Segmentation and Interpretation Using Multiple Physical Hypotheses of Image Formation

Article · December 1999

Source: CiteSeer

CITATIONS
5

READS
113

1 author:

Bruce A. Maxwell
Colby College
122 PUBLICATIONS **1,879** CITATIONS

SEE PROFILE

Segmentation and Interpretation Using Multiple Physical Hypotheses of Image Formation

Bruce A. Maxwell

CMU-RI-TR-96-28

Submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Robotics

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

26 July, 1996

Thesis Committee

Dr. Steven A. Shafer, Advisor

Dr. Katsu Ikeuchi

Dr. Andrew Witkin

Dr. Linda Shapiro

This research was sponsored by the Department of the Army, Army Research Office under grant number DAAH04-94-G-0006. Views and conclusions contained in this document are those of the author and should not be interpreted as necessarily representing official policies or endorsements, either expressed or implied, of the Department of the Army or the United States Government.

Abstract

One of the first, and most important tasks in single image analysis is segmentation: finding groups of pixels in an image that “belong” together. A segmentation specifies regions of an image that we can reason about and analyze. Having an accurate segmentation is a prerequisite for vision tasks such as shape-from-shading. A general-purpose segmentation algorithm, however, does not currently exist. Furthermore, the output of many segmentation algorithms is simply a set of pixel groupings; no attempt is made to provide a physical description of or make a connection between the image regions and objects in the scene.

Physics-based segmentation algorithms are based upon identifying coherent regions of an image according to a model of object appearance. These models have usually assumed that a scene contains a single material type, restricted forms of illumination, and uniformly colored objects. This work challenges these assumptions by considering multiple physical hypotheses for simple image regions. For each initial region, the framework proposes a set of hypotheses, each of which specifically models the illumination, reflectance, and shape of the 3-D patch which caused that region. Each hypothesis represents a distinct, plausible explanation for the color and intensity variation of that patch. The framework proposes comparing hypotheses of adjacent patches for similarity and merging them when appropriate, resulting in more global hypotheses which group elementary regions that are part of the same surface. A physical analysis of the hypotheses reveals strong constraints on hypothesis compatibility which significantly reduces the space of possible image interpretations and specifies which hypothesis pairs should be tested for compatibility.

A consequence of this framework is a new approach to segmenting complex scenes into regions corresponding to coherent surfaces rather than merely regions of similar color. The second part of this work presents an implementation of this new approach and example segmentations of scenes containing multi-colored piece-wise uniform objects.

The algorithm contains four phases.The first phase segments an image based upon normalized color. This provides a set of simple regions that can reasonably be assumed to be part of a single object. The algorithm then attaches a list of potential explanations, or hypotheses to each initial region.

The second phase examines adjacent hypotheses for compatibility. This work explores two methods of compatibility testing. The first is direct comparison, which estimates the shape, illumination, and material properties of each region and directly compares their compatibility. Unfortunately, this approach is limited by the tools for estimating the physical properties of an unconstrained scene. The second method uses weak tests of compatibility, which compare physical characteristics that must be compatible if two hypotheses are part of the same surface, but which are not necessarily incompatible between different objects. By using a number of these necessary, but not sufficient tests the algorithm rules out most incompatible hypothesis pairs.

The third phase builds a hypothesis graph from the results of the analysis. Each hypothesis is a node in the graph, and edges contain information about the cost of merging adjacent hypotheses. The fourth phase then extracts segmentations from the hypothesis graph and rank-orders them. Each segmentation contains exactly one hypothesis from each region and provides a potential physical interpretation of the scene.

After presenting the basic algorithm this work expands it in two directions. First, it explores the issue of preferring certain hypotheses over others depending on their compatibility with the image data. Second, it shows how the algorithm can expand to handle scenes of greater complexity by expanding the initial list of hypotheses and developing new tests of hypothesis compatibility.

Overall, this new algorithm, based on a general framework, is able to intelligently segment scenes with objects of greater complexity than previous physics-based segmentation algorithms. The results show that by using general physical models, the resulting segmentations correspond more closely to coherent surfaces in the scene than segmentations found using only color.

Acknowledgements

First and foremost I want to thank my advisor, Dr. Steven Shafer, for his energy, effort, and committment over the past four years. When I arrived at CMU, I had very little idea of who I wanted to work with, or even in what field. Steve's energy and vision persuaded me to work with him. Since then he has taught me how to organize my thoughts, think critically about the field of vision, and present my work in a persuasive and entertaining manner. He gave me the freedom to pursue my ideas, and the guidance necessary to keep me on track. Most importantly, he has believed in, supported, and defended my work. I want to express my thanks to him and wish him well in the future.

Thank you to my committee, Katsu Ikeuchi, Andy Witkin, and Linda Shapiro. Their suggestions and insightful comments helped me to maintain a high standard in my work.

I also wish to thank Martial Hebert. Even though he wasn't on my committee, he might as well have been. His support, comments, and innumerable letters of recommendation, were invaluable.

Thanks to Mark Maimone, Yalin Xiong, Reg Willson, and John Krumm. They introduced me to the Calibrated Imaging Lab, showed me the ropes, and helped me to understand the details of the field of vision. Yalin also had to put up with me as an office-mate for three years, a difficult task at best. Their willingness to answer questions about the inner workings of the CIL even after leaving CMU never ceased to amaze me.

Thanks to my wife, who had to put up with me, and without me, during the dark days of research and thesis writing. I hope she thinks it was worth it.

Thanks to my good friends and fellow Robograds, Ian Davis, Rob Driskill, and Barry Brummitt. They introduced me to the Robotics Institute, the dinner-coop, and the Viking Death Rat sports teams. The latter two organizations made up a significant portion of my social life for the past four years. I will miss them.

The support staff at CMU also deserves thanks. Patty Mackiewicz, Marce Zaragoza, Carolyn Kraft, Stephanie Riso, and Marie Elm all went out of their way to help me out. Thanks also to the CS facilities and hardware folks. This place would fall apart without them.

There are so many other people at CMU who made the past four years both enjoyable and challenging. They include Pete Rander, John Hancock, John Murphy, Rich Voyles, Dan Morrow, Harry Shum, Murali Krishna, Henry Schneiderman, Justin Boyan, Mike Nechyba, Lalitesh Katragadda, Chris Lee, Pat Rowe, Zack Butler, Jennifer Kay, Mike Smith, Mei Chen, Shumeet Baluja, Ben Brown, Jim Moody, Kate Fissel, Bill Ross, John Bares, Fritz Morgan, Dave Simon, Todd Jochem, Yangsheng Xu, Gary Fedder, Matt Mason, Mike Erdmann, and all of the faculty, staff, and robograds who make the Robotics Institute a great place to be.

Thanks to Dave Blankenship and Noah Salzman for helping get the CIL in order.

Thanks to all of the dinner-coop members who had to listen to me ramble on about my thesis, and thanks to Karen Haigh for putting the coop firmly on the WWW map.

Finally, thanks to the ECE basketball crew who put up with me learning how to play and helped to keep the stress level in my life to a reasonable level.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

Figure 1.1: Image containing objects made of different types of materials reflecting complex illumination and possessing varying shapes.

1.1. Overview

Solving the General Vision Problem is the Holy Grail of vision research. Broadly stated, the General Vision Problem is understanding a color image: the problem of identifying coherent surfaces or objects from a single image and explaining why they appear the way they do. A solution to the General Vision Problem would provide a description of objects and their incident illumination in images like Figure 1.1, which contain multiple items of differing materials and shape, many displaying interreflection between themselves and their neighbors. Ultimately, a solution requires understanding the physical phenomena that create a given image. That a solution exists for humans is manifest: an individual can look at a picture such as Figure 1.1 and provide a detailed physical description of the objects in the scene.

In the past decade much of the vision community has moved away from the General Vision Problem and focused on more constrained images or image sets such as those used for stereo, motion, active vision, and photometric stereo. For tasks like robot navigation, obstacle avoidance, and object tracking analyzing single images is not only slower, but also less accurate than using multiple camera systems and multiple image algorithms such as stereo.

Understanding single images, however, is still essential for any task where an active agent or video imagery is not available. Searching image data-bases by content is one example of an important task whose input consists of only single images. Furthermore, for unlabeled images there is no information about the materials, illumination, or shape of the objects in these images. We are presented with only the image data. Therefore, we do not have the luxury of using multiple image algorithms, active vision techniques, or tightly controlled image environemnts. Instead, we must return to our study of the General Vision Problem and understanding single images.

One of the first, and most important tasks in single image analysis is segmentation: grouping pixels that appear to "belong" together. A segmentation provides regions of an image that can be reasoned about and analyzed as a whole. The prior segmentation of an image is a prerequisite for analysis methods such as shape-from-shading to work.

This thesis focuses on the problem of segmenting a single color image by reasoning about the physics of image formation. Physics-based segmentation is difficult because of the complexity of the interaction of light and matter. Figure 1.1, for example, contains specular highlights on a number of the surfaces, complex shapes, interreflection as between the pails and the copper kettle, occlusion of objects by other objects, occlusion of the light sources or shadows, multiple light sources, and different materials.

1.2. History and previous work

Segmentation methods to date fall into two categories: feature-based segmentation, and physics-based segmentation. Feature-based segmentation methods divide an image into pixel groups according to characteristics such as color, intensity, or hue using standard image processing techniques such as region growing, region merging, and region splitting. This class of algorithms are based upon straightforward statistical models of the image data and do not search for underlying symbolic of physical meaning.

The statistical approach was taken partly because of the optimism of the 70's surrounding symbolic reasoning and artificial intelligence, which relegated to low-level vision the straightforward task of dividing an image into simple regions based upon color and brightness. More extensive low-level processing was considered unnecessary because it was assumed that programs using higher level reasoning would be able to understand, identify, and merge these simple regions as appropriate.

The segmentation in computer vision began with the work of Brice & Fenema in 1970, who first proposed scene analysis using regions [6]. They modeled images as regions with Gaussian distributions of color and intensity.

Based on this approach, Yakimovsky & Feldman developed a system for analyzing complex natural scenes [61]. Their approach started with unsupervised clustering of pixels followed by a step which merged similar regions based on their distance in feature space. This was followed by the application of a world model in an attempt to classify what each region represented. For example, the knowledge that trees are generally above and beside a road influenced the interpretation of green regions on the upper right or left of an image. His work showed reasonable results for two domains: road scenes while driving, and left ventricular angiograms.

The limitations of segmentation using just the R, G, B values as features, led to a search for the best set of color features for segmentation. Ohlander's thesis in 1975, for example, examined the histograms of nine color features and used region splitting to subdivide an image [44]. Ohlander also was the first to reason about shadows, occlusions, and highlights to obtain better segmentations. While this line of research showed promise, the lack of adequate reflection models and the use of feature-based segmentation as a first step limited the development of algorithms to reason about the image.

Because none of the standard color features would always return good results, researchers

explored dynamically changing the color features as the segmentation proceeded. In 1980, Ohta, Kanade, and Sakai implemented an Ohlander-style segmentation, but used a Karhunen-Loeve transformation of the R, G, B data to determine the optimal color features at each step [45]. Their results showed that dynamically changing the features was not necessary. From their analysis of what they claimed was a representative set of images, they proposed an "optimal" set of color features that were linear combinations of R, G, and B. Despite its "optimality," however, the segmentations still divided curved or shaded objects into small regions and did not attempt to find symbolic meaning in images.

The greatest drawback of feature-based segmentation methods is that they do not take into account the physics of image formation. Therefore, they are likely to split objects or surfaces that contain texture, varying intensity, and phenomena such as highlights. For example, according the Ohta, Kanade, and Sakai, the best color feature for segmenting an image is intensity. Intensity, however, will vary as much over the surface of a single object as it will between objects. For the most part, researchers understood that using knowledge about the world was important, but they tried to incorporate such knowledge on top of their statistical models. This ultimately crippled the computer's ability to reason effectively about image regions. To find objects in an image and obtain an intelligent grouping of pixels, the physics of image formation must guide the segmentation process.

Physics-based segmentation grew out of this idea. It attempts to use models of the interaction of light and matter to predict the appearance of objects under various conditions. It's goal is to find image regions that correspond to symbolic scene elements. In practical terms, this means finding one or more physical descriptions of the illumination, materials, and geometry that created the image so that subdivisions of it correspond to coherent surfaces or objects in the scene.

The first step to understanding object appearance is the development of models of the interaction of light and matter. The most commonly used model for body reflection is Lambert's law [17]. Torrance & Sparrow and Beckman & Spizochino were the first to develop models of surface reflection, or specular reflection based on geometry and electromagnetics, respectively [57][3]. However, they did not apply these models to computer vision.

Horn was the first to propose using physical models of image formation to analyze and understand images [18]. Theoretically, using Horn's model some physical characteristics of a surface, including shape, could be estimated from a single image. Unfortunately, the model was limited to perfectly diffuse, perfect reflecting surfaces--also known as white Lambertian surfaces--and point light sources. Horn's methods also assumed a single surface or prior segmentation and a single point light source in the scene. Furthermore, as it did not allow for noisy images or camera limitations--e.g. clipping of the color values to the camera's range--it was not easily applicable to real images.

One of the key steps towards achieving a physical understanding of real images was Shafer's dichromatic reflection model [53], presented in 1985. It combined the body and surface reflection models previously developed into an integrated model by noting that for inhomogeneous dielectrics--materials such as plastic with colored pigment particles held together by a clear medium--the appearance of an object was defined by the plane formed by the body color and surface reflection color. It allowed researchers, for the first time, to understand the appearance of a large class of actual materials: paints, plastics, acrylics, ceramics, and paper.

Figure 1.2: Scene containing inhomogeneous dielectric objects of uniform color.

Healey subsequently proposed the unichromatic reflection model for metals [16]. He showed that metals only exhibit surface reflection, but may modify the color of the surface reflection unlike inhomogeneous dielectrics.

More recently, Nayar et. al., and He et. al. have both proposed comprehensive reflection models which combine elements of the Beckman Spizzochino model, Torrance & Sparrow model, and Lambertian reflection [40][15]. Alternatives to Lambertian reflection are also being studied to more closely model the appearance of objects such as ceramics. The most comprehensive global model of reflection was put forward by Forsyth & Zisserman [14]. They propose a complete radiosity solution to reflection which incorporates interreflection between surfaces as well as multiple light sources. As such a solution requires a significant amount of knowledge about the scene, however, it has not yet been used for practical analysis.

Physics-based segmentation algorithms use these models of reflection to predict the appearance of objects in a scene. A highlight on a plastic object such as one of the donuts in Figure 1.2, for example, fits the skewed-T model developed by Klinker, Shafer, & Kanade [24]. Therefore, a physics-based algorithm will merge the highlight with the rest of the donut despite the difference in color and intensity, while a feature-based segmentation algorithm would divide the donut into two separate regions.

Previous physics-based segmentation methods have limited their analysis to scenes such as Figure 1.2 containing uniformly colored objects of known material types. They segment such scenes into regions corresponding to objects using color and one or two physical models to account for intensity and chromaticity variations due to geometry, interreflection, and highlights. These methods assume that a discontinuity in color between two image regions implies a discontinuity in other physical characteristics such as shape and reflectance.

Klinker, Shafer, & Kanade developed the first true physics-based segmentation method in the late ‘80s [24]. It uses the skewed-T model of the appearance of inhomogeneous dielectrics predicted by Shafer’s dichromatic reflection model. The algorithm begins by finding blocks of the image corresponding to areas of an object exhibiting only body reflection. Such regions vary in intensity, but not chromaticity, or normalized color, and thus form a linear cluster in R, G, B color space.

After identifying linear clusters, the program finds planar clusters that fit a skewed-T in color space, with the stem of the T being the highlight region of the object. Finally, the algorithm finishes the segmentation by modeling camera effects--e.g. clipping and blooming--which distort the skewed-T and fitting pixels affected by them into their appropriate regions.

Despite the power of this segmentation program, it was still applicable to a limited class of images. Metals or multi-colored objects could not be correctly segmented. Furthermore, the assumptions of Klinker *et al.* included a single color of illumination. This resulted in incorrect segmentations in regions with colored interreflection from nearby objects.

Finding solutions for these limitations was the next step in physics-based vision. Bajcsy *et al*. attempted to model interreflection and improve the parameter estimation by using hue, saturation, and intensity along with histogram analysis and splitting [2]. By also using a white reference card or object, their algorithm could discount the effects of global illumination, allowing it to identify regions of objects displaying interreflections, shadows, and highlights.

Brill proposed a slightly different model from Klinker, Shafer, & Kanade for inhomogeneous dielectrics and demonstrated its use in segmentation, but did not improve upon the capabilities of the planar cluster analysis [7].

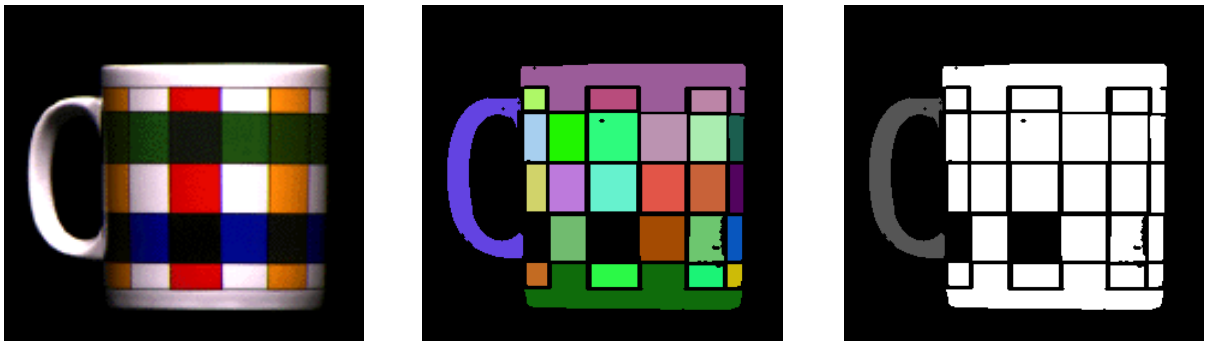
To expand the permissible range of materials in an image, Healey proposed the unichromatic reflection model for metals [16]. He then showed it could be used with the dichromatic reflection model to segment images containing both metals and inhomogeneous dielectrics under specific lighting conditions. His method used region splitting to find portions of the image that were coherent according to either the dichromatic or unichromatic reflection models.

As a result of these efforts, the vision community could claim it could segment images containing two materials--inhomogeneous dielectrics and metals--and images containing interreflection; but these methods have limitations. For Bajcsy *et. al.* to correctly identify interreflection, for example, a white reference plate or object is necessary in order to negate the effects of the global illumination, and the chromaticity of the global illumination must be constant over all surfaces in the image. Healey's method of differentiating metals and plastics requires a single white light source, and the metal cannot reflect other objects in the scene. None of these methods can handle multiple light sources with different chromaticities, and all of the major methods of physics-based segmentation of color images have assumed uniformly colored objects.

1.3. Developing a new framework

For physics-based segmentation to move ahead, at least the last requirement--that the scene contain only uniformly colored objects--must be relaxed. For example, shape-from-shading methods and illuminant direction estimators rely upon intensity values in an image, which means that combining the results of differently colored regions of an image without normalization will cause inaccuracies. A prior segmentation into objects allows for correct normalization of the intensities across region boundaries.

Beyond the needs of other algorithms, physics-based segmentation needs a larger framework in which to fit the specific reflection models. Previous methods have used a limited number of models in an ad-hoc manner, directly fitting the known models to the data using strict assumptions about the objects and illumination. To begin to relax assumptions about objects and their environ-



(a) (b) (c)

Figure 1.3: (a) Picture of a ceramic multi-colored mug taken in the Calibrated Imaging Laboratory, (b) segmentation based on chromaticity, (c) segmentation based on intrinsic characteristics: shape, illumination, and material type.

ment we must have a larger framework within which we can reason about and select the models we need to apply.

Why do multi-colored objects pose a problem to segmentation algorithms? Multi-colored objects like the mug in Figure 1.3(a), for example, obey the same reflection models used by previous physics-based methods. The change in color between two the image regions, however, does not necessarily imply a discontinuity in shape or other characteristics. To correctly interpret scenes containing more complex objects such as the mug, multiple physical characteristics such as shape, illumination, and material type must be examined to determine whether two adjacent regions belong to the same object. Previous physics-based vision methods would stop with a segmentation of the mug image like Figure 1.3(b). A better and more useful segmentation, however, is shown in Figure 1.3(c), which shows the image divided into coherent surfaces except for black regions of the image.

