistic RGB Probabilistic CIE-*Luv*

With Illumination Adjustment

Figure 8.27. Cross-section of the reconstructed Green building at MIT for various color agreement functions and illumination adjustments. The voxels are false colored to enhance contrast. The red line represents the true surface boundary and is computed from a manually derived groundtruth model of the building.



Original Sampled Image Reconstruction Error

Figure 8.28. Accuracy of the Green building model: The difference of the sampled image and the reprojected reconstruction is shown as the error image on the right. The RMS error in the estimated colors is less than 5% in all color channels over the entire image.

■ 8.3 Depth and Surface Estimation

Once volumetric models for the scenes are available, we can test the final portion of the algorithm which estimates depth. The volumetric model is used as input and estimates of the depth mean and variance are computed. These depth estimates were then placed

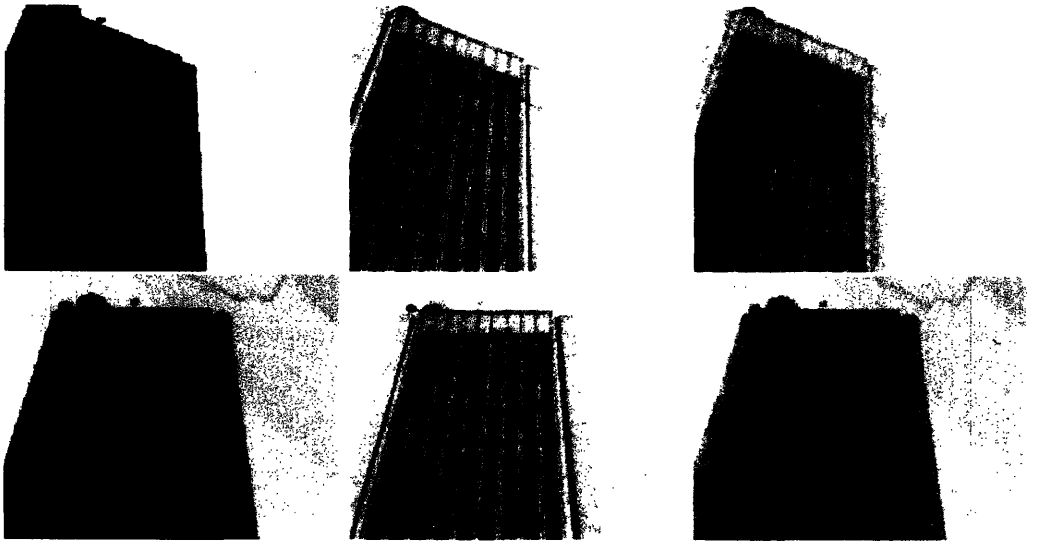


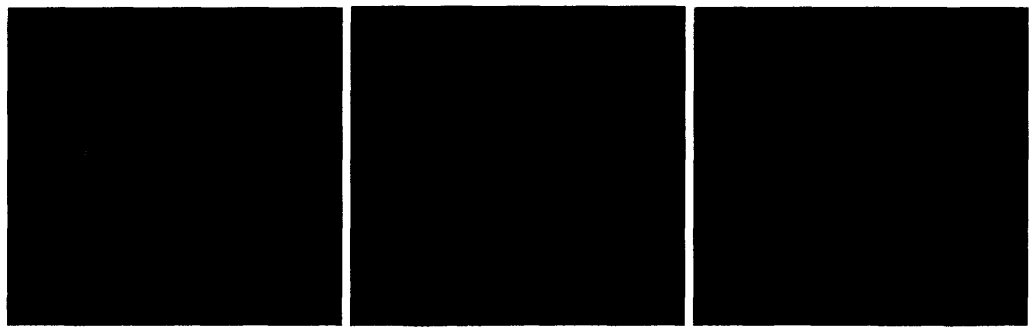
Figure 8.29. Comparison of reconstruction results with (right) and without (left) color adjustment according to illumination in the scene. The recovered color can be seen to be closer to the original (center) when illumination effects are taken into account.

Agreement Computation	Total Number of Voxels	Voxels close to groundtruth (%)	Distance weighted mean opacity for voxels close to groundtruth (%)
Exponential (Roxels)	212537	45.2	27.2
Probabilistic RGB	118236	40.6	27.4
Probabilistic CIE-*Luv*	116920	47.3	43.9

Table 8.4. Comparison of Green building reconstructions using different color agreement functions without illumination adjustment. Voxels are defined as being close to the surface if they lie within four voxels of the groundtruth model.

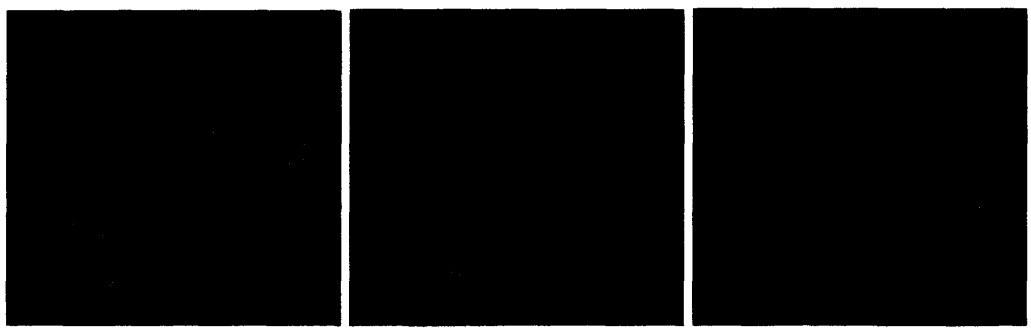
in the belief propagation system described in Chapter 6. Figure 8.33 shows an image from the cup sequence with the corresponding images of the estimated depth and surface before and after the BBP process. The estimates can be seen to be smoother after belief propagation (Figure 8.33 bottom row). A textured surface representation of the cup is shown in Figure 8.34.