The difficulty inherent in segmenting images potentially containing multiple materials and multi-colored objects is that by expanding the space of physical models considered for the shape, illumination, and material optics, a given image region can be described by a subspace of the general models; each point within this subspace is a valid explanation for the image region. Therefore, the mapping from appearance to physical explanation is one to many.

This is shown graphically in Figure 1.4. The boxes show three roughly identical image regions with radically different physical explanations. One region is formed by curved plastic under white directional illumination, one by shiny grey metal reflecting colored illumination, and one is a shaded diffuse planar surface under directional white illumination. Ideally, the region on the right that is part of a photograph should be grouped with the rest of the pixels that correspond to the photograph as they share a common surface and illumination. However, the region in the middle should not be grouped with the orange ring above it or the white base below. To make life even more difficult, the left-most box, being the surface of a mirror, should be grouped with its neighbors because they share coherent surface and illumination characteristics.

This complexity also appears in less complex images such as the mug in Figure 1.3(a). One of the squares on the mug, in isolation, could be explained as a red object under white light, a white object under red light, or a grey metal object reflecting a red environment. All of these are valid explanations for the image region. Therefore, to segment an image with multi-colored objects not

Figure 1.4: Image of an object (center), a mirror image of the object (left) and a photograph of the object (right). The highlighted regions are nearly identical in appearance but have different physical explanations.

only the model parameters, but also the models themselves must be correctly selected to accommodate qualitatively different shapes, materials, and illumination environments.

Independently of computer vision, Rissanen has analyzed model-based analysis from an information theoretic point of view and divided the problem space into three levels [51]. The first level is simple parameter estimation given a specific model. Previous physics-based segmentation methods fall into this category; they compare the image data to a reflectance model and estimate the parameters. A level two analysis involves selecting the best model from within a class, such as the best order polynomial for a set of data from the class of all polynomials. This type of analysis contains two parts: selecting the specific model from within its class, and then estimating its parameters from the data. The third level of analysis involves selecting the class of the model to use as well as the specific model and the parameters.

In order to handle quantitatively different physical explanations for an object's appearance, we must move the analysis from the primitive level 1 analysis--estimating parameters of a previously established model--to a level 3 analysis--selecting the model class--with a resultant increase in both problem complexity and perceptual power.

It is important to note that statistical and information theoretic methods exist for a level 2 analysis. Tools such as Rissanen's Minimum Description Length or Akaike's Information Criterion are effective measures of the cost of using a given order model versus the error between the model and the data. For a level 3 analysis, however, reasoning, intuition, and human experience are currently the best tools available. Therefore, a general framework must depend on reasoning about the physics of image formation and an object's appearance to select and instantiate the appropriate model classes.

Model selection, or instantiation has only recently been introduced to physics-based vision. Breton *et al*. have presented a method for selecting specific models from within a model class for both the illumination and shape [5]. For small patches on a surface, they discretize the space of illuminant directions and find the local shape based on each direction. By then selecting and combining the most compatible hypothetical surfaces a global shape for the object emerges. This method still considers only a single model for material type (Lambertian), and depends on the parameter space being small so that it can explore all possibilities.

To create a more general framework for segmentation, we must continue to expand the model space and consider multiple, quantitatively different physical explanations, or hypotheses for basic image regions. Furthermore, we have to understand the nature of a general solution to the segmentation problem. A single image may be ambiguous; we cannot simply select a single explanation for an image region or scene, but must entertain several possibilities. In other words, we can never expect to get *the* single correct interpretation of an image, only one or more *possible* correct interpretations.

This framework should also be able to reason about the possible interpretations of an image and make suggestions about which interpretations are better than others. This involves comparing the physical explanations to the image data and determining how "weird" a given explanation might be relative to other explanations for the same region.

This thesis presents a general framework for segmentation using multiple hypotheses of image formation that attempts to follow these guidelines. It then presents a segmentation algorithm based upon that framework and shows the results on a set of real test images containing multiple multi-colored piece-wise uniform dielectric objects. The chapters are organized as follows.

Chapter 2 examines the general parametric models of light and reflectance and develops a set of broad model classes that span the space of light and material interaction. It also shows that reasoning about these broad classes produces constraints that a segmentation algorithm can use to significantly reduce the search space of interpretations of an image.

Chapter 3 describes the segmentation system that grew out of the theoretical framework. The system is divided into four parts: the initial segmentation of the image into simple regions, the analysis of adjacent regions, the development of a hypothesis graph describing the interaction of neighboring physical explanations, and the extraction of the best image segmentations from the hypothesis graph.

Chapter 4 and Chapter 5 detail the specific methods of analysis used to compare adjacent regions and their hypothesized physical explanations. Chapter 6 then works through an example image and discusses in some detail the complexity of images containing multi-colored objects.

Chapter 7 shows how the framework can be modified to reflect the relationship between hypotheses and the image data they represent. This allows the system to prefer segmentations whose constituent hypotheses are more compatible with the image data.

Chapter 8 then describes how the system can be expanded according to the guidelines of the general theoretical framework to deal with more complex hypotheses and images. In particular, it shows how highlights are incorporated into the framework and that the system correctly interprets images containing multi-colored objects with highlights.

Finally, Chapter 9 discusses the relevance and importance of the framework, presents some concluding remarks, and suggests some directions for future research.

Chapter 2: Developing a theoretical framework

Images form when light strikes an object and reflects towards an imaging device such as a camera or an eye. The color and brightness of a point in an image is the result of the color and intensity of the incident light, and the shape and optical properties of the object. This chapter begins by presenting a formal model of these elements, how they interact, and how they are related to what we see in an image. Note that this description of image generation neglects camera effects such as those described by Willson [59].

The chapter then develops a taxonomy of scene elements, identifying useful subspaces of the general scene models. These subspaces form a set of broad classes for each scene element. Each image region has multiple physical explanations within the general scene model. Using the broad classes of the scene elements we can enumerate the set of physical explanations that could generate a given image region.

The next step is to explore how these region explanations interact. Given an initial segmentation of an image into simple regions, each region is described by a set of physical explanations. From the viewpoint of segmentation, we are interested in knowing which physical explanations within these sets could merge to form aggregate surfaces. An analysis of the broad classes shows that the physics of a scene strongly constrains this problem.

The chapter concludes by examining the pairs of physical explanations that can potentially merge to form coherent surfaces. Determining whether two adjacent image regions are part of the same surface requires understanding which elements of their physical explanations are coherent and which contain discontinuities. By identifying which elements are coherent, a segmentation algorithm can use various methods of image analysis to test for similarity. This theoretical framework is the basis for the segmentation algorithm presented in Chapter 3.

2.1. The Elements of a Scene

The elements constituting this model of a scene are surfaces, illumination, and the light transfer function or reflectance of a point in 3-D space. These elements have been defined as the *intrinsic characteristics* of a scene, as opposed to image features, or *extrinsic characteristics* such as edges or regions of constant color [55]. This section begins by providing a formal notation for each of the basic scene elements.

2.1.1 Surfaces

The model for objects in the real world uses 2-D manifolds, or *surfaces*. On a given surface, we can define local coordinates as a two-variable parameterization (u, v) relative to an arbitrary origin. The shape of the manifold in 3-D space is specified by a *surface embedding* function $S(u, v) \rightarrow (x, y, z)$, defined over an extent $E \subseteq (u, v)$. The surface embedding function maps a point

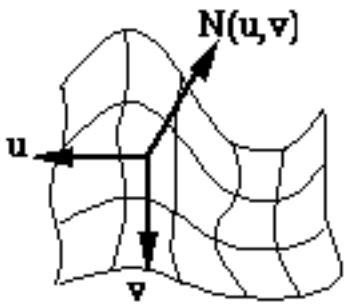


Figure 2.1: Local coordinate system on a surface patch.

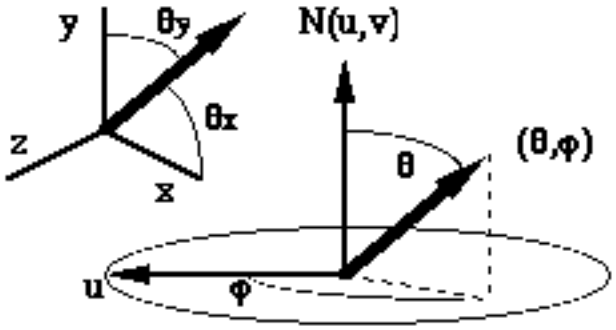


Figure 2.2: Specifying direction in global and local coordinates.

in the local coordinates of the manifold to a point in 3-D global coordinates. This global coordinate system is also anchored to an arbitrary origin, often specified relative to an imaging device. As shown in Figure 2.1, the surface embedding permits us to define a tangent plane $T(u, v)$ and surface normal $N(u, v)$ at each point on the manifold. In terms of the surface embedding function, the surface normal, or gradient of the surface at a point (u,v) is given by (1).

$$N(u, v) = \left(\frac{\partial S(u, v)}{\partial x}, \frac{\partial S(u, v)}{\partial y}, \frac{\partial S(u, v)}{\partial z} \right) \tag{1}$$

The tangent plane at (u,v) is the plane passing through the point $S(u,v)$ and perpendicular to the gradient $N(u,v)$. Using the tangent plane and surface normal we can define a local 3-D coordinate system at each surface point. This coordinate systems has two axes on the tangent plane and one in the direction of the surface normal. We can also define other useful properties, such as curvature, for each point using the surface embedding function.

It is important to note that this approach does not view the world as consisting of surfaces to be found, but as objects to be modeled. It is commonly presumed in machine vision that “surfaces” exist in nature, and that the job of the vision system is to discover them. This approach rejects that view, believing instead that surfaces are artifacts of the interpretation process and exist only within the perceptual system that is attempting to build a model of the world. In other words, there is no “correct” surface with which to model an object. Instead, the choice of manifold and surface embedding function is made by the modeler, and it depends largely upon the task and information at hand. Given a brick wall, for example, if the application is obstacle avoidance, a single plane could be chosen to model the entire wall. For other situations, such as segmentation, it might be necessary to model each brick as well as the troughs between them. At an even smaller scale, understanding the image texture in detail may require a model of each bump on each brick in order to interpret the wall. All are potentially useful “surfaces” to model the same wall, and all might be needed at various points in the visual process. Thus one object in the world can be modeled by many different surfaces, and the choice of model, or surface, is made by the interpreter. This view allows us to conceive of a perceptual process that incorporates numerous differing surfaces to describe an object, an important capability that other computational vision systems, which seek for a single “correct” surface, lack.

In order to parameterize light striking and reflecting from a surface, we also need to define a parameterization of direction. Global coordinate system directions are indicated by the ordered

pair (θ_x, θ_y), where θ_x specifies the angle between the direction vector and the x-axis, and θ_y specifies the angle between the direction vector and the y-axis as shown in the inset diagram in Figure 2.2. Local coordinate system directions are given by normal spherical coordinates specified by the ordered pair (θ, φ). θ is the *polar angle*, defined as the angle between the surface normal and the direction, and φ is the *azimuth*, defined as the angle between a perpendicular projection of the ray onto the tangent plane and a reference line on the surface. The reference line is usually defined as either the *u* or *v* axis as shown in the central diagram of Figure 2.2.

2.1.2 Illumination

Much research in machine vision assumes a single light source, often a relatively large distance away from the scene being imaged. More recently, Langer and Zucker proposed a computational illumination model that incorporates many forms of direct illumination [28]. However, numerous visual phenomena arise because of reflection from nearby objects acting as additional light sources. Computer graphics has long incorporated this idea into systems such as ray tracing and radiosity. In the field of machine vision, however, while interreflection has been studied between two objects, no general model exists for specifying the totality of illumination on a surface point.

To understand general images we cannot assume point lighting, three independent light sources, or other constructed illumination setup. A general model must allow us to specify any type of illumination, including interreflection from other objects, and still have identifiable subsets that fit with our traditional conceptions of illumination. The approach taken here begins by specifying and defining the parameters of a single ray of light and then extending this to a parametric model describing the totality of light arriving at a point.

A *photon* is a quantum of light energy that moves in a single direction unless something--like matter, or a strong gravity field--affects its motion. Thanks to the sun and artificial light sources, there are many photons moving in many directions at any given time. Collections of photons moving in the same direction at the same place and time constitute *rays* of light. As photons move, they oscillate about their direction of travel at a spectrum of *wavelengths* λ which specify the distance traveled in a single oscillation. The human eye is sensitive to photons with wavelengths that fall between approximately 380 and 760nm, and the spectral distribution of wavelengths present in a collection of photons determine what color we see. A charge-coupled device [CCD] camera responds to a slightly different range of wavelengths, and infrared and color filters are normally used to approximately match the color response of the human eye. The *polarization* of a population of photons specifies their oscillation and orientation with respect to the direction of travel, and it can affect the manner of reflection and transmission when light interacts with matter. Polarization is commonly represented using the Stokes parameters [8]. The Stokes parameters are four quantities which are functions of observable attributes of the electromagnetic wave. As an operational definition, consider four filters that, under natural illumination, will transmit half of the incident light. If the first filter is isotropic, passing all states equally, the second and third filters are linear polarizers whose transmission axes are horizontal and at +45 , and the fourth filter is a circular polarizer, then the four Stokes parameters are functions of the transmitted irradiances of these filters. Note that this selection of filters is not unique. We represent the Stokes parameters by the variable $s \in \{1, 2, 3, 4\}$ that indexes them to specify the relative energy of photons oscillating at different orientations.

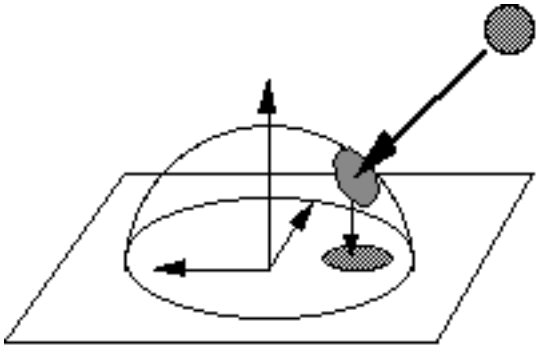


Figure 2.3: Orthogonal mapping of the illumination environment onto a plane.

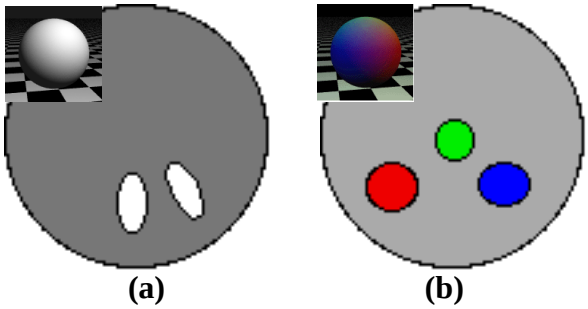


Figure 2.4: Visualizations of a) white uniform illumination, and b) general function illumination. The inset images show the appearance of a white sphere under the given illumination environment.

In a scene, light is being emitted or reflected in numerous directions, entering and leaving points throughout the area of interest. Using the parameters described above, we can specify a single ray of light at time *t* at position (x, y, z), moving in direction (θ_x, θ_y), of frequency λ and polarization *s* by the 8-tuple $(x, y, z, \theta_x, \theta_y, \lambda, s, t)$.

For the purposes of image formation, we want to specify the intensity of visible light that is incident from all directions on points (x, y, z) in global 3-D coordinates. In terms of the previously defined parameters, the *incident light energy field* function $L^+(x, y, z, \theta_x, \theta_y, \lambda, s, t)$ specifies the light energy arriving at the point (*x*,*y*,*z*) from direction (θ_x, θ_y) of wavelength λ and Stokes parameter *s* at time *t*. This function is similar to the *plenoptic function* defined in [1], or the *helios function* [39]. The presentation considers only single pictures taken at time *t*, making time a constant and removing it from the parameterization of illumination functions. As a result, we only need to consider the subspace of the incident light energy field $L^+(x, y, z, \theta_x, \theta_y, \lambda, s)$.

For a point in free space, rays arriving at that point can be mapped onto a sphere of unit radius [10]. This provides a visualization of the incident light on a surface point. The brightness and color of a point (θ_x, θ_y) on the sphere indicates the brightness and color of the incident light from that direction. This representation of the light energy field on the unit sphere for a 3-D point (x, y, z) is defined as the *global illumination environment* for that point. It is important to note that on opaque surfaces some of the incident light is blocked by the object matter itself, limiting the illumination environment to the hemisphere above the tangent plane. If the surface is transparent, the illumination environment will be the complete sphere, as light can arrive at the surface point from below as well as above. We can visualize the illumination environment for opaque surfaces by orthogonally projecting it onto a plane as in Figure 2.3. Two example illumination environments are shown in Figure 2.4. A rendering of what such illumination environments might look like is shown in the inset image beside each figure.

If we substitute the local surface coordinates (u, v) for the global coordinates (*x*,*y*,*z*), and the local spherical coordinates (θ, φ) for the global axis angles, we obtain the *local incident light energy field* $L^+(u, v, \theta, \varphi, \lambda, s)$, which also can be visualized on a hemisphere above the tangent plane to the local surface point for opaque surfaces. This representation is defined as the *local illumination*

environment for the surface point (u, v). The global and local illumination functions are distinguished by their parameters.

The total radiance of a patch of the illumination environment hemisphere with polarization specification s at wavelength λ, specified by the angles (θ, φ) and subtending $d\theta$ and $d\varphi$ is given by $L^+(u, v, \theta, \varphi, s, \lambda)\sin\theta d\theta d\varphi d\lambda$ [17]. The total irradiance at a point (u, v) is given by (2). The sine term is part of the solid angle specification, and the cosine term reflects the foreshortening effect from the perspective of the surface point. The inner integral is taken over θ.

$$E = \sum_s \int_\lambda \int_{-\pi}^{\pi} \int_0^{\frac{\pi}{2}} L^+(u, v, \theta, \varphi, s, \lambda)\cos\theta\sin\theta d\theta d\varphi d\lambda \tag{2}$$

2.1.3 Reflectance and the Light Transfer Function

In order for a point on a surface to be visible to an imaging system, there must be some emission of light from that point. As with the incident light energy field, we are interested in describing the light energy that is leaving a surface point (x, y, z) in every direction (θ_x, θ_y) in polarization state *s* for every wavelength λ. The light leaving a point is specified by the *exitant light energy field* $L(x, y, z, \theta_x, \theta_y, s, \lambda)$. This function has the same parameterization as the incident light energy field, and describes an intensity for every direction and wavelength. As with the incident light energy field, we can define a local coordinate version of the exitant light energy field $L^-(u, v, \theta, \varphi, s, \lambda)$.

The relationship between the incident and exitant light energy fields depends upon the macroscopic, microscopic, and atomic characteristics of the given point the light strikes. It is the gross characteristics of this relationship that allow us to identify and describe surfaces in a scene. Formally, the incident and exitant light energy fields are related by the reflectance, or *global light transfer function* $\Re(x, y, z; \theta_x^+, \theta_y^+, s^+, \lambda^+; \theta^-_x, \theta_y^-, s^-, \lambda^-; t)$ which indicates the exitant light energy field $L(x, y, z, \theta_x, \theta_y, s, \lambda)$ produced by one unit of incident light from direction $\left(\theta_x^+, \theta_y^+\right)$, of polarization s^+, and wavelength λ^+ for a particular surface point (x, y, z) at time *t*. The presentation assumes surfaces whose transfer functions do not change, making time a constant and allowing it to be dropped from the parameterization. Substituting the local coordinates (u, v, θ, φ) for the global parameters $(x, y, z, \theta_x, \theta_y)$ gives the *local light transfer function* $\Re(u, v; \theta^+, \varphi^+, s^+, \lambda^+; \theta^-, \varphi^-, s^-, \lambda^-)$.

The incident light energy, exitant light energy, and transfer function are related by the integral given in (3), which is written in terms of the local parameters. This integral says that the exitant light energy field is the sum of the self-luminance of the point, L_0, and the product of the transfer function and the incident light energy field integrated over the parameters of the incident light. The cosine term is due to foreshortening, and the sine term from the solid angle specification. The result of this integral is a function of the exitant light variables.

$$L^-(u, v, \theta, \varphi, s, \lambda) = L_0^-(u, v; \ldots - \ldots) + \sum_{s^+} \int_{\lambda^+} \int_{-\pi}^{\pi} \int_0^{\pi} L^+(u, v, \ldots + \ldots)\Re(u, v; \ldots + \ldots; \ldots - \ldots)\cos\theta^+\sin\theta^+ d\theta^+ d\varphi^+ d\lambda^+ \tag{3}$$

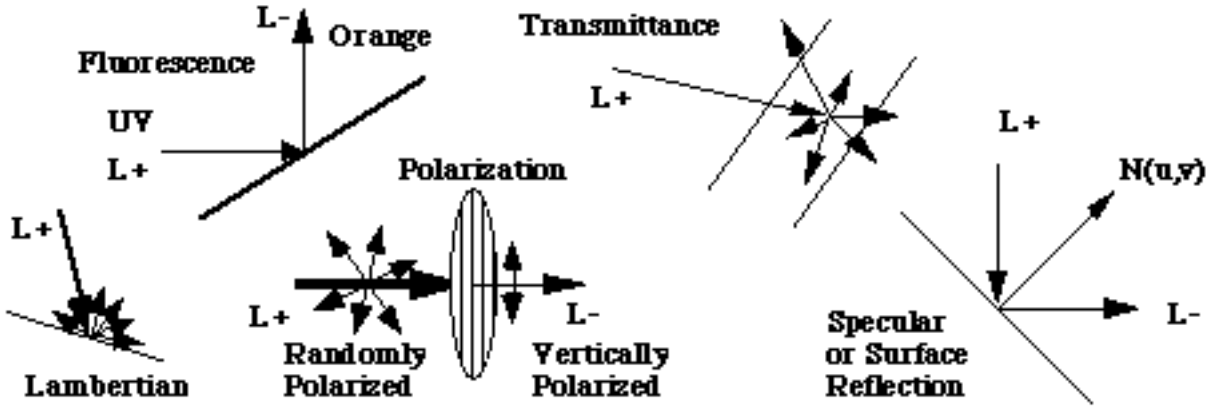


Figure 2.5: Some special cases of the light transfer function: Lambertian, fluorescence, polarization, transmittance, and specular or surface reflection

We can now undertake a structured analysis of the transfer function to show how it subsumes several common special cases such as: fluorescence, polarization, transmittance, and surface or specular reflection. These special cases are sketched in Figure 2.5. This analysis demonstrates the framework provided by the general transfer function.

For a non-fluorescing surface, if the incident light is of wavelength λ_0, then the exitant light energy field will also have wavelength λ_0, and no other wavelengths will be present. If, on the other hand, the same incident light strikes a fluorescent surface, there may be other wavelengths present in the exitant light energy field. In terms of the parameters of the transfer function, fluorescence implies there exists some pair of wavelengths (λ^+,λ^-) where $\lambda^- \neq \lambda^+$ for which $\Re > 0$.

Polarizing transfer functions modify the polarization of the incoming light. This effect can be seen in sunglasses, which often block the horizontal polarization mode. For non-polarizing surfaces, $\Re = 0$ whenever $s^+ \neq s^-$. For a polarizing transfer function, there exists some pair of Stokes parameters (s^+,s^-) where $s^- \neq s^+$ for which $\Re > 0$.

Transmitting surfaces allow some light to pass through them. Conversely, an opaque surface limits both the incident and exitant light energy fields to a hemisphere above the tangent plane for that surface. Transmittance occurs when either the exitant or incident light energy field bounds $\left(\theta^-, \varphi^-\right)$ and $\left(\theta^+, \varphi^+\right)$ are extended beyond the hemisphere above the tangent plane of the surface, implying that at least some of the exitant or incident light energy is passing through the material In terms of the parameters, a surface is transmitting if $\Re > 0$ when $\theta^- > 90°$ or $\Re > 0$ when $\theta^+ > 90°$.

Specular reflection, described in more detail later on, occurs when the incident light is only reflected about the local surface normal in the perfect specular direction. This restriction implies that the transfer function is zero except when $\varphi^- = \varphi^+ + \pi$ and $\theta^- = \theta^+$. It is important to note that surface reflection is relative to the local surface normal, and it is possible to have an optically rough surface where the local surface normals vary relative to the overall surface [3][57].

Finally, Lambertian surfaces--also called perfectly diffusing surfaces--reflect incident light

(a) **(b)**

Figure 2.6: Visualizations of a) grey metal, and b) light blue plastic (dielectric).

equally in all directions. For a unit energy ray of light from direction (θ, φ), the exitant light energy in all directions is specified by the expression $\rho \cos\theta$, where ρ is the albedo, or fraction of the incident light energy reflected by the material.

Figure 2.6 shows a method of illustrating specific transfer functions. A sphere with the specific transfer function properties sits above a matte black and white checkered surface under a dark grey sky with a white point light source shining on it from above and to the right of the viewer. Because all illumination is of uniform spectrum (i.e. grey), any color in the image is due to the transfer function. The checkerboard pattern is present to highlight the specularity of the object. Figure 2.6(a) is an illustration of a grey metal transfer function with no body reflection, and Figure 2.6(b) shows a matte blue dielectric transfer function with a small amount of surface reflection.

2.2. General Hypotheses of Physical Appearance

We have defined a 3-D world model for individual points and their optical properties, but how does a whole surface appear in a digitized computer image? To describe a surface and its appearance, we need a nomenclature for the aggregation of appearance properties in the 3-D world and how these aggregations map to an image.

We have defined surfaces with an extent and embedding and a transfer function $\Re$ over a surface. The combination of a surface and a transfer function we call a *surface patch*. Because the transfer function can vary arbitrarily, there are no constraints on the appearance of a general surface patch in an image. Frequently, however, the transfer function at nearby points on a surface displays some type of identifiable coherence. Coherence does not imply uniformity, and covers a broad scope of possible aggregations such as uniformity, repetitive patterns, or irregular textures. Some properties that commonly impart coherence include material type, color, roughness, and the index of refraction. We can model an object's appearance with a surface patch whose transfer function is similarly coherent.

A surface patch with a coherent transfer function, however, will not always display the coherence in an image. Differing illumination over the surface patch or occluding objects can mask or modify the appearance of the patch to an imaging system. For the purposes of image analysis, we would like to specify not only coherence in the transfer function, but coherence in the exitant light energy field, which is what is viewed by the imaging device. To achieve coherence in the exitant light energy field, we must add to the surface/transfer function pair a coherent illumination envi-

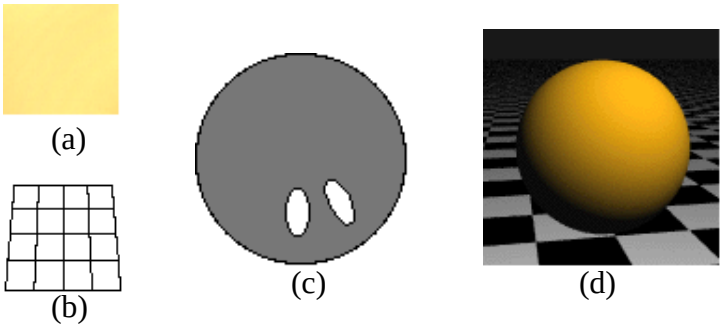


Figure 2.7: Hypothesis visualization: a) the actual region, b) wire-diagram of the shape, c) illumination environment, and d) transfer function.

ronment over the surface patch. This combination is an *appearance patch:* a surface patch whose points exhibit a coherent transfer function and illumination environment, and whose exitant light energy field exhibits a coherence related to that of the transfer function over the entire patch, and which is not occluded from the imaging system.

Given an appearance patch, we can imagine that the exitant light energy field over the patch maps to a set of pixels in the image. The exitant light from a surface caught by the imaging device determines the color and position of the set of pixels related to that surface. The physical explanation for a given exitant light energy field from a given surface patch is a *hypothesis* $H = \langle S, E, \Re, L^+ \rangle$. The four elements of a hypothesis are the surface embedding S, the surface extent E, the transfer function $\Re$, and the incident light energy field L^+. With these functions, it is possible to completely determine the exitant light energy field, assuming no self-luminance.

To illustrate a hypothesis, we combine the representations previously developed into a 3-panel display of the characteristics of *S, L,* and $\Re$ as shown for a yellow region in Figure 2.7.

The basic connection between a physical explanation and a group of image pixels is provided by a *hypothesis region* $HR = \langle P, H \rangle$, defined as a set of pixels P that are the image of the hypothesis H. The combination of the hypothesis elements represents an explanation for the color and brightness of every pixel in the image region. For simplicity, we assume the image is formed by a pinhole camera at the origin looking at the canonical view volume. To represent the fact that a single region may have more than one possible explanation, we define a *hypothesis list* $HS = \langle P, H_1, \ldots, H_n \rangle$ to be a set of pixels *P* with an associated list of hypotheses $H_1, \ldots, H_n$, where each hypothesis H_i provides a unique explanation for all of the pixels in *P,* and only the pixels in *P.*

Finally, let P_i be a set of non-overlapping pixel regions in an image, and let $HS_i = \langle P_i, H_{i1}, \ldots, H_{in} \rangle$ be the corresponding hypothesis sets. We define a segmentation of $P = \bigcup_i P_i$ as a set of hypotheses HS_i containing one hypothesis for each region in P_i. Of course, to be physically realizable, these hypotheses must be mutually consistent. The goal of low-level vision, in terms of this vocabulary, is to produce one or more segmentations of the entire image.

In summary, the general model for a scene consists of three elements: surfaces, illumination, and

the light transfer function or reflectance of a point or surface in 3-D space. These elements constitute the *intrinsic characteristics* of a scene, as opposed to *image features* such as pixel values, edges, or flow fields [55]. The combination of models for these three elements is a *hypothesis* of image formation. By attaching a hypothesis to an image region we get a *hypothesis region*: a set of pixels and the physical process which gave rise to them. When an image region has multiple hypotheses, we call the combination of the image region and the set of hypotheses a *hypothesis list*.

Note that a hypothesis list can also be thought of as an equivalence class: a hypothesis list is a set of physical explanations that generate the same set of pixels in an image [cite some manipulation people]. A hypothesis list is complete if it contains all possible physical explanations for a specific set of pixels given a scene model. In this nomenclature, physics-based segmentation methods based on restricted scene models use limited hypothesis lists containing one, or at most two hypotheses. While this simplifies the task of obtaining a final segmentation, or choosing one hypothesis from each hypothesis list, it limits the range of physical scenes they can analyze.

It is important to realize that without prior knowledge of image content, no matter how an image is divided there are numerous possible and plausible hypotheses for each region. The color of an image region may vary because of changes in the illumination, the transfer function, or both. Likewise, intensity varies because of changes in the shape, illumination, transfer function, or any combination of the three. Many algorithms, in particular shape-from-shading, work because they assume the image variation is due to changes in only one element of the hypothesis [9].

With a general scene model and the hypothesis framework we can begin to relax these assumptions. The task ahead is to enumerate the set of hypotheses contained in each hypothesis list, or the set of physical explanations in each equivalence class. We want to base these hypotheses on the general scene model previously developed. Furthermore, we would like for the hypothesis list to be as complete as possible, without having to specifically enumerate the infinite set of possible physical explanations. Somehow, we must identify the qualitatively different physical explanations and generate classes within the general parametric models.

2.3. Taxonomy of the Scene Model

In addition to enumerating a complete set of hypotheses for an image region, we want to focus our attention on the more likely physical explanations. In an ideal world “likelihood” would be quantifiable and could be used directly as the basis for generating and rank-ordering the possible hypotheses for a given region. The weirdness, or improbability of a hypothesis might be represented by three axes indicating the complexity of the shape, transfer function, and illumination environment. More probable explanations would be those closer to the origin of the three axes. The further from the origin, the more improbable the hypothesis elements would become. By generating hypotheses close to the origin, or with only one unlikely element, we could begin with a small set of simple, probable hypotheses and generate more improbable ones only if necessary. Unfortunately, this is a difficult concept to measure directly and the separate axes would almost certainly be non-linear and not independent.

It is possible to quantify complexity, however, using a criterion such as the minimum description length [MDL] principle [51]. While this is not equivalent to our concept of weirdness or improbability (a complex description is not necessarily improbable), the two are often correlated in the

world. The MDL principle states that, given a parameterization for describing a family of models, the best model for describing a set of data is the one that best satisfies two constraints: 1) it is a good model of the data (best fit), and 2) it can be expressed in the fewest number of binary digits, or shortest length. The MDL principle has been used successfully in several computer vision tasks (e.g., Leclerc [31], Darrell *et al*. [11], Krumm [27] and Leonardis [35]). Our goal is to discover a set of hypotheses that both accurately describe the data set and are simple to represent. Therefore, if we use the MDL principle as a guide, given a set of hypothesis lists each of whose hypotheses models its respective image region equally well, the best segmentation of an image is the least complex one.

It is important to note that the description length principle has two components: the complexity of the description, and how well that description fits the data. A combination of the two components is used to select the best model. When we are dealing with a set of plausible hypotheses for an image region the individual hypotheses ought to fit the data equally well. This implies that the term indicating the goodness of fit is approximately constant for all plausible hypotheses. Therefore, rank-ordering the hypotheses for a region using only a measure of complexity should be sufficient to satisfy the MDL criteria.

Note, however, that the complexity of a hypothesis is dependent upon the image data and all of the elements of the hypotheses. For example, in isolation a planar surface is simpler to represent than a curved surface. However, as part of a hypothesis that explains an image region showing smooth variation in intensity, the planar hypothesis may be more complex overall because it requires a more complex illumination environment or transfer function to produce the same appearance as the hypothesis proposing a curved surface. The important lesson is that hypotheses cannot be reliably rank-ordered in terms of complexity without considering the image region they represent.

As there are a large number of hypotheses for any image region, how to select the initial hypothesis set for each region is a crucial decision. One important consideration of the MDL principle is that the optimal model, or model set must be among those tested for shortest length. Three possible approaches that could be taken to generate this model set are:

1. Generate a large number of possible hypotheses and test them
2. Generate incrementally according to some search criterion
3. Generate a small, but comprehensive set, using broad classes of the hypothesis elements; expand this set incrementally if all of its constituents are ruled out as possibilities

As indicated by previous discussion, the first approach seems pointless and intractable. Breton *et al.* were able to use this approach and create a discrete mesh of possible light source directions for a “virtual” point source [5]. Because our model has many more parameters in both the illumination environment and the transfer function, however, such coverage by a discrete mesh is intractable. The second approach has merit, but a search algorithm faces some difficult challenges. First, the space of hypotheses is continuous and achieving sufficient resolution may be computationally intractable. Second, it is unclear what criteria would drive the search. For example, consider developing a reliable estimate of the distance to the goal--as required by a search algorithm such as A*--when the exact relationship between the parameters is unknown. Third, the problem-space is ill-conditioned as small changes in some parameters can require large changes in others in order

to generate the same exitant light energy field. As an example consider changing the position of a light source by a small amount over a wavy surface. In order to generate the same exitant light energy field, the transfer function of the surface would have to change dramatically.

Instead, through careful analysis we propose dividing the space of possible models into broad classes. Given that we are looking for simple hypotheses, it makes sense to identify subspaces of the general parameterizations which are both simple and likely to occur in everyday images. We can use these broad classes to assign an initial hypothesis set to each image region, instantiating the details of a particular hypothesis--i.e., finding the actual shape, the specific colors, surface roughness, and other characteristics--as they are available and needed in the segmentation process. This method abstracts the problem to a simpler domain and allows us to use the results of the analysis as a guide through the higher dimensional problem space.

The next three subsections derive broad classes from the general parameterized models. These classes are simple, yet comprehensive enough to cover a wide range of possible environments and objects. Furthermore, while they are abstractions of more detailed models, they contain sufficient information to facilitate reasoning about different physical interpretations and the relationships of these interpretations between neighboring regions. In this way they balance the competing demands for having a complete list, focusing on simplicity, and keeping the number of hypotheses to a reasonable level.

2.3.1 Taxonomy of Surfaces

Surfaces have numerous levels of complexity. A cube, for example, can be modeled as a set of planar patches, a polyhedron, or a superquadric. As noted previously, when modeling objects in the real world, surfaces may take on any amount of complexity, depending upon the needs of the modeler. To reason about merging adjacent hypotheses, we need to know whether they have compatible shapes--i.e. fit together at the boundaries. When the boundaries are compatible, we should consider merging the two regions.

This analysis initially considers only one characteristic of a surface: is it curved, or is it planar? Clearly a curved surface can be arbitrarily close to planar, so in practical application this distinction must be made using a selected threshold. The curved/planar distinction allows for straightforward reasoning at an abstract level about merging two hypothesis regions. A finer distinction requires a specific method for modeling curved surfaces. When a surface representation method is chosen, reasoning about merging two curved surfaces can be done based on that representation--e.g. matching two spheres, superquadrics, generalized cylinders, or polynomial surfaces. Note that absolute depth is not a necessary requirement for reasoning about merging. If two regions' boundaries do not match in relative shape, they should not be merged. If the regions' boundaries do match, a merger is not ruled out on the basis of shape.

2.3.2 Taxonomy of Illumination

Several special forms of the illumination function are often used in both computer vision and computer graphics to represent light conditions in a scene. The general form of L^+, given by $L^+(u, v, \theta_x, \theta_y, \lambda, s, t)$, contains these special cases as subspaces of its parameters space. Figure 2.8 identifies several subspaces for this function and shows their relationships. Not shown in Figure 2.8 is the all-encompassing set of time-varying illumination functions. As noted previously, we

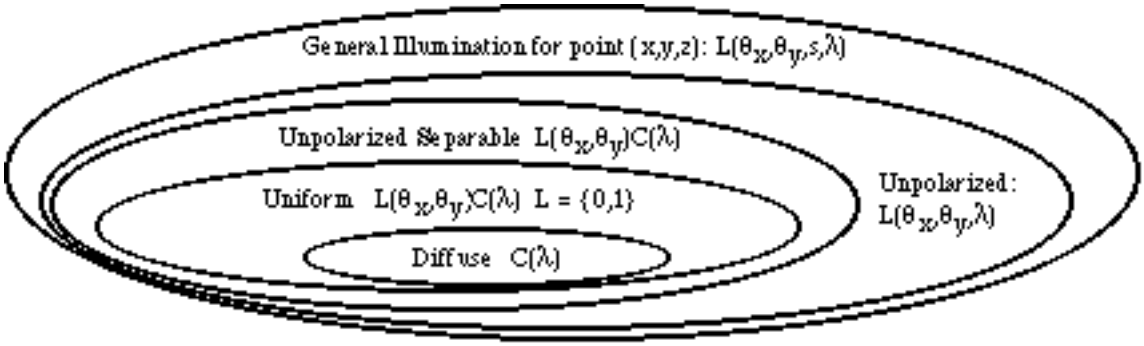


Figure 2.8: Subspaces of the global incident light energy field L(x,y,z,θ_x,θ_y,λ,s).

assume time-invariant illumination, making time a constant and allowing us to remove it from the parameterization. This leaves us with time-invariant illumination functions, shown as the largest space in Figure 2.8. Within the space of general time-invariant illumination functions is the subspace of unpolarized time-invariant illumination $L^+(u, v, \theta_x, \theta_y, \lambda)$. For most images of interest all of the illumination in a scene is characterizable by this function. Scenes with illumination outside this subspace are rare, and would be those illuminated by a polarized light source such as a laser, or by a time-varying light source with significant variation over the course of the image capture process.