Figure 8.35 demonstrates the belief propagation on an image from the Green Building dataset. The average variance or ambiguity in depth over the entire image falls from $3.16m$ before BBP, to $0.09m$ afterwards. Notice that the mean depth estimates are also smoothed after the BBP due to the neighboring measurements contributing to one another and providing more reliable overall estimates. The surface obtained from this image is shown in Figure 8.36. The surface is textured and shown overlayed on a manually computed ground truth estimate to the shape of the building. The estimated



Exponential (Roxels) Probabilistic RGB Probabilistic CIE-*Luv*

Without Illumination Adjustment



Exponential (Roxels) Probabilistic RGB Probabilistic CIE-*Luv*

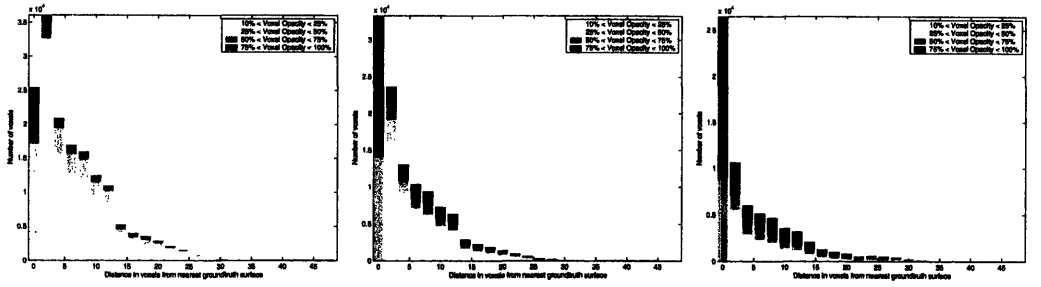
With Illumination Adjustment

Figure 8.30. Cross-section of the reconstructed Green building at MIT for various color agreement functions and illumination adjustments. The voxels are false colored to enhance contrast. The red line represents the true surface boundary and is computed from a manually derived groundtruth model of the building.

surface can be seen to provide a good representation of the true building façade.

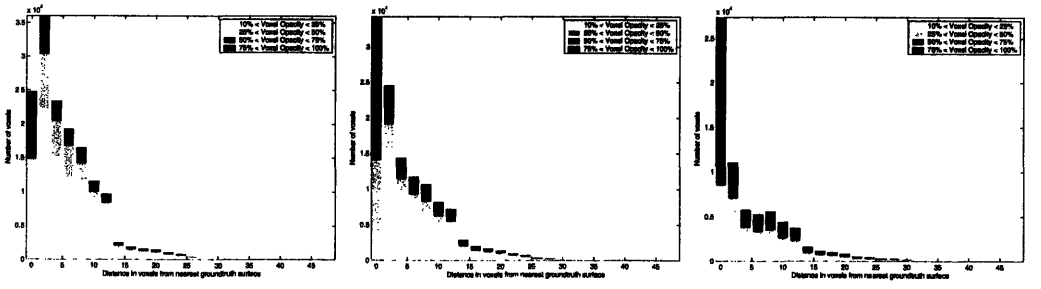
■ 8.4 Summary

In this chapter, we have presented the results of running the volume reconstruction algorithm on a number of inputs with varying levels of difficulty. Results are given for simple synthetic images with perfectly calibrated cameras and no illumination variation. The results can be seen to faithfully recover the shape and color of the target objects. Results are also present for real data sets including two examples from the City Scanning dataset. Although the challenges of working with real data can be seen in the results, the algorithm is still successful in recovering a good approximation to the shape of the objects. The following chapter concludes with a summary of the work presented in this thesis and examines future directions for the research.



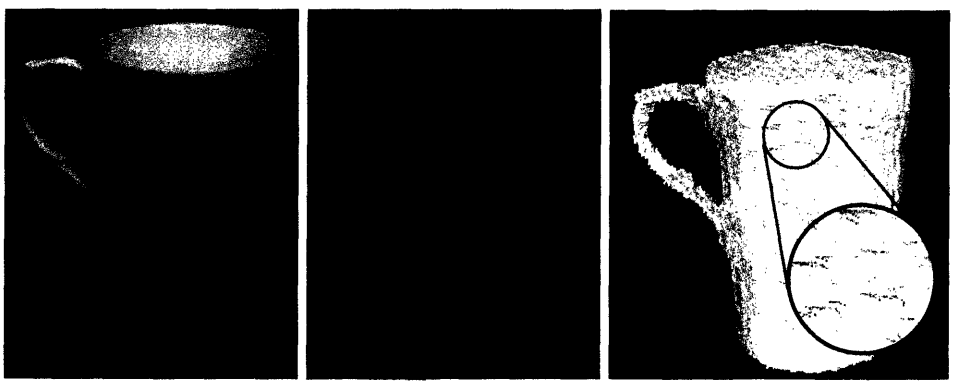
Exponential (Roxels) Probabilistic RGB Probabilistic CIE-*Luv*

Figure 8.31. Histograms of reconstructed voxel distance from the groundtruth surface for the green building dataset using the exponential color matching (left) from the Roxels approach, probabilistic RGB matching (center), and probabilistic CIE-*Luv* matching (right) without illumination adjustment.