One common assumption in computer vision is that the illumination over the hemisphere is constant in its hue and saturation, but of varying brightness. Mathematically, this subspace is represented by the *separable* illumination functions. Separable illumination functions are those which can be expressed as $L^+(x, y, z, \theta_x, \theta_y)C(x, y, z, \lambda)$, where $L^+(x, y, z, \theta_x, \theta_y)$ specifies the incoming intensity in a given direction at (*x,y,z*), and $C(x, y, z, \lambda)$ the color of the illumination. A more restrictive subspace of separable illumination is the *uniform* illumination subspace which is defined for the point (x,y,z) as $L^+(\theta_x, \theta_y)C(\lambda)$, where $L^+(\theta_x, \theta_y) = \{1, \alpha\}$. Note that α represents the background, or ambient illumination commonly used in computer graphics. This definition states that each direction in a uniform illumination environment has the same color and one of two brightness values: light or dark. Some commonly used special cases of uniform illumination include:

1. Point light source at $(\theta_{x0}, \theta_{y0})$: $L^+(\theta_x, \theta_y) = \begin{cases} 1 & (\theta_x = \theta_{x0}) \text{ and } (\theta_y = \theta_{y0}) \\ 0 & \text{otherwise} \end{cases}$
2. Finite disk source subtending a cone of apex angle α centered at $(\theta_{x0}, \theta_{y0})$:
$$L^+(\theta_x, \theta_y) = \begin{cases} 1 & \text{angle between } (\theta_x, \theta_y) \text{ and } (\theta_{x0}, \theta_{y0}) < \alpha \\ 0 & \text{otherwise} \end{cases}$$
3. Perfectly diffuse illumination: $L^+(\theta_x, \theta_y) = 1$ for all θ_x and θ_y.

In the last case, L^+ is trivial and the illumination is fully characterized by $C(\lambda)$.

As shown by the computer graphics community, these three simple cases play an important role in

modeling illumination; a large number of illumination environments can be modeled using one or more point, finite disk, or ambient light sources [13]. The uniform illumination class captures all three cases.

In the field of computer vision, the illumination model of Langer and Zucker, a source and aperture model of illumination, is similar to the uniform illumination class. They have shown their model to be useful for analysis of scenes illuminated by diffuse or direct illumination and combinations of these such as a skylight on a cloudy day [28]. Both our model and theirs focus on the illumination as seen by different points in the scene, rather than on the actual composition of the light source itself.

When reasoning about hypotheses, we would like to have a small number of classes, with most of them being highly constrained. The three subspaces--diffuse, uniform, and general illumination--are all useful in that two of these cases are highly constrained, and comprehensive as they provide good coverage of common illumination environments. Diffuse illumination approximates objects in shadow or not directly lit. Uniform illumination approximates man-made and natural light sources, and we must include general illumination because in some situations it is necessary--such as the colored objects reflected by the teakettle in Figure 1.1. Figure 2.4 illustrates both a uniform illumination environment and a general illumination environment along with their effects on white dielectric spheres.

2.3.3 Taxonomy of the Transfer Function

As with the illumination function, the transfer function can be subdivided into commonly occurring subspaces. These generally fall within the space of non-polarizing, opaque, and non-fluorescing transfer functions. We assume that the transfer functions of all objects within a scene are represented within this subspace. This assumption implies three constraints:

- the polarization parameters are separable and, as we consider only unpolarized incident light, can be removed from the parameterization;
- λ^+ and λ^- can be combined into a single parameter λ since non-fluorescence implies that $\Re = 0$ whenever $\lambda^+ \neq \lambda^-$;
- the direction of incident and exitant light is limited to a hemisphere above the tangent plane for the point (u, v).

These assumptions allow us to rewrite the transfer function as $\Re(u, v, \theta^+, \varphi^+, \theta^-, \varphi^-, \lambda)$, where $0 < \theta < 90°$. They do not, however, restrict the nature of the transfer function between neighboring points. Transfer functions exhibiting coherence over the extent of (u, v) form subspaces of the more general function. Two restrictive, but common subspaces are transfer functions exhibiting piecewise-uniform and uniform characteristics over their extent. In the uniform surface subspace, the transfer function is constant with respect to the parameter pair (u, v) and can be rewritten as $\Re(\theta^+, \varphi^+, \theta^-, \varphi^-, \lambda)$. For this subspace the transfer function is identical to the well-known *spectral bi-directional reflectance distribution function* [spectral BRDF] for a uniform surface [42].

This analysis concentrates on the spectral BRDF. We model the BRDF as containing two important and overlapping elements: surface reflection, and body reflection. Their proposed relationship

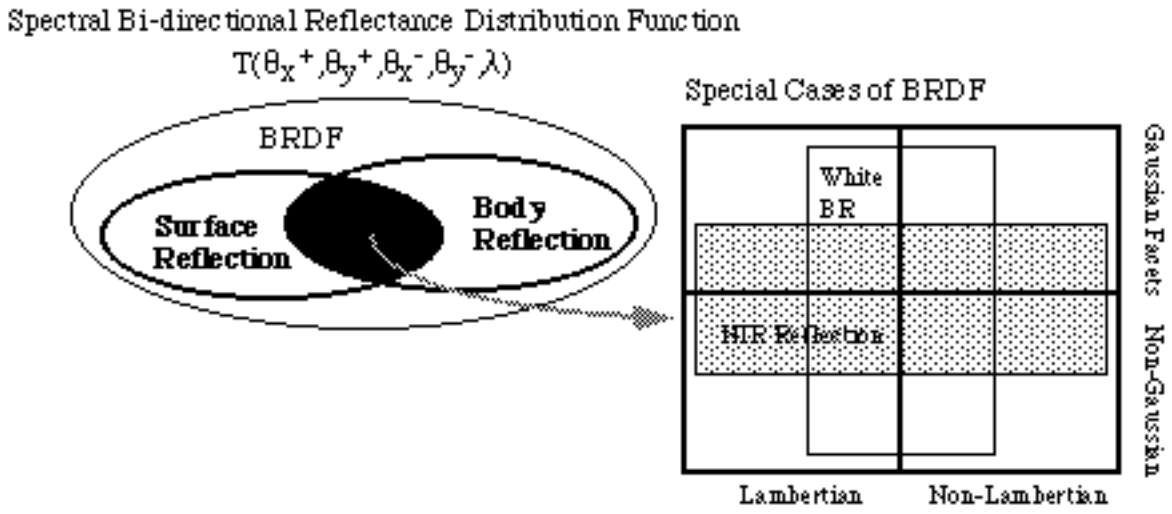


Figure 2.9: Taxonomy of the spectral bi-directional reflectance distribution function

within the BRDF and the interaction of the union of these subspaces is shown in Figure 2.9.

Surface reflection, as noted previously, takes place at the interface between an object and its surroundings. The direction of the exitant light energy is governed by the surface normal at the point of reflection; it is reflected through the local surface normal in the "perfect specular direction." The amount of light reflected is determined by Fresnel's laws, whose parameters include the angles of incidence and emittance, the index of refraction of the material, and the polarization of the incoming light. For white metals and most man-made dielectrics the surface reflection can be considered constant over the visible spectrum [20][22]. Materials whose surface reflection fits this assumption form a useful subset, shown in Figure 2.9, and are said to have *neutral interface reflection* (NIR) [34]. The surface reflection from an NIR material is approximately the same color as the illumination. Common materials for which the surface reflection is more dependent upon wavelength include "red metals" such as gold, copper, and bronze, all of which modify the color of the reflected surface illumination [16].

Many materials displaying surface reflection are optically "rough." They possess microscopic surfaces with local surface normals that differ from the macroscopic shape, as shown in Figure 2.10(a). A subset of these rough surfaces are those with roughness characteristics--such as microscopic slopes or heights--that have a Gaussian distribution. Several reflection models, such as Torrance-Sparrow and Beckmann-Spizzochino, have been developed for rough surfaces using a Gaussian distribution assumption for some surface characteristic [3][40][57]. These models fit into our taxonomy of transfer functions as shown in Figure 2.9.

Metals are an example of a material that displays only surface reflection. Because of the nature of the metal atoms, virtually no light penetrates beyond the surface of the material. Healey modeled their appearance with the *unichromatic reflection model*, and models for surface reflection from both smooth and rough surfaces apply directly to metals *[16]*.

A more complex form of reflection, body reflection, occurs when light enters a surface and strikes colorant particles as shown in Figure 2.10(b). The colorant particles absorb some of the wave-

**Figure 2.10: a) Microfacet surface reflection model,
b) body reflection model of transparent medium with pigment particles.**

lengths and re-emit others, coloring the reflection. The photons that are re-emitted go in random directions, striking other colorant particles, and ultimately exiting the surface as body reflection. Surfaces whose colorant particles re-emit equally all wavelengths of visible light form the "white" subset of transfer functions with body reflection.

Because of the stochastic nature of this reflection, a common assumption is that the body reflection is independent of viewing direction. Surfaces whose transfer functions display this independence are called Lambertian because they obey Lambert's Law, which states that the reflection is dependent upon the incoming light's intensity and cosine of the angle of incidence [17]. Other models of body reflection that are dependent upon viewing direction are being researched [34][60]. The white subset and Lambertian subset relationships are shown in Figure 2.9.

Many interesting and useful transfer functions exhibit both body and surface reflection. Common materials simultaneously displaying these types of reflection include plastic, paint, glass, ink, paper, cloth, and ceramic, most of which can be modeled with the NIR assumption. Transfer functions within this overlapping region have been approximated by the *dichromatic reflection model* [53] [56].

The objects of interest for this segmentation framework are those whose transfer functions fall within the union of body reflection and surface reflection. Objects with these properties naturally divide into two categories: metals and dielectrics. Metals, as noted previously display only surface reflection; dielectrics always display body reflection, and may also display a strong surface reflection component, depending upon the surface characteristics. Illustrations of these two classes of the transfer function are shown in Figure 2.6.

The bimodal appearance of dielectrics suggests a further subdivision of the dielectric category into dielectrics displaying body reflection, and dielectrics displaying both body reflection and strong surface reflection. While there will be points on a surface that display a balanced mixture of each, there is a clear distinguishing feature between points displaying only body reflection and points displaying both body reflection and strong surface reflection. Body reflection modifies the color of the incident light by the color of the object. Strong surface reflection on a dielectric surface with a neutral interface does not modify the incident light color. Therefore, these two situations have a qualitatively different appearance within the overall category of dielectrics. A clear example of this is a highlight on a plastic object.

2.4. Fundamental Hypotheses

The taxonomies developed for S, L^+, and $\Re$ allow us to identify sets of broad classes based upon partitions of the parameter space. In summary, the broad classes for each hypothesis element are:

- Surfaces = {planar, curved}
- Illumination Environment = {diffuse, uniform, general}
- Transfer Function = {metal, dielectric, dielectric displaying surface reflection}

There are eighteen possible combinations of these broad classes, subdividing the space of hypotheses for an image region into eighteen subspaces. Each of these subspaces is parameterized by the color values (wavelength spectrum) of the illumination and the transfer function.

2.4.1 Generating the Fundamental Hypotheses

Because of the large number of possible color distributions, to reason about hypotheses we subdivide L^+ and $\Re$ into two classes: uniform spectrum (white or grey), and non-uniform spectrum (colored). This divides L^+ into six forms of illumination, and $\Re$ into six forms of the transfer function. The possible combinations of surface, illumination, and transfer function are defined as the set of *fundamental hypotheses* for an image region.

We denote a specific fundamental hypothesis using the 3-tuple (<**transfer function**>, <**illumination**>, <**shape**>). The three elements of a hypotheses are defined as follows.

- <**transfer function**> := Colored dielectric | White dielectric | Colored dielectric displaying surface reflection | White dielectric displaying surface reflection | Col. metal | Grey metal
- <**illumination**> := Col. diffuse | White diffuse | Col. uniform | White uniform | Col. complex | White complex
- <**shape**> := Curved | Planar

Simple combination of the classes of the hypothesis elements (2 x 6 x 6) indicates there are 72 possible hypotheses. However, not all 72 are applicable to every region. Consider first a colored region. To possess color, either L^+ or $\Re$ must have a non-uniform spectrum. If we remove from consideration the 24 uniform illumination/uniform transfer function hypotheses, 48 fundamental hypotheses remain for a colored image region.

Conversely, the elements of the hypotheses for a grey or white image region must postulate no color. A situation where both the illumination and the transfer function are colored and yet their combination is grey is possible, but we assume this situation to be rare enough to neglect it for most images. This implies there are 24 fundamental hypotheses for a uniform spectrum region. Therefore, for a given image region we have to consider at most either 48 or 24 fundamental hypotheses.

To more explicitly show the structure of the fundamental hypotheses we arrange them as shown in Figure 2.11 and Figure 2.12. Note that each leaf represents both a planar and a curved hypothesis. The trees represent taxonomies of the fundamental, or simplest hypotheses and classify the different physical explanations for gray and colored image regions. The leaves of these trees are a finite set of simple, comprehensive explanations for the color and brightness of every pixel within an

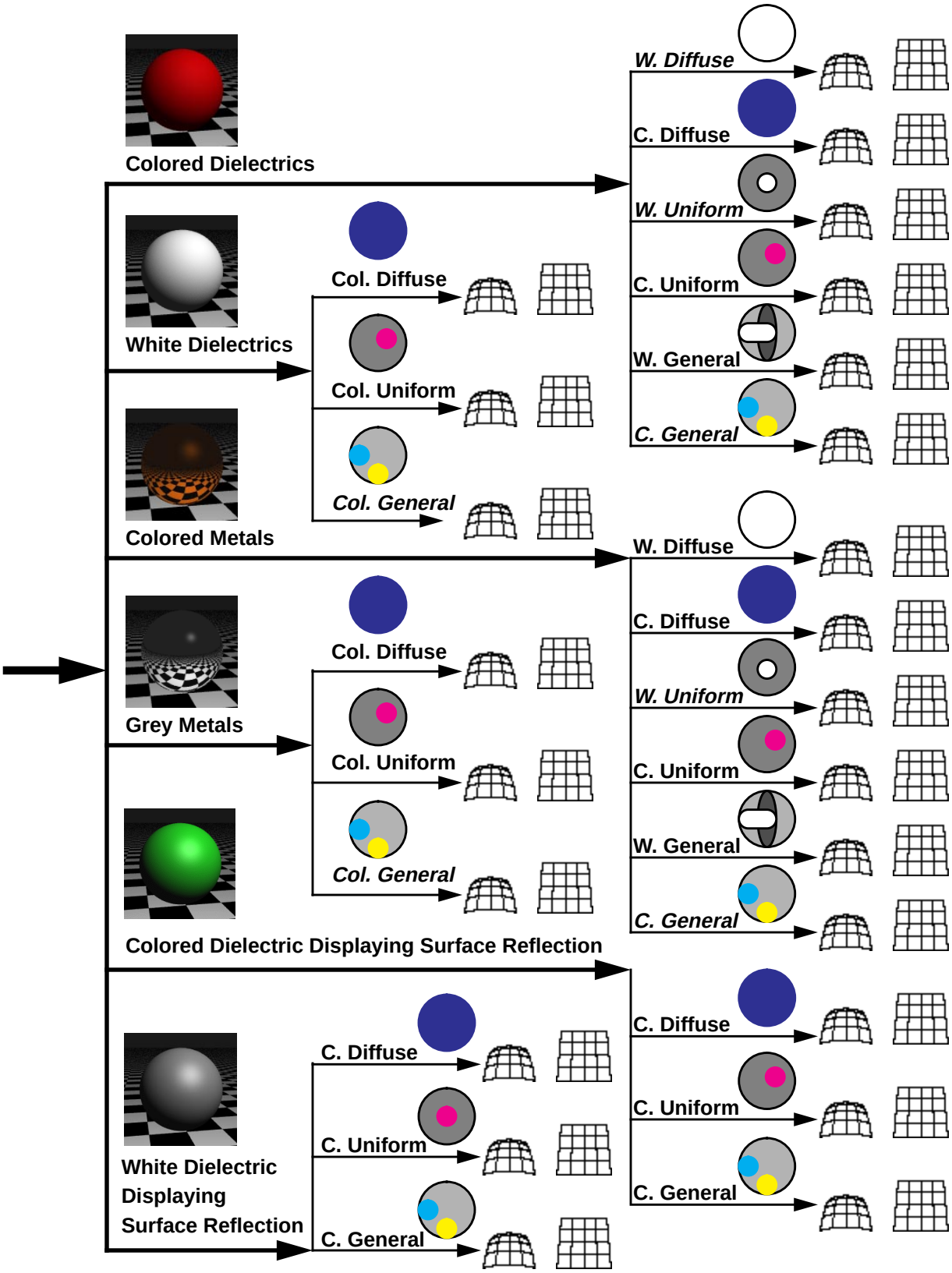


Figure 2.11: Taxonomy of hypotheses for a colored region. Each leaf represents both a planar and a curved hypothesis.

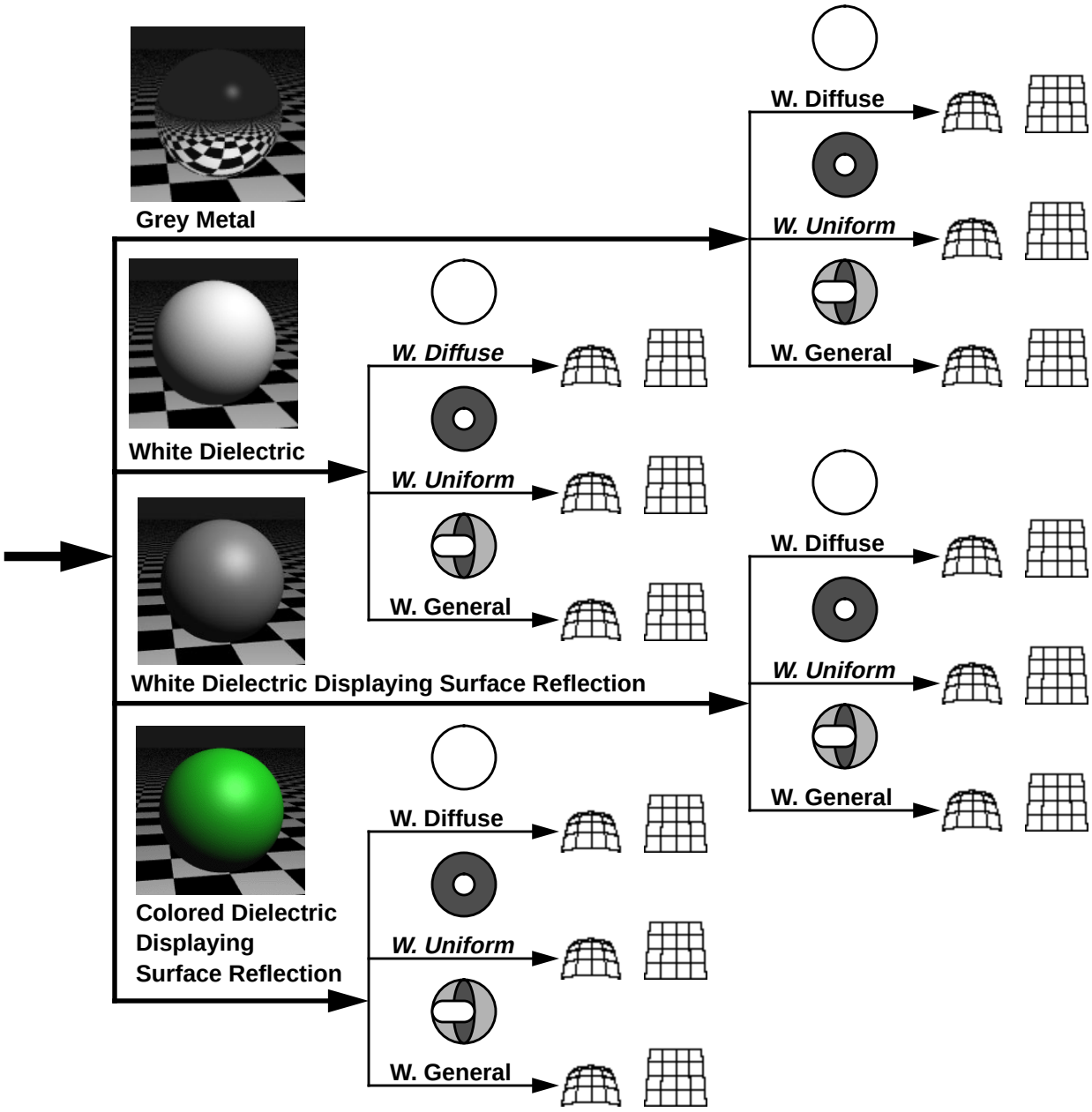


Figure 2.12: Taxonomy of a white or grey region. Note each leaf represents both a planar and a curved hypothesis.

image region. Using the set of fundamental hypotheses as the initial hypothesis list for each region, we can begin to reason about and merge hypothesis regions into more sensible global hypotheses that correspond more closely with what we consider to be objects in the scene that created the image.