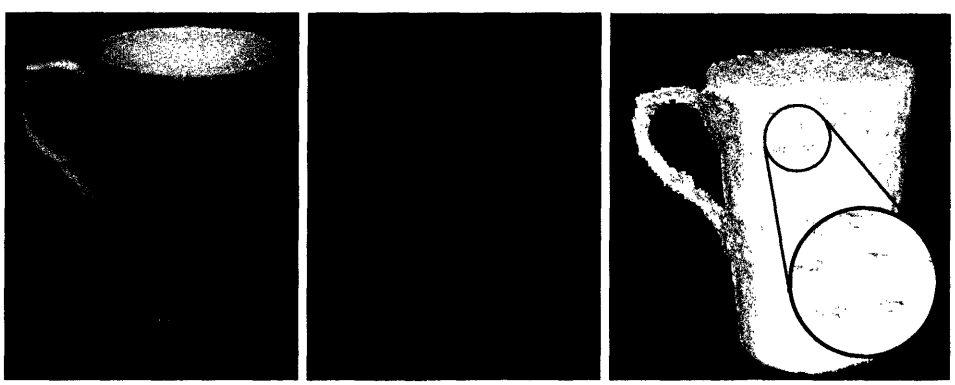


Exponential (Roxels) Probabilistic RGB Probabilistic CIE-*Luv*

Figure 8.32. Histograms of reconstructed voxel distance from the groundtruth surface for the green building dataset using the exponential color matching (left) from the Roxels approach, probabilistic RGB matching (center), and probabilistic CIE-*Luv* matching (right) with illumination adjustment.



Before Bayesian Belief Propagation



After Bayesian Belief Propagation

Figure 8.33. Depth maps before (top) and after BBP (bottom) for an image from the cup sequence.

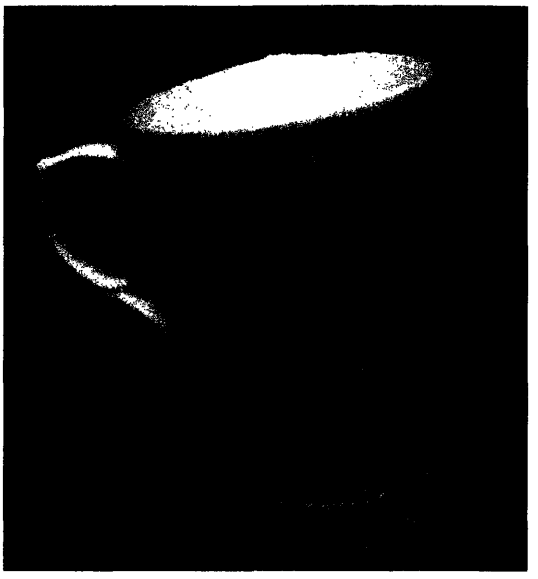
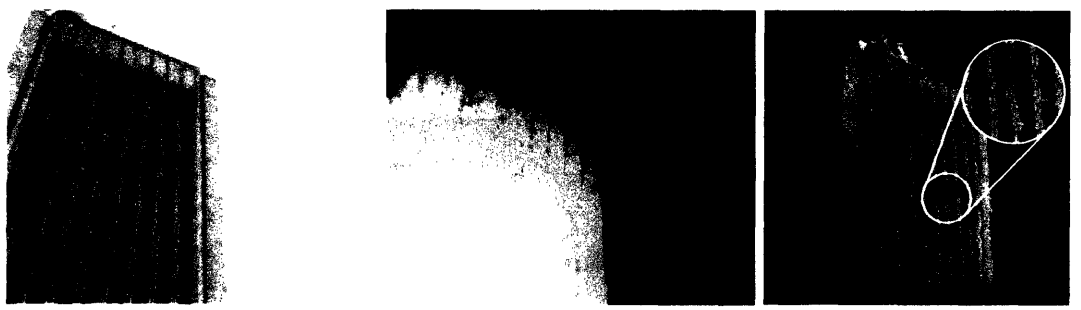
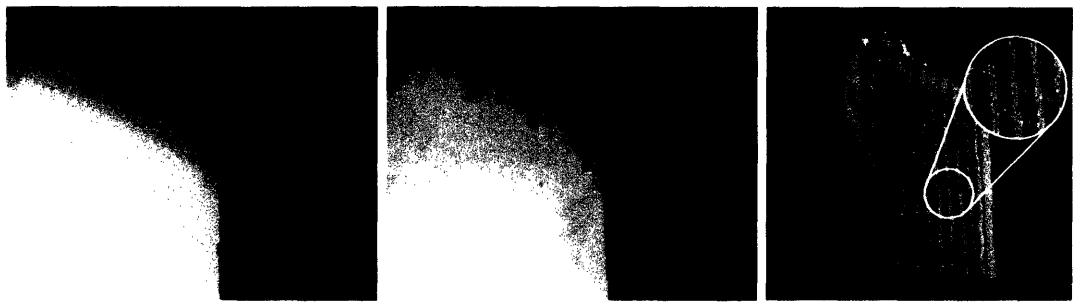


Figure 8.34. Estimated surface of the cup from a novel viewpoint. The surface is textured to improve visual fidelity.