2.4.2 Analyzing the Fundamental Hypotheses

The taxonomy of Figure 2.11 implies that all of the fundamental hypotheses possess equivalent value for describing regions of an everyday scene. We believe this is not the case for most images.

To concentrate our efforts on the more common hypotheses, we can subdivide the 72 hypotheses into two groups, or tiers, reflecting how common or rare a hypothesis seems to be. Common hypotheses we place in tier one and less common hypotheses in tier two.

We begin with a structured analysis of each subtree of the taxonomy for the hypotheses of a colored image region, considering in turn each of the four classes of material. We are guided in our analysis by two general rules which take into consideration the estimated size relationships of subspaces of the taxonomies.

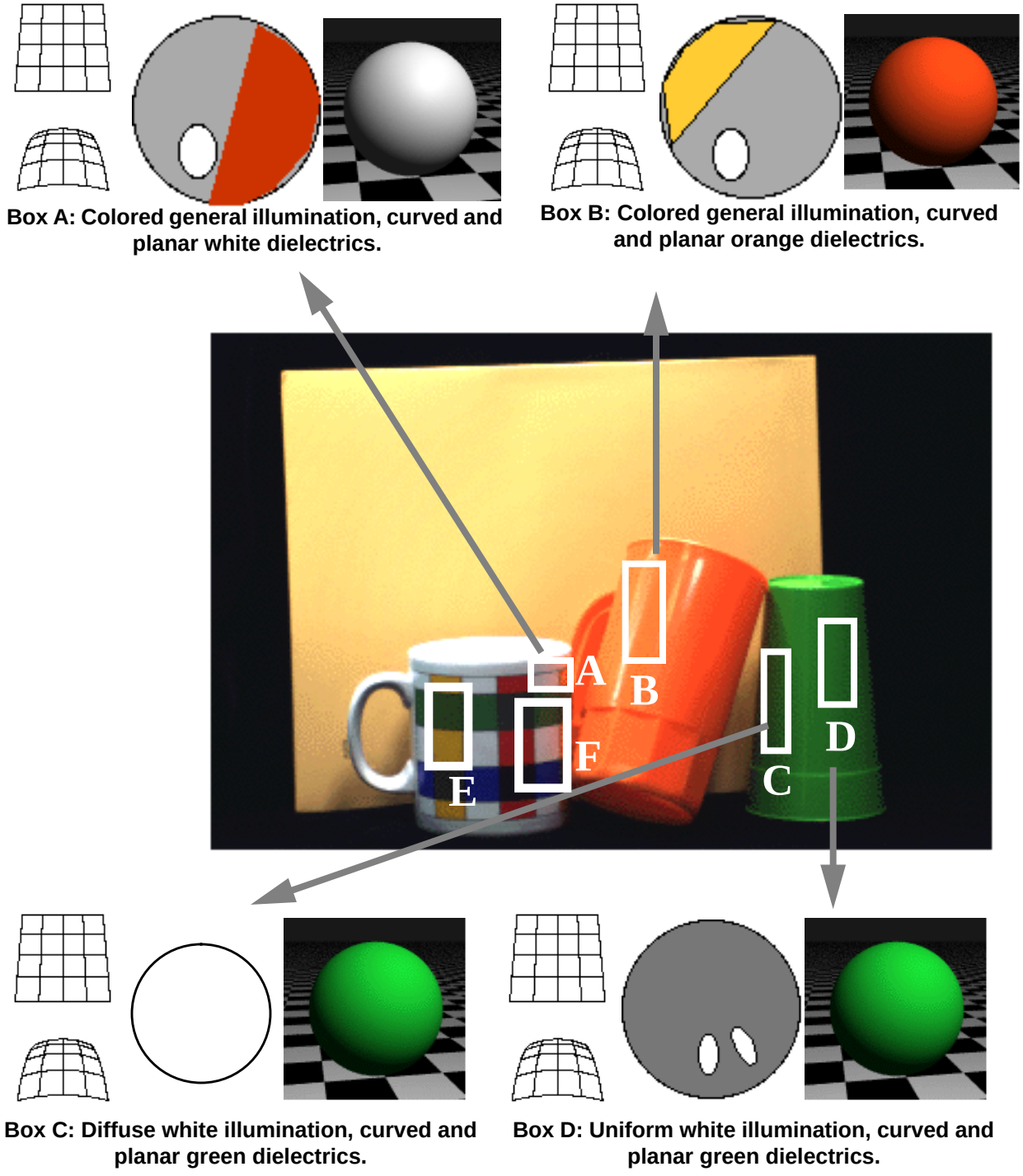
- If a subspace is both common and a good approximation of a larger encompassing space, place the subspace in tier one, and the larger space in tier two.
- If a subspace is uncommon or not a good approximation of a common larger space, place the subspace in tier two and the larger space in tier one.

We begin by looking at the hypotheses concerned with colored dielectrics. These twelve hypotheses are grouped into six pairs according to the illumination environment. The first two, curved and planar dielectrics under diffuse white lighting are often used as a model for surfaces in shadow where no light source is directly incident [13]. An example of this case appears within box D of Figure 2.13. Such situations are common in everyday pictures compared with colored diffuse illumination such as might exist in a darkroom. Therefore, we place curved and planar colored dielectrics under diffuse white illumination in tier one. Tier one hypotheses are italicized in both Figure 2.11 and Figure 2.12.

The next two hypotheses, curved and planar colored dielectrics under uniform white illumination, represent a significant subset of surfaces in a typical scene such as Figure 2.13. Boxes A and E are two examples. Sunlight can also be approximated by a uniform source when considering dielectrics because its effect on dielectric surfaces usually overwhelms any other illumination.

Curved and planar dielectrics under general function white illumination are an interesting pair of hypotheses. In the real world, they are probably some of the most common hypotheses, as uniform and diffuse lighting are only approximations. In the case of dielectrics, however, uniform and diffuse lighting models are probably sufficient for most situations. The reason is that dielectrics, unlike metals, have a strong body reflection component; they reflect some of the light from each incident direction in each exitant direction. In the extreme case, a perfectly Lambertian surface reflects the incident light from a single direction equally in all directions. The exitant light energy field caused by a single strong incident light source can overshadow any additional exitant light energy due to illumination from other directions. Therefore, in scenes where there are one or more white light sources of possibly varying intensity, we propose that the illumination can be adequately modeled as a set of uniform brightness white sources. Following the first rule noted above, this removes the need for white general illumination, allowing us to place it in tier two.

Curved and planar colored dielectrics under general colored illumination, however, are not well-modeled by any other hypotheses in tier one. In everyday scenes these hypotheses are needed to model interreflection such as occurs in boxes B and C in Figure 2.13. Because of this, we must place them in tier one. With colored general illumination in tier one, curved and planar colored dielectrics under colored diffuse illumination fall into tier two because they are both rare and subsumed by colored general illumination, of which they are not a good approximation. Likewise, we argue that planar and curved colored dielectrics under colored uniform illumination are both rare



Box A: Colored general illumination, curved and planar white dielectrics.

Box B: Colored general illumination, curved and planar orange dielectrics.

Box C: Diffuse white illumination, curved and planar green dielectrics.

Box D: Uniform white illumination, curved and planar green dielectrics.

Figure 2.13: Fundamental dielectric hypotheses for a colored region.

and subsumed by the colored general case. A darkroom is one example where there would be a colored light source and all diffuse illumination would have the same chromaticity, but uncontrived examples are difficult to find.

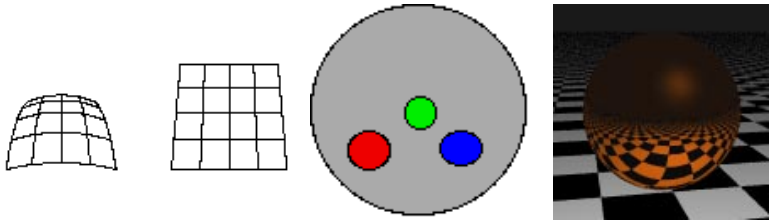
The next major branch corresponds to the six hypotheses for white dielectrics under colored illumination. In common scenes we suggest that situations corresponding to these hypotheses are rare. Probably the most common occurrence of these is interreflection between a colored object and a white dielectric object such as a white wall. In these cases, the white object is lit by both a direct light source and some type of colored reflection from a nearby object. Box A in Figure 2.13, for example, shows a white portion of the mug that is lit by both the direct white illumination and interreflection from the orange cup. The illumination environment corresponding to this case can only be represented by a general function illumination environment as both the direct illumination and the interreflection are significant. The hypotheses corresponding to colored diffuse reflection are less common, generally occurring when the white object is in shadow from direct sources but still experiences reflection from a nearby colored object. Colored uniform sources--blue light bulbs, for example--are not common in human environments. Given this analysis, we propose that curved and planar dielectrics under general function colored illumination be placed in tier one, and the other four hypotheses in tier two.

White metals under colored illumination form the next major branch of the taxonomy. Unlike dielectrics, incident light from almost all directions is significant to the appearance of a metal surface patch. This can be seen in box C of Figure 2.14, where interreflected light that is dim relative to the global light source still has a significant effect on the appearance of the metal object. For this reason, the hypotheses with general function colored illumination are the most common. It is rare for a metal surface to be lit only by colored uniform illumination, or to have the same color and intensity light incident from all directions as under diffuse illumination. Furthermore, unlike dielectrics, diffuse illumination environments are not good approximations because the exitant light energy field in a given direction is dependent on only one direction of the incident light energy field. Therefore, the two hypotheses with colored general function illumination belong to the first tier, and the other four hypotheses--colored diffuse and uniform illumination--belong to the second tier.

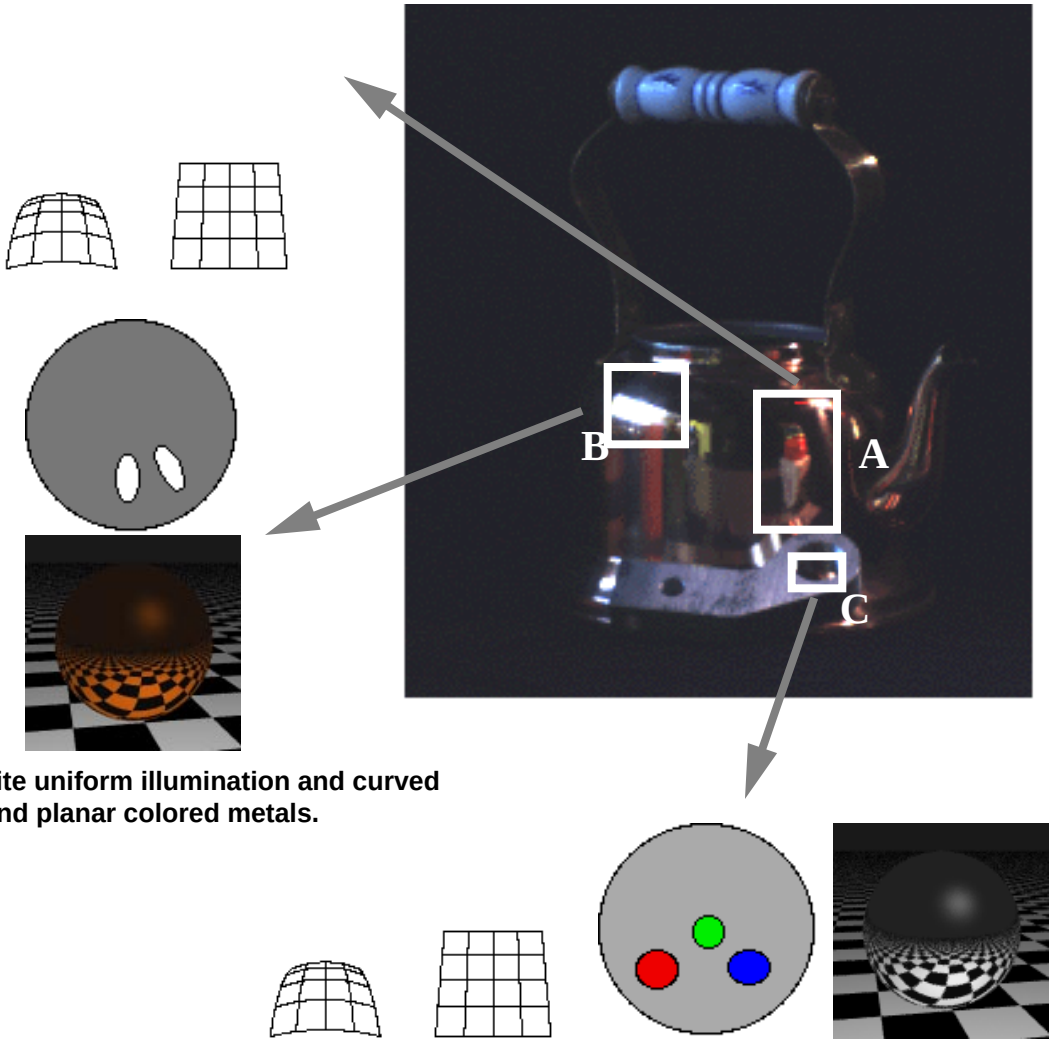
The final branch of hypotheses contains the colored metals under white and colored illumination. Consider first the six hypotheses of colored metal under colored illumination. As with grey metals, hypotheses with colored general function illumination such as box C are the most common situations for colored metal objects as seen in box A of Figure 2.14. Colored uniform and diffuse illumination are not good approximations. This places the colored general illumination hypotheses in tier one, and the other four in tier two.

With regard to the six white illumination hypotheses, we propose that uniform illumination is sufficient for modeling colored metal under white illumination such as box B of Figure 2.14. True diffuse illumination is rare--the metal object will at least be reflecting the camera! We realize that the approximation of general white illumination by white uniform may not be valid for all cases, but it is sufficient for our current discussion. From this analysis, the two hypotheses with uniform illumination belong in tier one; the other four belong in tier two.

The analysis of the two branches of dielectrics displaying surface reflection is similar to that for white dielectrics under colored illumination. In both cases the color of the reflection is due to the



Box C: Colored general illumination and curved and planar colored metals.



Box B: White uniform illumination and curved and planar colored metals.

Box A: Colored general illumination, curved and planar grey metals.

Figure 2.14: Fundamental metal hypotheses.

illuminant. If we continue with the assumption that colored diffuse and colored uniform lighting are rare cases of general colored illumination, then the former 8 hypotheses fall into tier 2 and the latter 4 in tier 1. The case of white and colored dielectrics displaying complex colored surface reflection is common when looking at especially shiny or smooth objects. A billiard ball, for example, will clearly reflect the environment around it. For rougher or less shiny objects, however, the body reflection from a direct illumination source will overshadow the surface reflection of the environment. Because of their limited applicability to non-shiny objects, the hypotheses proposing colored and white dielectrics displaying surface reflection of a colored general illumination environment may or may not belong in tier one depending upon the nature of the scenes being examined. For this discussion we leave them in tier one.

The overall result of this analysis is that there are 18 common fundamental hypotheses for a colored region in tier one, and 30 less common or rare fundamental hypotheses in tier two. Note that all seven non-specular illumination/transfer function combinations are present in either Figure 2.13 or Figure 2.14; all of these fundamental hypotheses may exist in deceptively simple images.

For a white or grey region, whose potential hypotheses are given in Figure 2.12, a similar analysis applies. We begin with the branch of hypotheses proposing grey metals. As with colored metals under white illumination, we propose that white uniform illumination is both more useful than white diffuse illumination and a reasonable approximation of white general illumination. Therefore, the two hypotheses proposing grey metals under white uniform illumination fall into tier one, the other four into tier 2.

The second branch, white dielectrics, is similar to the case of colored dielectrics under white illumination. As in that case, we argue that white uniform illumination is a good enough approximation of white general illumination to place the latter in tier two. White dielectrics under white diffuse illumination, a good approximation for shadowed areas, and white dielectrics under white uniform illumination fall in tier one.

The last two branches, white and colored dielectrics displaying surface reflection, have similar analyses. In both cases, we propose that white uniform illumination is a good enough approximation to white general illumination to place the latter in tier two and the former in tier one. The remaining hypotheses all propose dielectrics displaying surface reflection of diffuse white environments. We give two arguments for why these hypotheses belong in tier two. First, a rough or non-shiny dielectric will not show significant surface reflection except for a strong light source. Diffuse illumination is normally dim compared to direct illumination, making it unlikely that it would have a significant effect upon the appearance of a rough dielectric. Second, a very shiny dielectric will not only reflect the diffuse illumination, but also the objects in the environment around it, forcing the illumination into the general colored illumination category rather than white diffuse. Therefore, we conclude that colored and white dielectrics specularly reflecting a white diffuse environment properly belong in tier two.

Overall, this analysis results in 10 white/grey hypotheses in tier one, and 14 in tier two.

2.4.3 Merging the Fundamental Hypotheses

Having developed a small set of physical hypotheses for describing a given image region, we attach this hypothesis list to each of the simple regions initially found in an image. In general, we define a segmentation of the image to be a set of hypotheses, one from each initial region, that

covers the image. It is important to realize that, while all of the fundamental hypotheses are possible explanations for a given image region, only one hypothesis is correct.

To find these correct hypotheses and obtain a good segmentation, we need to minimize the number of hypotheses in the segmentation by combining hypotheses that are compatible (i.e., that appear to belong to the same object by some criterion). Combining compatible hypotheses is the key to obtaining an intelligent segmentation.

To explore the space of hypothesis combinations, we start by looking at pairs of hypotheses. In particular, we explore the different combinations of hypotheses from two adjacent regions.

A brute force approach would look at all combinations of the fundamental hypotheses for each adjacent region pair. Unfortunately, a brute force method is not only unreasonable, but also too computationally expensive for even simple images because of the exponential explosion of the number of hypotheses. For a segmentation method based on this framework to be tractable, the interaction between hypothesis regions and the nature of the physical explanations must provide constraints.

For a merger between regions to be desirable, there must be some coherence between the hypothesized physical explanations. This coherence manifests itself in the three general variables: shape, illumination, and transfer function. If two neighboring hypotheses are sufficiently similar, it may be a desirable merger. By definition there must be a discontinuity between neighboring regions. The particular form of this discontinuity is dependent upon the initial segmentation method. This implies a discontinuity in at least one of the hypothesis elements. Because of the general viewpoint principle--things don't line up for almost all viewpoints [55]--having two simultaneous discontinuities along the border of adjacent hypothesis regions is an unlikely occurrence if the regions belong to the same object. Therefore, we propose that for adjacent hypothesis regions to belong to the same object the discontinuity between them must be a simple one and *must involve only one of the hypothesis elements*.

In addition to this general postulate, we apply four other rules:

1. hypothesis regions of differing materials should not be merged (this includes differently colored metals such as Box I in Figure 2.14),
2. hypothesis regions with incompatible shape boundaries should not be merged,
3. hypothesis regions of differing color that propose the physical explanation to be colored metal under white illumination should not be merged, and
4. hypothesis regions proposing different colors of diffuse illumination should not be merged.

While the first rule may be restrictive at a more abstract level--e.g, object recognition--it is necessary to make the problem tractable. It can also be argued that combining different material types is not appropriate for a low-level segmentation algorithm. The second rule is necessary so that overlapping objects with similar characteristics are not merged. The third rule results from the fact that the surface reflection, or material properties of the surface, determine the color of hypotheses proposing colored metal under white illumination. Therefore, if two of these hypothesis regions differ in color but have the same illumination environment, they must be different materials and should not be merged.

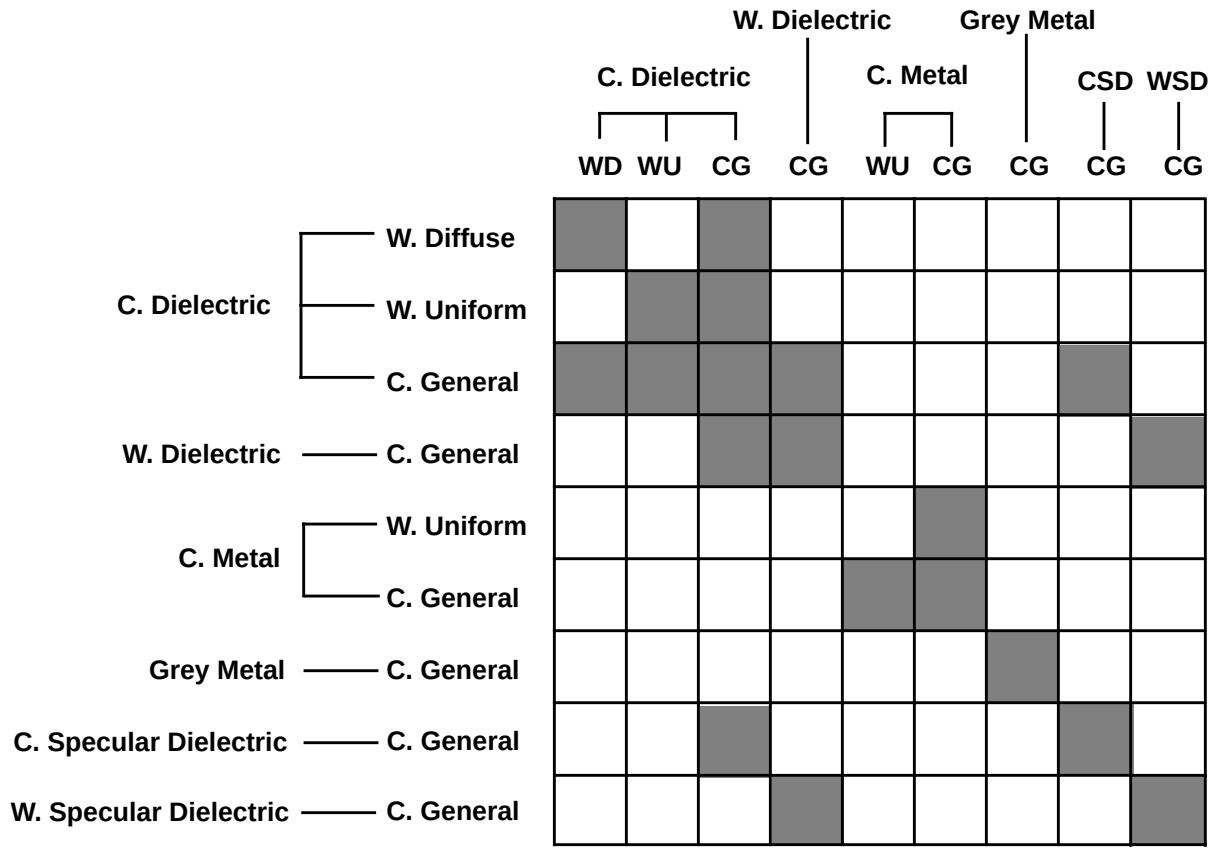


Figure 2.15: Table of potential merges of the fundamental hypotheses for two colored regions. Shaded boxes indicated hypothesis pairs that should be tested for compatibility. Unshaded squares are incompatible.

The last rule is due to the physics of illumination. Diffuse illumination specifies that the color and intensity of the illumination is constant over the illumination hemisphere. Now consider two adjacent surface patches under differently colored diffuse illumination. If the adjacent patches are both visible and part of the same surface, there will be overlap between the illumination environments. Therefore, they cannot both be illuminated by differently colored diffuse illumination unless the illumination is such that each point on the illumination hemisphere appears one color from one appearance patch and a different color from the adjacent appearance patch. Such an illumination environment is unlikely at best and is reasonably discarded.

The result of applying these rules to the merger of two adjacent colored image regions is shown in Figure 2.15. Because we only have to analyze the shaded boxes, instead of having to consider 182 combinations, or $2 * 9^2$, we only need to look at 40. The importance of this result is that we *do not significantly increase* the number of hypotheses being considered for the two regions. Instead of having 18 hypotheses each for two regions we now have 40 for the composite region. The rules reduce the number of mergers that need to be considered by at least a factor of 4.

For completeness, we also show the potential merges of a colored and a white region in Figure 2.16, and the potential merges of two white regions in Figure 2.17. Like Figure 2.15, these tables

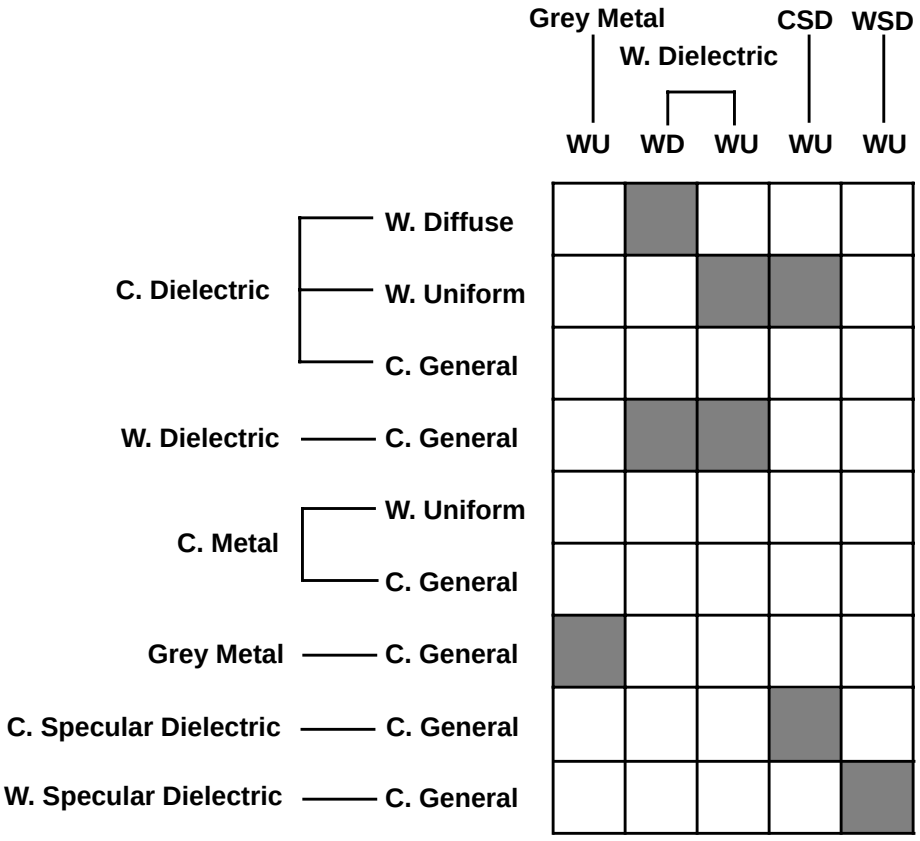


Figure 2.16: Table of potential merges of the fundamental hypotheses for a colored region and a white region. Shaded boxes indicated hypothesis pairs that should be tested for compatibil-

also demonstrate the considerable reduction in the potentially interesting hypothesis pairs. Of the 90 possible combinations for a white and a colored region, only 16 are of potential interest, and of the 50 possible combinations of two white regions, only 22 are of potential interest. Note, however, that the case of merging two white regions will be rare depending on the initial segmentation algorithm. If the initial segmentation method is based upon color, for example, then having two separate, but adjacent white regions will not occur except in unusual cases.

2.5. Merger analysis

As shown in Figure 2.15, there are 40 potential mergers that must be considered for each pair of adjacent hypothesis regions. A merger is desirable to make if it can be ascertained that only a single discontinuity exists between the two regions. Furthermore, if the defining characteristic of the initial regions is coherence in color space, shape should not be the cause of a region's boundary. Therefore, if two hypotheses are part of the same surface, then either the transfer function or the illumination must contain any discontinuity between the two hypotheses.

This implies that shape is always a major factor in the decision to merge two hypotheses for all 30 different illumination/transfer function combinations. The next three subsections give a detailed analysis of the merge requirements for three representative cases: row 2, column 2 of Figure 2.15, row 2, column 4 of Figure 2.16, and row 3, column 3 of Figure 2.15. The first case represents a

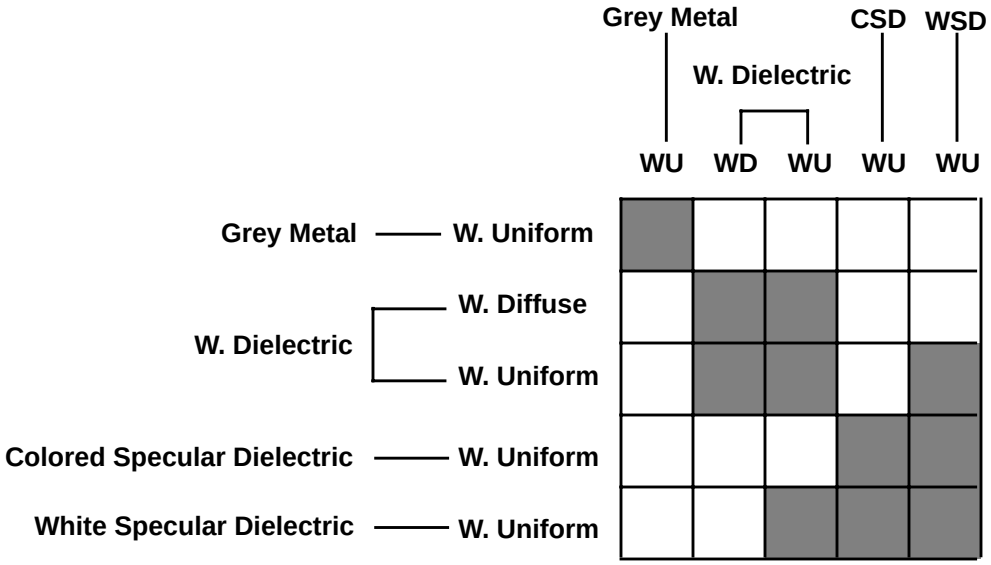


Figure 2.17: Table of potential merges of the fundamental hypotheses for two white regions. Shaded boxes indicated hypothesis pairs that should be tested for compatibility.

merger between two colored dielectrics under white uniform illumination. A clear example of this case is box E of Figure 2.13. The second case is examines a merger between a colored dielectric displaying body reflection and a colored dielectric displaying surface reflection, both under white illumination. The third case is a merger between two colored dielectrics under general colored illumination. The first two cases are relevant to the segmentation algorithm described in later chapters. The third case is interesting because it highlights the potential complexity of merging two hypotheses.

2.5.1 Merging colored dielectrics under white illumination

The fundamental hypothesis specifying a colored dielectric under white uniform illumination is extremely common in human environments. It is also common for an object or surface to have patches of different colors. One example is the mug in Figure 2.13. Thus, we need to understand the issues involved in merging two hypotheses specifying color dielectrics under white uniform illumination

While the table in Figure 2.15 tells us to consider merging adjacent hypotheses proposing colored dielectrics under white illumination, it does not specify which element possesses the discontinuity. Nor does it directly tell us the nature of the discontinuity, or what methods we might use to compare the hypothesis elements for similarity.

The first task is to determine which element of the hypothesis possesses a discontinuity if the two regions are part of the same surface. If we have an initial segmentation based on color or chromaticity, this is equivalent to asking which scene element causes the color change between the regions. For this pair of hypotheses, the cause of the color change must be the transfer function as the illumination for both regions is white. Therefore, both the illumination and the shape of the two regions must be coherent to consider merging the hypothesis pair.

The next task is to determine the nature of the discontinuity. In this case the discontinuity is a

change of color in the transfer function. The question is whether the other characteristics changed as well. If the two regions belong to the same object in a scene, it is reasonable to assume that the surface patches have similar properties. For example, the surface roughness should be similar. A discontinuity in such color independent properties discourages a merger; strong similarity encourages a merger. Therefore, useful methods of analysis for this case are those that compare the illumination, shape, and non-color aspects of the transfer function.

2.5.2 Merging a colored dielectric with a highlight

As with the previous case, the transfer function contains the discontinuity between these two hypotheses if they are part of the same surface. One of the hypotheses, a colored dielectric displaying body reflection under white uniform illumination, specifies that its image region is the color of the material. The other hypothesis, a colored dielectric displaying surface reflection under white uniform illumination, is a qualitatively different subspace of the transfer function and specifies that its region reflects the color of the light source. As with the first case, the shape and illumination must be coherent for the two hypotheses to be part of the same surface.

The nature of the discontinuity is a switch between two qualitatively different transfer functions. While the first case compares two points within a single subspace of the transfer function, this case compares points in different subspaces. These two subspaces are related by their coherence in color space as identified by Shafer's dichromatic reflection model and used by Klinker *et. al.* and others [53][24][2][16] to segment images containing highlights. Therefore, not only can we use methods of analysis that look for similar illumination and shape, but also knowledge about how the two transfer function subspaces are related. The latter is a powerful statement, as it strongly constrains the relative appearance of the two regions if they are part of the same surface.

It is important to note that a second type of discontinuity may also exist in the transfer function. In addition to the qualitative change in the type of reflection, the color of the transfer function may also change between the two hypotheses. This situation would happen if a portion of a highlight region abutted a differently colored region of the same object. This case requires a more sophisticated application of the dichromatic reflection model, taking into account the relationship of the highlight hypothesis with its other neighbors.

2.5.3 Merging colored dielectrics under colored general illumination

This case is perhaps the most complex situation involving dielectric materials. The discontinuity between the two hypotheses can appear in either the transfer function or the illumination. Consider, for example, box F of Figure 2.13. The white square on the mug within box F appears both white and orange. The change in color across the surface is due to the illumination environment becoming strongly orange because of interreflection from the cup. Between the white region within box F and the green region above it, however, the illumination environment is approximately the same. In this case the change in color is due to the transfer function.

For both types of discontinuity, there is no switching between subspaces as with the previous case. Both hypotheses are simply different instantiations of the same form of the hypothesis elements. To analyze this case we need to look for similarity in the shape and either the transfer function or the illumination, depending upon the specific interpretation of the combined hypothesis.

The important point is that there are *two possible interpretations* for the combined hypothesis. In

one case the underlying transfer function remains the same and the color change is due to the illumination; in the other, the illumination remains the same and the transfer function change color. Both are equally plausible explanations for the appearance of the combined image region.

The effect is that for this hypothesis pair we have to undertake two separate analyses, and then maintain two separate aggregate hypotheses during the segmentation process. This makes it a more complex case than the other hypothesis pairs because the color change is not limited to only a single hypothesis element.

2.5.4 Merging other hypothesis pairs

Knowing where discontinuities are expected and where they are not is the key to applying vision operators to the two regions. We want to obtain measures of similarity which will allow or block a merger of the two regions. Possible methods of comparing the illumination include using light source direction estimators such as Pentland's or Zheng & Chellapa's, and illuminant chromaticity estimators such as Lee's [49][63][33]. Shape-from-shading techniques such as Bischel & Pentland's approach could be used to estimate and compare the shape of adjacent regions [4]. Another approach is to combine the illumination and shape estimation as in [5], which may lead to more robust estimates of both.

Work on roughness estimation by Novak and Stone & Shafer is also relevant for analyzing the color independent characteristics of the two regions [43][54]. By comparing the roughness of two image regions, we obtain a measure of the similarity of the non-color aspects of the transfer functions.

By applying appropriate tests for each hypothesis pair, we arrive at a measure of similarity which will encourage or discourage a merger of the two regions. Clearly, there are some cases where a lack of tools makes analysis difficult. In particular, dielectrics under general illumination and metals under any illumination present a significant challenge to existing approaches. However, there are a few cases where there are sufficient vision tools to perform the necessary calculations.

Table 1 gives a summary of the discontinuities for the 22 illumination/transfer function pairs in Figure 2.15 and Figure 2.16. The far right column also suggests possible approaches for comparing hypothesis elements for similarities. The pairs from Figure 2.17 are not included because, as noted previously, they will rarely occur given an initial segmentation based on color.

2.6. A strategy for segmentation

The previous sections presented a framework for thinking about an image in terms of the physical explanations for simple image regions. If we assume an initial segmentation based upon color, then we can attach a set of fundamental hypotheses to each image region. This is the same concept as a sensor value or group of sensor values mapping to an equivalence class of physical explanations. These equivalence classes all possess a similar form in terms of the broad hypotheses considered for each one.

However, the likelihood of each hypothesis given the sensor data is different within each equivalence class depending on the particulars of the sensor data. We can think about this at the pixel level, in which case we have little information to go on, or the region level, in which case we have more information to go on. Furthermore, if we start merging hypotheses together according to

Table 1: Merger discontinuities and methods of analysis

Input Region 1	Input Region 2	Discontinuity	Constraints/Approaches
white diffuse/ col. dielectric	white diffuse/ col. dielectric	transfer function	shape-from-shading on a cloudy day[28]
white diffuse/ col. dielectric	colored general/ col. dielectric	illumination	known object color restricts the color of region 2
white uniform/ col. dielectric	white uniform/ col. dielectric	transfer function	shape-from-shading, illuminant direction & color
white uniform/ col. dielectric	colored general/ col. dielectric	illumination	shape-from-shading, known object color restricts color of region 2
colored general/ col. dielectric	colored general/ col. dielectric	illumination or transfer function	estimate illumination environment, orientation-from-color [cite?]
colored general/ col. dielectric	colored general/ white dielectric	transfer function	illuminant color known from region 2, orientation-from-color
colored general/ white dielectric	colored general/ white dielectric	illumination	known light source color, orientation-from-color
white uniform/ colored metal	colored general/ colored metal	illumination	known metal color, estimate roughness
colored general/ colored metal	colored general/ colored metal	illumination	estimate metal color & roughness
colored general/ white metal	colored general/ white metal	illumination	estimate & compare roughness
white diffuse/ col. dielectric	white diffuse/ white dielectric	transfer function	SFS on a cloudy day
white uniform/ col. dielectric	white uniform/ white dielectric	transfer function	SFS, light source orientation
white uniform/ col. dielectric	white uniform/ col. spec. dielectric	transfer function	dichromatic reflection model analysis
colored general/ white dielectric	white diffuse/ white dielectric	illumination	orientation-from-color and SFS on a cloudy day to compare shape
colored general/ white dielectric	white uniform/ white dielectric	illumination	orientation-from-color and standard SFS to compare shape
colored general/ grey metal	white uniform/ grey metal	illumination	estimate & compare roughness
colored general/ col. spec. dielectric	white uniform/ col. spec. dielectric	illumination	estimate & compare roughness
colored general/ white spec. dielectric	white uniform/ white spec. dielectric	illumination	estimate & compare roughness
colored general/ col. spec. dielectric	colored general/ col. spec. dielectric	illumination	estimate & compare roughness
colored general/ white spec. dielectric	colored general/ white spec. dielectric	illumination	estimate & compare roughness
colored general/ col. dielectric	colored general/ col. spec. dielectric	transfer function	dichromatic reflection model analysis
colored general/ white dielectric	colored general/ white spec. dielectric	transfer function	dichromatic reflection model analysis

some criteria, then we get more information because the merges filter out some hypotheses, and we get structurally different equivalence classes for merged regions.

From this train of thought, we can begin to see the outline of a segmentation algorithm. The first step is to segment the image into simple regions that can reasonably be assumed to be portions of a single object. An example of this type of segmentation would be Klinker *et. al.*'s linear segmentation phase [24]. We then attach the set of fundamental hypotheses to each simple image region.

The next step is to determine which adjacent hypotheses could potentially merge into aggregate surfaces. We get this information from the tables in Figure 2.15, Figure 2.16, and Figure 2.17. Given that two hypotheses could potentially merge, we look at Table 1 to see which hypothesis elements must be coherent and what methods we can use to test for coherence. The result of these tests gives us information with which to make a merge decision.