Before Bayesian Belief Propagation



After Bayesian Belief Propagation

Figure 8.35. Effects of Bayesian Belief Propagation (BBP) on an image from the Green building dataset. The depth map and surface after BBP (bottom row) presents a smoother overall appearance than before BBP (top row). The average variance in the depth falls from $3.16m$ before BBP to $0.09m$ after.

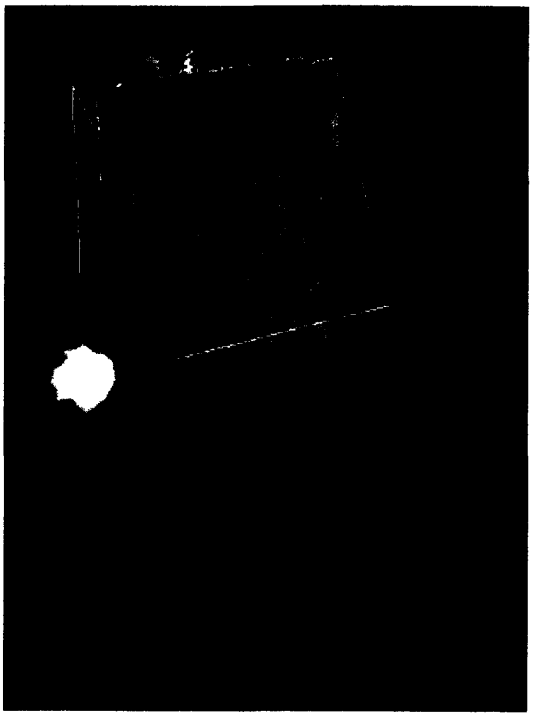


Figure 8.36. Surface estimation: An estimated portion of surface from the Green building overlayed on manually computed ground truth of the entire building.

Chapter 9

Conclusion and Future Work

9.1 Conclusions

The general problem of 3-D scene recovery from images is difficult. Much progress has been made performing the reconstruction on images acquired under controlled conditions. Indeed, the voxel coloring approach described in [36] leads to the understanding that it is a solved problem. This and other algorithms fail however when applied to data acquired away from the static conditions of an indoor rig. The changes in observed object colors due to lighting changes can cause the simple color based consistency checks used by these algorithms to fail.

In this thesis, we have presented an algorithm to perform volumetric reconstruction of urban environments from multiple calibrated images. The algorithm is based on the Responsibility weighted voxels (Roxels) approach introduced by DeBonet and Viola [18], but extends on this work by improving the agreement matching function and adds the ability to deal with varying illumination. Illumination variations are automatically estimated directly from high dynamic range images and then used to normalize the observed surface colors across images. A probabilistic approach to matching color distributions is also presented and used to improve the robustness of the reconstruction process. We have shown that performing the matching in the perceptually uniform CIE-*Luv* color space, over the more commonly used RGB space, leads to improved matching accuracy and reduced sensitivity to illumination variation. The algorithm also redefines the reconstruction process by centering it around the source of the information, namely the images rather than the volume itself. The reconstruction algorithm presented exhibits the following characteristics:

- Fully Automatic
- Iterative
- Does not place constraints on camera positions
- Performs multi-resolution reconstruction
- Easily parallelizable
- Explicitly deals with illumination variation across images

The major contributions of the work involve novel techniques for:

- Recovering 3D scene (shape and color) under canonical illumination
- Probabilistic Color Matching
- Obtaining surfaces from probabilistic voxels
- Where possible, detection and factoring out of illumination

The results presented in Chapter 8 verify the application of the algorithm but also indicate that this work is only a step in the direction of a complete general algorithm. The work still suffers from limitations that should be addressed as future research work.

■ 9.2 Limitations and Future Work

The results acquired from City Scanning data set highlight some limitations of the algorithm in dealing with real data. These problems and suggested future work directions can be categorized into the various sections of the algorithm, such as:

- Basic Volume Reconstruction
- Illumination Estimation
- Color Matching
- Surface Estimation

Volume Reconstruction

The two major limitations of this approach are due to the computational complexity and memory usage. Despite the optimizations discussed in Chapter 5, the algorithm can take several CPU hours to perform large reconstructions. The iterative nature of the opacity estimator is such that it requires good initial estimates to be effective. The ability to support multiple depth hypotheses simultaneously in the volume can be considered a double edged sword; where supporting multiple true depth estimates may be advantageous, evidence supporting false positives in the voxels are more likely to persist unless dominated and suppressed by true positive observations. When this is not possible, these false positives lead to visible noise in the final reconstruction. This noise can clearly be seen in the real data examples presented in Chapter 8. One suggested direction for future work would be the use of a final sweep though the volume using the algorithms defined in [36, 67]. A binary consistency check per voxel in this case could be used to eliminate false voxels and produce a cleaner model from real data.