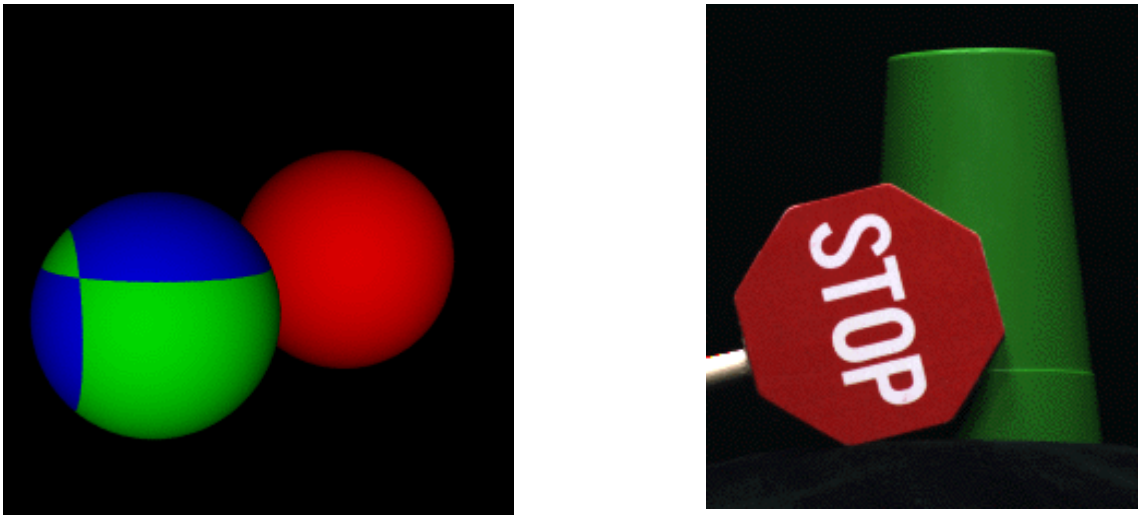
Finally, we have to search the space of segmentations, considering the possible merge and not-merge combinations. Recall that a segmentation is a set of hypotheses such that each initial region has one and only one explanation. Because we consider aggregate hypotheses, a pair of image regions may be represented by a single aggregate hypothesis.

Ultimately, we are searching for a set of likely segmentations according to some measure of goodness. This measure must be in part based upon the results of analyzing regions for similarity; an aggregate hypothesis may or may not be a better representation of a given region pair. The measure may also reflect how simple an explanation is given the image data. For example, for a uniform intensity image region, a hypothesis proposing a planar surface may be preferred over a hypothesis proposing a curved surface.

The final output of the algorithm would be a set of probable segmentations, rank-ordered according to the measure of goodness. Furthermore, each segmentation provides a physical explanation of the image and specifies which simple image regions join together to form coherent surfaces in the scene. Such an output not only contains more information about the scene than previous segmentation algorithms and finds coherent multi-colored surfaces in an image, but also returns a result that reflects the ambiguity actually present in images.

Chapter 3 gives a detailed overview of a segmentation system based on this approach. The initial system implements four of the fundamental hypotheses, two for each colored region, and two for each white region. Chapters 7 and 8 then expand the system by showing how it can rank hypotheses according to their compatibility with the image data and then increasing the initial hypothesis list to include dielectrics displaying surface reflection. The implementation shows that our framework for segmentation provides a new and effective means of approaching the segmentation problem and providing more intelligent and useful results.

Chapter 3: Segmentation algorithm



(a) (b)

Figure 3.1: (a) Synthetic image of two spheres with a single light source generated by Rayshade, (b) real image of a painted wooden stop-sign and a plastic cup illuminated by fluorescent panel lights taken in the Calibrated Imaging Laboratory, CMU.

3.1. System overview

A consequence of the framework developed in Chapter 2 is a new approach to segmenting complex scenes into regions corresponding to coherent surfaces rather than merely regions of similar color. This chapter presents an algorithm implementing this new approach. The algorithm contains four phases.

The first phase segments an image based upon normalized color. This provides a set of simple regions that can reasonably be assumed to be part of a single object. The algorithm then attaches a list of potential explanations, or hypotheses to each initial region.

The second phase examines adjacent hypotheses for compatibility. We explore two methods of compatibility testing. The first is direct comparison, which estimates the shape, illumination, and material properties of each region and directly compares their compatibility. The second uses weak tests of compatibility, which compare physical characteristics that must be compatible if two hypotheses are part of the same surface, but which are not necessarily incompatible between different objects. By using a number of these necessary, but not sufficient tests the algorithm rules out most incompatible hypothesis pairs.

The third phase builds a hypothesis graph from the results of the analysis. Each hypothesis is a

node in the graph, and edges contain information about the cost of merging adjacent hypotheses.

The fourth phase then extracts segmentations from the hypothesis graph and rank-orders them. Each segmentation contains exactly one hypothesis from each region and provides a potential physical interpretation of the scene.

3.2. Initial partitioning algorithm

To test the segmentation method, we use pictures of multi-colored objects on a black background. Figure 3.1(a) and (b) are two example test images. Figure 3.1(a) is a synthetic image created using Rayshade (a public domain ray tracer). Figure 3.1(b) was taken in the Calibrated Imaging Laboratory at Carnegie Mellon University, as was the picture of the mug in Figure 1.3(a). The pole of the stop-sign is unpainted light wood, the stop-sign itself is painted red with white lettering, and the cup is green plastic. The lighting in the image comes from two fluorescent panel lights; one is above and to the right, the other above and left.

The first step in segmentation is to identify pixel regions that display coherence in some feature space. In a color image, the most obvious characteristic linking together groups of pixels is their color. The simplest such groupings are aggregates of pixels with identical color. A reasonable starting assumption might be that a set of connected pixels with the same color correspond to a single surface patch within a scene. Using regions of uniform color overlooks much of the information contained in the image and breaks it into regions that are too small.

An approach of slightly greater complexity is to group together pixels displaying the same basic color ratios, or chromaticity, but with varying brightness. Mathematically, *chromaticity* is defined by “normalized color” coordinates, as defined in by

$$(c_{nr}, c_{ng}, c_{nb}) = \left(\frac{r}{r+g+b}, \frac{g}{r+g+b}, \frac{b}{r+g+b} \right) \quad (1)$$

where *r*, *g*, and *b* are the red, green, and blue intensity values, respectively [23]. Chromaticity can also be thought of as the hue and saturation of a color without the intensity information. Regions of uniform chromaticity are similar to Klinker’s linear clusters, and in fact equal linear clusters for scenes taken under white uniform illumination.

Klinker *et al.* [24] note that linear clusters, and therefore regions of uniform chromaticity, may represent two distinct objects if both are dark or poorly illuminated. Typically, however, a *uniform chromaticity region* [UCR] represents a single surface patch under a single illumination environment. This requires coherence from the physical elements generating the UCR. Clearly, it is possible to construct an image with UCRs that do not have such coherence in the physical world, and this approach will not correctly handle such situations.

The benefit of using UCRs is we can reasonably assume that they correspond to a single appearance patch in the physical world, setting constraints on the associated hypotheses. Over the patch the transfer function and the illumination environment are coherent. Because it is a single appearance patch we assume it is a single surface.

3.2.1 Normalized color segmentation

To find UCRs in an image, we use region growing with normalized color, or chromaticity, as the

image feature of interest. The algorithm traverses the image in scanline order looking for seed regions where the current pixel and its 8-connected neighbors have similar normalized color and none of these pixels already belong to another region or are too dark. When it finds a seed region, it puts the current pixel on a stack and begins region growing based on chromaticity.

When the region has finished growing, the search for another seed region continues until all pixels in the image have been checked. In the end, all pixels that are part of a region are marked with their region id in the region map. All other pixels are either too dark, or are part of a discontinuity or rapidly changing portion of the image. For now the algorithm ignores these pixels and concentrates on the homogeneous regions.

The initial segmentation algorithm uses four parameters to control the growing process and threshold the image. These four parameters are: the dark threshold, the local normalized color threshold, the global normalized color threshold, and the region size threshold.

The dark threshold specifies whether pixel is bright enough to be interesting. Also, as the test pictures all have black backgrounds, the dark threshold separates the background from the objects of interest. No pixel that is less than the dark threshold is assigned to a region. A typical value for the dark threshold is between 8 and 12 on a scale of [0,255], about 4% of the maximum pixel value.

The local normalized color threshold specifies how similar the normalized color of two adjacent pixels must be for them to be part of the same region. This threshold determines how smoothly the normalized color of a region may change from pixel to pixel. If an object has some texture or imperfections in its appearance, for example, the local threshold must be relaxed for the algorithm to classify the object pixels as one region. Conversely, a tighter threshold is sufficient for synthetic images or smooth, uniformly appearing objects. In both cases, however, if an image contains a smooth enough edge or slow enough transition between objects then they will be classified as the same region unless some other growing criterion is used. A typical local normalized color threshold is 0.04, where this is a Euclidean distance between two normalized colors.

The global normalized color threshold provides this second growing rule. It specifies the maximum change in normalized color that may occur between the initial seed pixel and all other pixels in the region. This threshold will halt the growth of a region even in the presence of a smooth boundary if the normalized color of the new pixels strays too far from that of the seed pixel. A typical global threshold is 0.15, where this represents the Euclidean distances between two normalized colors. Note that the chromaticity space is only two-dimensional as the sum of the chromaticity values for a point must sum to one. Thus, these two thresholds specify circles in the chromaticity space. Furthermore, the chromaticity of all pixels in a region fall within a circle with a radius equal to the global threshold and centered at the normalized color of the seed region.

Finally, the region size threshold specifies the minimum region size for a connected pixel cluster. The system uses all connected regions larger than the minimum region size and discards the rest. A standard minimum region size is 100 pixels, although images containing highlight or small object regions require a smaller minimum size.

In addition to the basic region growing, the initial segmentation algorithm contains two other steps motivated by the tests for compatibility between regions. First, after building the region map based on chromaticity, the algorithm shrinks each region once in an 8-connected manner. Because pixels near a border often overlap an adjacent region, the information they contain may be con-

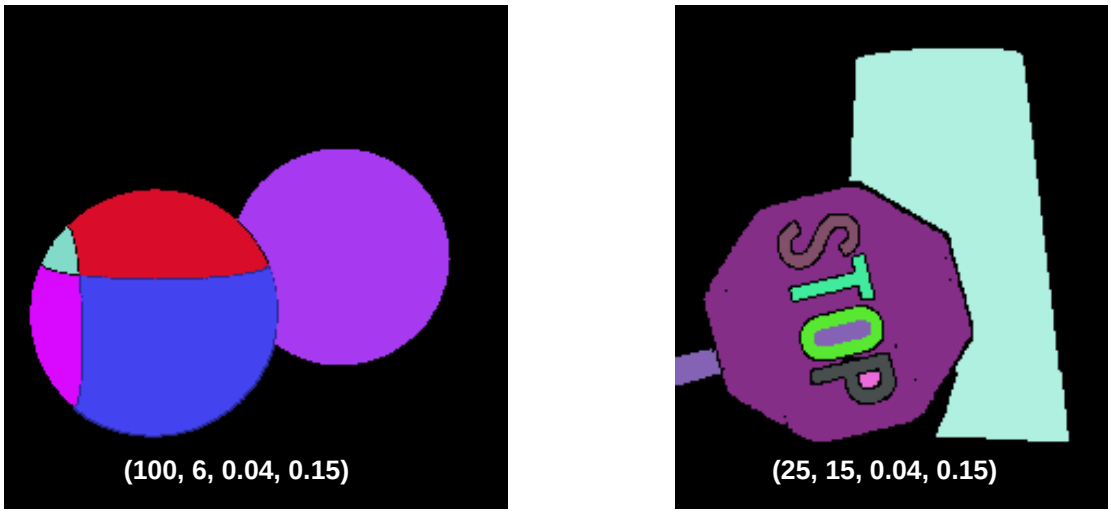


Figure 3.2: Initial segmentations of the images in Figure 3.1 (a) and (b), respectively. The numbers shown are the size threshold, dark threshold, and local and global normalized color thresholds, respectively.

taminated. This contamination causes problems for any analysis that uses border pixels to compare adjacent regions. Shrinking eliminates most of the noisy border pixels.

The shrinking step, however, may divide some regions into two or more parts. To handle this problem the algorithm includes a connected region extraction step that renumbers connected regions in the region map, giving unique identities to the separate portions of divided regions.

A brief analysis shows that the initial segmentation algorithm is linear in the number of image pixels. In the initial growing stage, each pixel is visited at least twice: once to test if it is a seed region, and once to test if it is part of a region. The algorithm will visit only the region boundary pixels more than twice. Each pixel is then visited once during the shrinking step, and twice more during the final region extraction. Although the algorithm requires a minimum of five passes through the image, its average running time is linear in the number of image pixels.

Figure 3.2 shows the initial segmentations of the two test images in Figure 3.1. The 4-tuple associated with each initial segmentation gives the parameter values for the size threshold, dark threshold, local threshold, and global threshold, respectively.

Given the existence of more complex physics-based segmentation methods, a valid question is why not use a segmentation algorithm such as Healey's normalized color method [16], Klinker *et. al.*'s linear and planar cluster algorithm [24], or Bajcsy et. al.'s normalized color method [2]? There are legitimate problems with using any of these methods.

Healey's normalized color method, while it does attempt to identify metals in an image, has two conflicts with our overall framework. First, it requires the entire scene to be illuminated by a single spectral power distribution. Interreflection, especially with respect to metals, confuses the algorithm. Second, the algorithm may confuse white or grey dielectric objects with metal objects or highlights, again causing problems.

We actually implemented Klinker *et. al.*'s linear cluster algorithm and ran it on numerous test

images. We found two problems. First, without using the complete algorithm--which requires the assumption that all objects in a scene are dielectrics--variations in the normalized color due to highlights or noise are not well captured. Second, because of the need to find linear clusters as seed regions, the method breaks down on planar surfaces and regions of almost uniform color. Since many regions in our test images are small portions of the same object, the latter occurs frequently, making it difficult for the method to identify the separate parts of the same object.

Finally, although Bajcsy *et. al.*'s algorithm does identify regions of interreflection and shadow, it requires a white reference in the image with which to obtain the color of the illumination. We want to be able to segment images without a white reference patch or white object. Furthermore, the general framework is not restricted to scenes with white global illumination, a requirement of their approach.

Overall, previous physics-based approaches are too restrictive in terms of their assumptions about the scene to use as a front-end to a more general framework. Furthermore, the region growing based upon normalized color is fast and returns a reasonable set of regions.

3.2.2 Finding border pixels

As we will see later, the border pixels for each region play an important role in testing for compatibility. After finding all of the viable initial regions the system then identifies the exterior border pixels for each one.

We use a wrap-around algorithm to find the border pixels. The algorithm starts at the border pixel nearest to the upper left corner of the image and works its way clockwise around the region. At each pixel it uses the direction between the current pixel and the previous pixel to determine the order in which to search for the next border pixel. The algorithm assumes 4-connected regions and returns a 4-connected border polygon. As the algorithm only searches the exterior border, it does not find the borders of holes within any region.

After finding the exterior region borders, the system then analyzes each border pixel to find the closest region, if any, and the closest point within the closest region. This results in sets of border pixel pairs, where each pair contains one pixel from each of the adjacent regions. This local analysis also allows the system to determine adjacency between regions.

For each border pixel the system performs the following analysis. First, it determines the local normal vector to the polygon by looking at the previous and next border pixels in the border polygon. These determine whether the search for a nearby region is horizontal, vertical, or any of the 45 directions. The search always moves away from the region.

Next, the algorithm searches along the local normal for a specific distance. For all of the test images this distance is 8 pixels. If it finds another region within this distance, then it attaches the region id and the coordinates of the nearest pixel to the current border pixel. The algorithm also increments a counter which holds the number of “hits” each region receives. If the search for adjacent pixels hits a region fewer than a specified number of times, then the algorithm does not consider the regions to be adjacent. For all of the test images, there must be at least 5 hits for the system to label two regions as adjacent. This threshold is necessary when dealing with images such as the mug which contains regions that touch for only a few pixels at the corners. Note that because the algorithm uses the local border normal to find the nearest pixel in an adjacent region; two adjacent regions may not contain exactly the same sets of border pairs.

We calculate region adjacency this way because it is a free byproduct of calculating border pixel pairs, which we have to calculate anyway. Also, as adjacent regions will not normally touch because of the shrinking step described previously, we have to search several pixels outwards from the current region to find the adjacent regions. Therefore, we cannot use more standard region adjacency calculation methods that only look at neighboring pixels [26]. In terms of computational cost, more standard methods generally examine all of the pixels in an image, whereas our method only examines border pixels and nearby pixels within a certain distance of the border, saving computation time.

3.3. Attaching hypotheses

Once the algorithm has a set of initial regions it assigns a hypothesis list to each one. The color of the region and complexity of the scene determine the content of each list. Colored regions receive only hypotheses that contain at least one colored element. Conversely, white regions receive hypotheses with no colored elements.

The general complexity of the scenes a system will deal with determines the complexity of the initial hypotheses. For example, if the scenes contain only dielectrics then the initial hypothesis lists need not contain metal hypotheses. For general purpose segmentation in an unconstrained environment, the initial hypothesis lists must expand.

The basic implementation of the algorithm uses the hypothesis list H_c = {(Colored dielectric, White Uniform, Curved), (Colored dielectric, White uniform, Planar)} for colored regions and the hypothesis list H_w = {(White dielectric, White uniform, Curved), (White dielectric, White uniform, Planar)} for white/grey regions. These hypotheses are arguably the most important fundamental hypotheses as they represent colored and white/grey dielectric surfaces like plastic, paint, ceramics, and paper. For the initial system we do not use the other fundamental hypotheses for two reasons. First, we wanted to initially use hypotheses for which effective single image methods of analysis exist. This is a limitation for metal surfaces, in particular. Second, we wanted to minimize the complexity of the problem during the initial system development. Given the initial system, we show how to expand the initial hypothesis list in Chapter 8.

After the initial segmentation, a region is labeled as white/grey if

$$(c_{nr} - 0.33)^2 + (c_{ng} - 0.33)^2 + (c_{nb} - 0.33)^2 < 0.0016 \quad (2)$$

where (c_{nr}, c_{ng}, c_{nb}) is the average normalized color of the region defined by (1). This defines a circle in color space around the perfectly white normalized color, which is (0.33, 0.33, 0.33). The images in the test set, taken under both incandescent and fluorescent lighting, determined the choice of threshold.

In Chapter 8 we expand the initial hypothesis list to include planar and curved colored dielectrics displaying surface reflection. Because the illumination is still white, and surface reflection is the color of the light source, this hypothesis falls into the white region lists. Using the expanded lists, the algorithm segments scenes containing multi-colored dielectrics displaying surface reflection into coherent surfaces. This directly improves upon previous physics-based segmentation methods because it finds coherent multi-colored surfaces, relaxing the assumption of uniformly colored objects. Furthermore, it demonstrates the benefits of a general framework: the algorithm is

able to handle more complex scenes by adding more hypotheses to the initial lists and using appropriate tests of hypothesis compatibility.

3.4. Hypothesis analysis

After finding the initial regions and specifying their hypothesis lists, the algorithm attempts to determine which hypotheses of neighboring regions are compatible. In other words, it tries to answer the question: which hypotheses should be merged into a single surface?

If it could estimate and represent each element of each hypothesis for all of the initial hypothesis lists, then the algorithm would need only to look at the tables in Figure 2.15, Figure 2.16, and Figure 2.17 to find the possible mergers and then directly compare neighboring hypothesis pairs. If the elements for two adjacent hypotheses h_1 and h_2 were coherent according to a specified set of criteria, then the regions corresponding to these hypotheses should be part of the same object in any segmentation using h_1 and h_2.

It is important to realize that different hypothesis pairs for the same two regions may not produce the same answer. For example, the adjacent hypothesis pair P_1 = ((colored dielectric, uniform illumination, curved), (colored dielectric, uniform illumination, curved)) may exhibit coherence, but the hypothesis pair P_2 = ((colored dielectric, uniform illumination, curved), (colored dielectric displaying surface reflection, uniform illumination, curved)) may not. The basic reason for this is that different hypotheses produce different interpretations of the same image data. If a highlight on a smoothly curving surface is interpreted as a dielectric displaying body reflection then the estimated surface will probably contain significant curvature. Alternatively, if it is interpreted as a highlight, the estimated surface should more closely follow the actual surface shape.

During the course of developing this portion of the algorithm we explored two methods for proceeding with the analysis. The more obvious and direct method we call *direct instantiation*. This involves finding estimates of and representations for the specific shape, illumination environment, and transfer function of each hypothesis. By directly comparing the representations of two adjacent hypotheses, we obtain an estimate of how similar they are. As this test compares the intrinsic characteristics, what we are looking for is similarity or compatibility in the intrinsic characteristics. Chapter 4 describes this approach in detail.