Illumination Estimation

The outdoor illumination estimator described in Chapter 3 is successful in normalizing the surface color estimates before checking them for consistency. Admittedly, these estimates are based on very gross assumptions about the nature of both the surface properties and illumination effects. We presume that the input images are acquired in an outdoor environment during visible daylight conditions and that these conditions are modeled accurately using the CIE generalized sky model. In order for the algorithm to be applicable to images acquired for scenes other than outdoors, the illumination estimator must be modified to identify and compensate for more general spatially vary illumination.

Color Estimation

The novel color estimation algorithm is shown to improve on methods typically used for color consistency for volume reconstructions. The use of probability to model the colors leads to a simple and meaningful comparison of color distributions leading to a full gradation of consistency rather than a binary value. It can be argued that representing a single color with additive noise using a Normal distribution is acceptable. However, the assumption that color mixtures vary as a multi-variate Gaussian is inaccurate and would perhaps be better modeled as a multi-modal Gaussian where each mode represent a different color in the mixture. The colors are also represented and manipulated in CIE-Luv space which is calibrated according to a $D65$ white point. In the general case, this could be modified according to what might be known regarding the input images. Calibrating the color-space according to the type of light source e.g. incandescent, fluorescent, sunlight etc. would lead to improved color matching performance.

Gaussian Assumption for Depth

The main assumption of the work on depth image recovery is that depth estimates vary as Gaussian random variables. We make a single hypothesis at the depth (using the mean) despite the possibility of multiple hypotheses existing, allowing the variance in the depth estimate to capture this possibility. An improvement of this method could use a more sophisticated approach such as particle filters [33] along each ray and thereby supporting multiple hypotheses simultaneously. This would however, place a greater burden on the computational and storage requirements of the algorithm.

Surface Estimation

As mentioned earlier, a surface representation of a model is generally more favorable than a volumetric one since it is often more compact and easier to texture map. The depth estimation method described does not provide a complete representation of the 3-D model, but instead multiple surfaces, each representing a different portion of the scene. These multiple depth maps must be combined and textured appropriately to produce such a complete model. Methods already exist for combining sets of surface

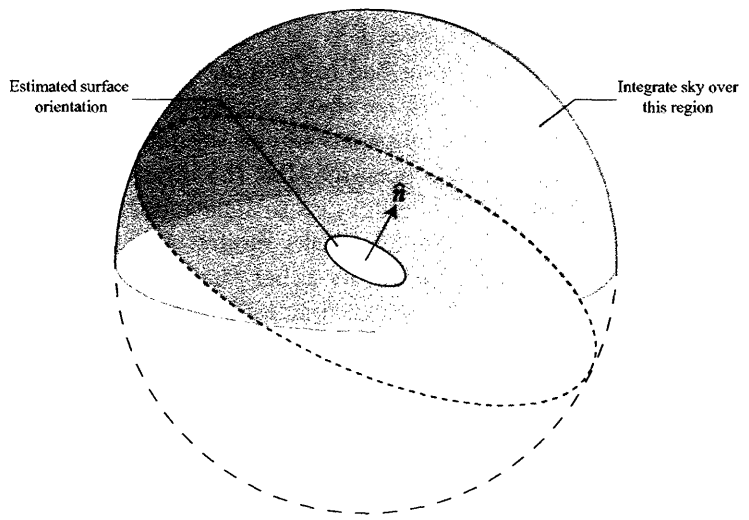


Figure 9.1. In the surface normal is known, a more accurate estimate of the illumination can be obtained by integrating over the visible portion of the sky.

points such as the Iterative Closest Point (ICP) method proposed by Besl [4] which uses rigid transforms to minimize the distance between two sets of surface points. Although these methods can sometimes fail for objects that contain surfaces with high curvature, we do not expect this to be an issue when dealing with models of urban environments. More robust volumetric integration methods also exist where voxels are used to store a level-set representation of the scene to form a complete implicit surface. A polygonized surface can then be found by locating the zero-crossings using a marching cubes or marching tetrahedra strategy.

The surface estimator itself is still in an early stage of development but could already be further utilized with the illumination model to better estimate surface reflectance. This would enable the use of more sophisticated BRDF models rather than the simple Lambertian model currently being used. If approximate surface estimates were computed at every iteration, although computationally expensive, would allow for better use of the illumination model by integrating over the portion of sky than irradiates each surface patch using its normal (Figure 9.1).

Appendix A

Color Space Conversion Formulae

A.1 RGB to XYZ

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{A.1}$$

A.2 XYZ to RGB

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 3.240479 & -1.537150 & -0.498535 \\ -0.969256 & 1.875992 & 0.041556 \\ 0.055648 & -0.204043 & 1.057311 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{A.2}$$

A.3 XYZ to CIE-*Luv*

The non-linear relations for $L*$, $u*$, and $v*$ are given below:

$$L* = 116 * (Y/Y_n)^{\frac{1}{3}} - 16$$

$$u* = 13L * *(u' - u'_n)$$

$$v* = 13L * *(v' - v'_n).$$

The quantities u'_n and v'_n refer to the reference white or the light source; for the 2 observer and illuminant C, $u'_n = 0.2009$, $v'_n = 0.4610$. Equations for u' and v' are given below:

$$u' = 4X/(X + 15Y + 3Z) = 4x/(-2x + 12y + 3)$$

$$v' = 9Y/(X + 15Y + 3Z) = 9y/(-2x + 12y + 3).$$

The transformation from (u', v') to (x, y) is:

$$x = 27u'/(18u' - 48v' + 36)$$

$$y = 12v'/(18u' - 48v' + 36).$$

A.4 CIE-*Luv* to XYZ

The transformation from CIE-*Luv* to XYZ is performed as following:

$$u' = u/(13L*) + un$$

$$v' = v/(13L*) + vn$$

$$Y = ((L * +16)/116)^3$$

$$X = -9Yu'/((u' - 4)v' - u'v')$$

$$Z = (9Y - 15v'Y - v'X)/3v'$$

Appendix B

Project Build Instructions

This chapter describes how to checkout code for the algorithm described in this thesis. Instructions are also given on how to build (compile) and execute the code. Before proceeding with these instructions, please ensure the following:

- You have an account with the MIT Computer Graphics Group,
- You have membership to user groups *graphics* and *city*,
- You are working at a computer running either UNIX or Linux.

B.1 Checkout and Compile Instructions

B.1.1 Code Checkout

First, check out the CVS source root tree of the `city` project:

```
% setenv CVSROOT /u5/city/
```

Next confirm that the CVSROOT environment variable is correctly set using:

```
% env | grep CVSROOT
```

Move to the directory in which you want to install the `city` project and checkout the entire directory:

```
% cvs checkout -P city_base
```

B.1.2 Compiling

Move to the `city_base/src` directory from the directory in which you installed the `city` project. To compile the entire tree, execute the following `make` command:

```
% make clean
```

and then

```
% make -k
```

To compile the `carve` portion of the project, first move to the carve directory:

```
% cd image/Apps/carve/carvesrc
```

and then:

```
% make clean
```

followed by

```
% make -k
```

■ B.2 Execute

To execute the carve project, simple run one of the scripts in the carve directory. For example, to execute the algorithm on the green building dataset:

```
% ./run_green
```

■ B.2.1 Command line switches

The carve program has a number of command line switches to simplify the execution on various datasets. A complete set of the switches along with short eplanations of usage can usage be recalled using:

```
% ./carvesrc/carve -
```

when in the carve directory.

Switch	Description	Default Value
node_path	Base Node Directory	
-I *image_path*	Use this as image directory	
-C *camera_path*	Use this as camera pose directory	
-SD	Use this as sphere directory	
-S *sphere_suffix*	Use this sphere file for 2D Referencing	
-start *start_node*	Start node number	
-end *end_node*	End node number	
-dnw *num*	Default node width string	4
-near *near_distance*	Camera Near Distance	200.0
-far *far_distance*	Camera Far Distance	5000.0
-noAdaptive	Do not use Adaptive sampling	
-vs *voxel_size*	Voxel Size	300.0
-voxdir *voxel_path*	Voxel directory	
-mergevoxlist *voxel_file_list*	A list of vox filenames	
-othreshold *opacity_threshold*	Remove voxels below opacity before division	0.0
-cthreshold *confidence_threshold*	Remove voxels below confidence before division	0.0
-savevoxfile *voxel_filename*	Save the current voxel space into a file	
-loadvoxfile *voxel_filename*	Load a voxel space from a file	
-bs	Perform background subtraction	
-bb	Assume black pixels to be background	
-cb	Estimate constant background	
-bgmdir *background_path*	background (sky) model directory	
-noinfo	No info file available	
-roxels	Use Roxels Exponential matching	
-Luv	Use Luv for matching instead of RGB	
-hsv	Use HSV for matching instead of RGB	
-linRad	Linearize Color values	
-iter *num_iterations*	Number of iterations (voxel divisions)	2
-pass *num_passes*	Number of passes of algorithm for each iteration	5
-q	Quiet mode	
-scale *scale_factor*	Factor to scale visualized sphere	100.0
-model *model_name*	Inventor Model File	
-batch	Batch mode	
-iterate	Force *num_iterations* iterations in batchmode	
-bbox *bbox_file*	Customize the bounding box	
-PID *carve_id*	Carve ID	
-fbg	Filter estimated background color from samples	
-fc	Filter sample ray as it passes through voxels	
-interp	Interpolate pixels in image	

Bibliography

[1] Hewlett packard website on srgb: http://www.srgb.com.

[2] Matthew E. Antone and Seth Teller. Scalable extrinsic calibration of omni-directional image networks. In *IJCV*, pages 143–174, 2002.

[3] A. Azarbayejani and A.P. Pentland. Recursive estimation of motion, structure, and focal length. *PAMI*, 17(6):562–575, June 1995.

[4] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 14, pages 239–256, February 1992.

[5] Christopher M. Bishop. *Neural networks for pattern recognition.* Oxford University Press, 1996.

[6] A. Blake, A. Zisserman, and G. Knowles. Surface descriptions from stereo and shading. *IVC*, 3(4):183–191, 1985.

[7] R.C. Bolles, H.H. Baker, and D.H. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *IJCV*, 1(1):7–56, 1987.

[8] David H. Brainard and William T. Freeman. Bayesian color constancy. *Optical Society America*, 14(7):1393–1411, 1997.

[9] P. Burt and E. Adelson. The laplacian pyramid as a compact image code, 1983.

[10] F. J. Canny. A computational approach to edge detection. *IEEE Trans PAMI*, 8(6):679–698, 1986.

[11] Shenchang Eric Chen. Quicktime VR – an image-based approach to virtual environment navigation. In *SIGGRAPH '95 Conference Proceedings*, pages 29–38, aug 1995.

[12] R.L. Cook and K.E. Torrence. A reflectance model for computer graphics. *Computer Graphics*, 15(4):187–196, July 1981.

[13] Satyan Coorg. *Pose Imagery and Automated Three-Dimensional Modeling of Urban Environments.* PhD thesis, MIT, 1998.

[14] Satyan Coorg and Seth Teller. Matching and pose-refinement with camera pose estimates. In *Proc. DARPA IUW*, May 1997.

[15] Barbara M. Cutler. Aggregating building fragments generated from geo-referenced imagery into urban models. Master's thesis, Dept. of Electrical Engineering and Computer Science, MIT, 1999.

[16] G.B. Davis, D.J. Griggs, and G.D. Sullivan. Automatic detection of cloud amount using computer vision. *Atmospheric and Oceanic Technology*, 9:81–85, Feb 1992.

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

[18] J. S. DeBonet and P. Viola. Roxels: Responsibility weighted 3d volume reconstruction. In *ICCV*, September 1999.

[19] S. Demey, A. Zisserman, and P. Beardsley. Affine and projective structure from motion, 1992.

[20] U. R. Dhond and J. K. Aggarwal. Structure from stereo – A review. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(6):1489–1510, November-December 1989.

[21] C. Dyer. Volumetric scene reconstruction from multiple views, 2001.

[22] O.D. Faugeras. *Three-Dimensional Computer Vision.* MIT Press, 1993.

[23] O.D. Faugeras, S. Laveau, L. Robert, G. Csurka, and C. Zeller. 3-d reconstructions of urban scenes from sequence of images. In *INRIA*, 1995.

[24] G.D. Finlayson, M.S. Drew, and B.V. Funt. Diagonal transforms suffice for color constancy. In *ICCV93*, pages 164–171, 1993.

[25] William T. Freeman and David H. Brainard. Bayesian decision theory, the maximum local mass estimate, and color constancy. In *ICCV*, pages 210–217, 1995.

[26] B. Funt and V. Cardei. Bootstrapping color constancy, 1999.

[27] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *Pattern Recognition and Machine Intelligence*, 6(6):721–741, November 1984.

[28] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. In *SIGGRAPH '96 Conference Proceedings*, pages 43–54, August 1996.

[29] C.W. Greeve. *Digital Photogrammetry: an Addendum to the Manual of Photogrammetry.* American Society of Photogrammetry and Remote Sensing, 1997.

[30] C. Harris and M.J. Stephens. A combined corner and edge detector. In *Alvey88*, pages 147–152, 1988.

[31] Berthold K. P. Horn and Michael J. Brooks. *Shape from Shading.* The MIT Press, Cambridge, MA, 1989.

[32] Berthold Klaus Paul Horn. *Robot Vision.* MIT Press, Cambridge, MA, 1986.

[33] Michael Isard and Andrew Blake. Contour tracking by stochastic propagation of conditional density. In *European Conference on Computer Vision*, pages 343–356, 1996.

[34] Takeo Kanade, Peter Rander, and P. J. Narayanan. Virtualized reality: Constructing virtual worlds from real scenes. *IEEE MultiMedia*, 4(1):34–47, - 1997.

[35] Sing Bing Kang and Richard Szeliski. 3-d scene data recovery using omnidirectional multibaseline stereo.

[36] K. Kutulakos and S. Seitz. A theory of shape by space carving. Technical Report TR692, Computer Science Dept., U. Rochester, 1998.

[37] Eric P. F. Lafortune, Sing-Choong Foo, Kenneth E. Torrance, and Donald P. Greenberg. Non-linear approximation of reflectance functions. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 117–126. ACM Press/Addison-Wesley Publishing Co., 1997.

[38] E. H. Land. An alternative technique for the computation of the designator in the retinex theory of color vision. *Proceedings of National Academical Science*, 83:3078–3080, 1986.

[39] E. H. Land. Recent advances in retinex theory. *Vision Research*, 26:7–21, 1986.

[40] E. H. Land and J. J. McCann. Lightness and retinex theory. *Journal of the Optical Society of America*, 61:1–11, 1971.

[41] R. Lenz and R. Tsai. Techniques for calibration of the scale factor and image center for high accuracy 3D machine vision metrology. In *Proc. IEEE International Conf. on Robotics and Automation*, 1987.

[42] Marc Levoy and Pat Hanrahan. Light field rendering. In *SIGGRAPH '96 Conference Proceedings*, pages 31–42, August 1996.

[43] So Zu Li. *Markov Random Field Modeling in Computer Vision.* Springer-Verlag, 1995.

[44] H.C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.

[45] D. Marr and T. Poggio. Cooperative computation of stereo disparity. Technical report, 1976.

[46] S. Marschner, S. Westin, E. Lafortune, and K. Torrance. Image-based brdf measurement, 2000.

[47] W. Matusik, H. Pfister, M. Brand, and L. McMillan. A data-driven reflectance model. *Computer Graphics*, 37(Annual Conference Series), 2003.

[48] Wojciech Matusik, Chris Buehler, and Leonard McMillan. Polyhedral visual hulls for Real-Time rendering. pages 115–126.

[49] Wojciech Matusik, Chris Buehler, Leonard McMillan, and Steven Gortler. An efficient visual hull computation algorithm.

[50] Wojciech Matusik, Chris Buehler, Ramesh Raskar, Steven J. Gortler, and Leonard McMillan. Image-based visual hulls. In Kurt Akeley, editor, *Siggraph 2000, Computer Graphics Proceedings*, pages 369–374. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000.

[51] Leonard McMillan and Gary Bishop. Plenoptic modeling: An image-based rendering system. In *SIGGRAPH '95 Conference Proceedings*, pages 39–46, August 1995.

[52] JP Mellor, Seth Teller, and Tomás Lozano-Pérez. Dense depth maps for epipolar images. *Proc. DARPA IUW*, May 1997.

[53] R. Mohr, F. Veillon, and L. Quan. Relative 3D reconstruction using multiple uncalibrated images. In *CVPR93*, pages 543–548, 1993.

[54] S. Muller, W. Kresse, and F. Schoeffel. A radiosity approach for the simulation of daylight, 1995.

[55] D. Nitzan, A.E. Brain, and R.O. Duda. The measurement and use of registered reflectance and range data in scene analysis. In *IEEE*, volume 65, pages 206–220, 1977.

[56] M. Okutomi and T. Kanade. A multiple-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4):353–63, 1993.

[57] Richard Perez, R. Seals, and J. Michalsky. All-weather model for sky luminance distribution-preliminary configuration and validation., 1993.

[58] B. Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18(6):311–317, 1975.

[59] M. Pollefeys, R. Koch, M. Vergauwen, A. A. Deknuydt, and L. J. Van Gool. Three-dimensional scene reconstruction from images. pages 215–226.

[60] Charles A. Poynton. Frequently asked questions about color. FAQ available at http://home.inforamp.net/~poynton/ColorFAQ.html.

[61] A. J. Preetham, Peter Shirley, and Brian E. Smits. A practical analytic model for daylight. In Alyn Rockwood, editor, *Siggraph 1999, Computer Grahics Procedings*, pages 91–100, Los Angeles, 1999. Addison Wesley Longman.

[62] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes: The Art of Scientific Computing.* Cambridge University Press, Cambridge (UK) and New York, 2nd edition, 1992.

[63] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing (2nd ed.).* Cambridge University Press, Cambridge, 1992.

[64] P. Pritchett and A. Zisserman. Wide baseline stereo matching. In *ICCV98*, pages 754–760, 1998.

[65] A.C. Prock and C.R. Dyer. Towards real-time voxel coloring. In *DARPA98*, pages 315–321, 1998.

[66] Charles Rosenberg, Martial Hebert, and Sebastian Thrun. Color constancy using kl-divergence. In *ICCV01*, pages 239–246, 2001.

[67] S.M. Seitz and C.R. Dyer. Photorealistic scene reconstruction by voxel coloring. In *CVPR97*, pages 1067–1073, 1997.

[68] S.M. Seitz and K.N. Kutulakos. Plenoptic image editing. In *ICCV98*, pages 17–24, 1998.

[69] J.E. Shields, T.L.Koehler, M.E. Karr, and R.W. Johnson. Automated cloud cover and visibility systems for real time applications, 1990.

[70] Ilya Shlyakhter, Max Rozenoer, Julie Dorsey, and Seth J. Teller. Reconstructing 3d tree models from instrumented photographs. *IEEE Computer Graphics and Applications*, 21(3):53–61, 2001.

[71] C.C. Slama. *Manual of Photogrammetry.* American Society of Photogrammetry and Remote Sensing, 1980.

[72] R. Szeliski. Bayesian modeling of uncertainty in low-level vision. *International Journal of Computer Vision*, 5(3):271–301, 1990.

[73] R. Szeliski and S. B. Kang. Recovering 3D shape and motion from image streams using non-linear least squares. *Journal of Visual Communication and Image Representation*, 5(1):10–28, 1994.

[74] Richard Szeliski and Polina Golland. Stereo matching with transparency and matting. In *ICCV*, pages 517–526, 1998.

[75] C. J. Taylor and D. J. Kriegman. Structure and motion from line segments in multiple images. *PAMI*, 17(11):1021–1032, November 1995.

[76] Seth Teller. Automatic acquisition of hierarchical, textured 3D geometric models of urban environments: Project plan. In *Proceedings of the Image Understanding Workshop*, 1997.

[77] Seth Teller. Automated urban model acquisition: Project rationale and status. In *Proceedings of the Image Understanding Workshop*, 1998.

[78] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9:137–154, 1992.

[79] R. Tsai. A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automation*, RA-3(4), August 1987.

[80] Y. Tsin, R. Collins, V. Ramesh, and T. Kanade. Bayesian color constancy for outdoor object recognition, 2001.

[81] Yair Weiss. Interpreting images by propagating bayesian beliefs. *M. C. Mozer, M. I. Jordan and T. Petsche, editors, Adnaces in Neural Information Processing Systems*, pages 908–915, 1997.

[82] Lance Williams. Pyramidal parametrics. *Computer Graphics (SIGGRAPH '83 Proc.)*, 17(3):1–11, July 1983.

[83] P.R. Wolf. *Elements of Photogrammetry*. McGraw-Hill, 1974.

[84] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Understanding belief propagation and its generalizations. In *IJCAI*, 2001.

[85] Yizhou Yu and Jitendra Malik. Recovering photometric properties of architectural scenes from photographs. In *SIGGRAPH '98 Conference Proceedings*, pages 207–217, 1998.