An alternative method of analysis, *weak compatibility testing*, does not directly model the hypothesis elements. Instead, it tests certain physical characteristics of adjacent hypotheses. The similarity of these characteristics are necessary but not sufficient tests of hypothesis compatibility. By using multiple tests, however, weak compatibility testing succeeds in finding most incompatible hypothesis pairs. The focus of weak compatibility testing is to search for incompatibility between adjacent hypotheses. Chapter 5 gives a detailed description of our implementation of weak compatibility testing.

Whichever method is used, the system examines every pair of potentially compatible hypotheses. The specific characteristics of the hypotheses determine which methods of analysis the system applies. For example, given the two hypothesis pairs P_1 and P_2 defined above, P_1 requires tools that work on surfaces showing only body reflection, P_2 requires tools that work with both body and surface reflection.

3.4.1 Direct instantiation

Direct instantiation was our first attempt at determining hypothesis compatibility. We tried to harness traditional methods of image analysis to obtain estimates of the shape and illumination of adjacent hypotheses.

While this approach is theoretically attractive, direct instantiation of hypotheses is difficult. We implemented the direct instantiation approach for the hypotheses (Colored plastic, White Uniform illumination, Curved) and (White plastic, White Uniform illumination, Curved) for which some tools of analysis do exist for finding both the shape and illumination of a scene. We used Zheng & Chellappa's illuminant direction estimator [63] and Bichsel & Pentland's shape-from shading algorithm [4]. These experiments and their results are detailed in Chapter 4.

The experiments showed that existing tools for analyzing the intrinsic characteristics of a scene cannot, in general, be used on small regions of an image because such image regions violate basic assumptions necessary for the tools to function properly. The traditional literature generally assumes that an algorithm is applied to an entire object or image, not a small portion of one object. Furthermore, if we attempt to generalize direct instantiation to other hypotheses or more complex situations, we are currently limited by the lack of image analysis tools.

Another, perhaps even more important drawback of direct instantiation is that it forces the system to make commitments about hypothesis characteristics early in the segmentation process when it has the least information about the scene. Because of this, while direct comparison of the hypothesis elements is theoretically the best way to make merge/not-merge determinations, it is more likely to be wrong.

3.4.2 Weak compatibility testing

An alternative to direct instantiation is to use the knowledge constraints provided by the hypotheses to find local physical characteristics that have a predictable relationship between hypothesis pairs that are part of the same object. If the characteristics do not match the prediction, we can rule out merging the adjacent hypotheses. However, the converse is not necessarily true; a match may not be sufficient to rule out the possibility that the regions are different surfaces.

Therefore, these comparisons are necessary but not sufficient tests of compatibility, or *weak compatibility tests*. As we can calculate these physical characteristics locally, however, they are more appropriate for region-based analysis than the direct-instantiation techniques. Furthermore, the weak methods are more robust and give good results. The basic idea is that by using multiple weak compatibility tests we hope to find all, or almost all of the discontinuities. Chapter 5 presents a detailed analysis of the three tests used by the basic system.

3.5. Creating the hypothesis graph

The system organizes the information it obtains in the analysis stage by generating a graph representation of the space of possible segmentations. This allows for both local and global reasoning about how the hypotheses fit together. It is important to realize that the analysis stage is only a local analysis, focusing on a single pair of hypotheses at a time. What we are looking for, however, is a good global solution which maximizes some function of the local results. The hypothesis graph allows us to search the set of possible segmentations and find good solutions.

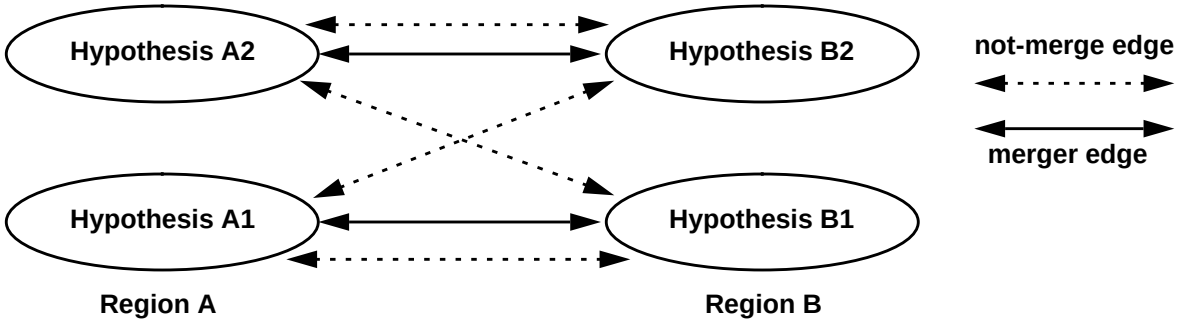


Figure 3.3: Hypotheses for two regions. Hypothesis pairs (A2, B1) and (A1, B2) are incompatible and discontinuity edges connect them. Hypothesis pairs (A1, B1) and (A2, B2) are potentially compatible and both merge and discontinuity edges connect them.

The form of the hypothesis graph is as follows. Each hypothesis in the image forms a node, and edges connect all hypotheses that are adjacent in the image as shown in Figure 3.3. If two adjacent hypotheses cannot merge, then a single *discontinuity edge* connects them as between hypotheses A1 and B2 in Figure 3.3. Alternatively, if the adjacent hypotheses potentially can merge according to Figure 2.15, Figure 2.16, or Figure 2.17, then two edges connect the corresponding pair of nodes as between the hypotheses A1 and B1 in Figure 3.3. The dashed edge is a discontinuity edge, indicating the cost of not merging the two hypotheses. The solid edge is a *merge edge*, whose value is based upon the results of the pair-wise local analysis. All edges have values in the range [0..1].

Given a complete hypothesis graph of this form, the set of possible segmentations of the image is the set of subgraphs such that each subgraph includes exactly one hypothesis, or node from each region and exactly one edge connecting each adjacent pair of hypotheses. To find a segmentation, therefore, requires the selection not only of a single hypothesis to represent each region, but also the type of edges connecting each hypothesis to its neighbors. For example, one valid segmentation of the hypothesis graph in Figure 3.3 would be hypotheses A2 and B2 connected by a merge edge. This hypothesis graph contains six valid segmentations in all, one for each edge in the graph. Figure 3.4 shows a hypothesis graph like Figure 3.3 with edge weights and all of the possible segmentations for it.

The most important aspect of generating the hypothesis graph is how to select the edge weights, as they are the basis for classifying good segmentations. This problem has two parts. The first is how to map the local analysis results to a single weight for the merge edge. The second is how to assign values to the discontinuity edges.

Assigning weights to the merge edges is straightforward; the weight of a merge edge should be proportional to the results of the compatibility tests. If the compatibility tests find strong coherence between two hypotheses, then we assign the merge edge a value approaching one. If, on the other hand, the tests indicate little coherence between a hypothesis pair, then the edge receives a value approaching zero. Given this method of assignment, then good segmentations will be those that *maximize* the values of the edges in the final segmentation. Figure 3.4, for example, shows a hypothesis graph and it's valid segmentations in order of their likelihood.

Chapters 4 and 5 describe the mapping from compatibility tests to merge edge values in more

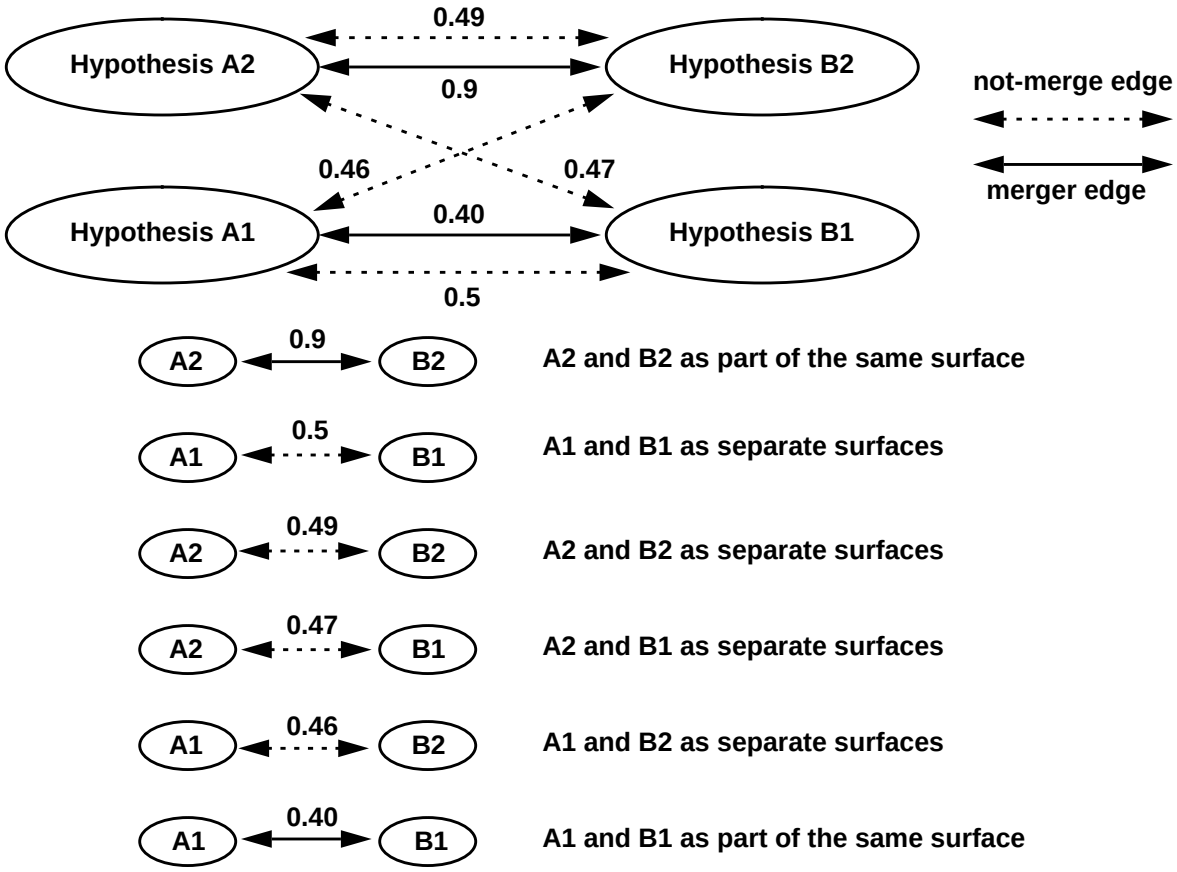


Figure 3.4: Example hypothesis graph and the set of valid segmentations ranked according to the sum of the edge values in each segmentation.

detail. In short, each of the compatibility tests maps to a likelihood in the range [0..1]. The system assigns weighted averages of these likelihoods to the merge edges. Results close to one indicate strong coherence between hypotheses; results close to zero indicate a lack of coherence.

A more difficult question is what value to assign to the discontinuity edges. A simple thought experiment shows that it must be somewhere in the middle of the [0..1] range. Consider, for example, the situation where all discontinuity edges receive a value close to 0. In this case the merge edges would be preferred in almost every case, and the best segmentations of the image would be those where every region is part of the same surface. For example, if the not-merge edges in Figure 3.4 possessed values close to zero, the top two segmentations would be those joining A2 and B2, and A1 and B1, respectively.

Alternatively, if the value of the discontinuity edges is 1, or close to 1, then the discontinuity edges would be preferred in almost all cases, resulting in segmentations where every initial region is a distinct surface. In Figure 3.4 this would mean that the system would prefer all of the segmentations proposing regions A and B as distinct surfaces.

We could define the weight of the discontinuity edges to be one minus the value of the associated merge edges. This approach, however, presents two problems. The first is that it does not specify how to assign values to discontinuity edges between incompatible hypotheses, or hypotheses with

no corresponding merge edge. For example, only a discontinuity edge connects hypotheses A2 and B1. If these discontinuity edges receive a value close to one, then good segmentations will be those containing incompatible hypothesis pairs. Conversely, if they receive a value close to zero, then the system may never select these incompatible hypothesis pairs.

The second problem is that we could still get discontinuity edge values close to zero or close to one. If, for example, the edge in Figure 3.4 connecting hypotheses A1 and B1 had a value of 0.05, then the discontinuity edge for that pair would have a value of 0.95. This would make the segmentation specifying hypotheses A1 and B1 as separate surfaces preferable to the segmentation containing the 0.9 merge edge between hypotheses A2 and B2. We argue that this should not be the case.

When selecting between a merge edge and a discontinuity edge of similar values we argue that by selecting the coherent hypothesis pair we maximally reduce the complexity of the segmentation. In other words, if there is a strong merge edge connecting two hypotheses of adjacent regions we want to prefer that pair over any discontinuity edge. By selecting a merge edge, which implies there is at most one discontinuity between the elements of the two hypotheses, we are saying these two image regions are part of the same surface and we can use a single model for the coherent hypothesis elements. The returns to concept of the Minimum Description Length described in Chapter 2 [51]. The best segmentation is that which best describes the scene with the minimum length model.

Ultimately, the value of the discontinuity edges should be related to that of the merge edges. We should prefer a discontinuity edge to a merge edge when its associated merge edge is weak. For example, in Figure 3.4 the merge edge connecting hypotheses A1 and B1 is weak with a value of 0.4. It makes sense to select the not-merge edge in that case. In an attempt to find a balance between the different types of edges, we assign an initial value of 0.5 to all discontinuity edges. To make this selection effective, we tune the compatibility test likelihoods so that results greater than 0.5 are likely merges, and results less than 0.5 are unlikely. Chapter 5 describes this process.

Note that if we have a rank-ordering of the hypotheses for a region we can adjust the discontinuity edge values to reflect the ordering. Such a rank-ordering must be based upon a comparison of each hypothesis to its region's image data. For example, we might prefer a hypothesis proposing a planar surface over a hypothesis proposing a curved surface for a region of uniform intensity. We explore this direction in Chapter 7. However, in the interest of maintaining the tuning between the discontinuity and merge edges, such adjustments must be small and the values should remain close to 0.5.

The complete hypothesis graph for Figure 3.1(a) is shown in Figure 3.5. The values given for the merge edges reflect a weighted average of the three weak compatibility tests. Chapter 5 presents the specific weights and details of the analysis. The hypothesis graph for Figure 3.1(b) is shown in Figure 3.6 without the discontinuity edges for clarity.

In Figure 3.5 we see that strong merge edges connect the different regions of the blue and green sphere. The most likely final segmentations, therefore, will merge the hypotheses for that sphere. However, weak merge edges connect the red sphere to the blue and green sphere. Therefore, the most likely final segmentations will specify the two spheres as different surfaces.

A similar situation exists with the stop-sign and cup hypothesis graph. Weak merge edges connect

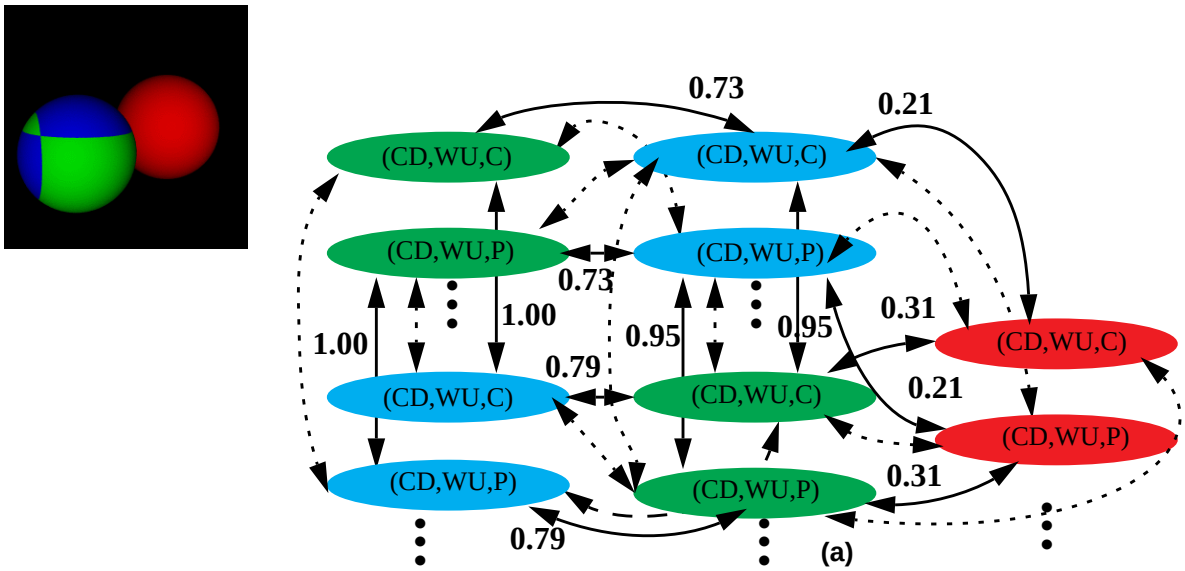


Figure 3.5: Complete hypothesis graph for Figure 3.1(a). The solid lines indicate merge edges with the given likelihoods. The dashed lines indicate discontinuity edges with values of 0.5.

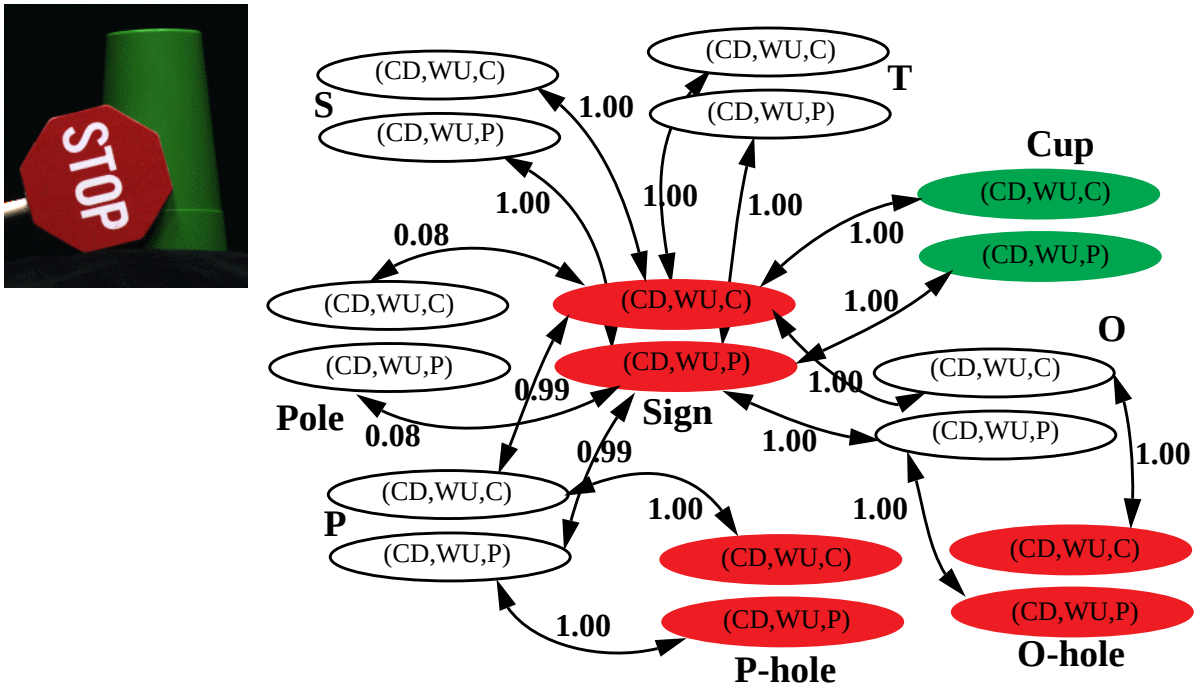


Figure 3.6: Hypothesis graph for Figure 3.1(b). For clarity, only the merge edges and their values are shown.

the pole and the sign, and the cup and the sign. Strong merge edges connect the sign and its letters. The most likely final segmentations, therefore, will merge the hypotheses for the sign and its letters into a single surface, while specifying that the cup and the pole as separate surfaces. The next

section describes how we can find good segmentations of the image given the hypothesis graph.

3.6. Extracting segmentations

Before selecting an approach to extracting segmentations from the hypothesis graph, we must first determine the extent of the space of segmentations. If it is a simple matter to search the space of possible segmentations, then we can always find the best, or most likely segmentation of the image. If, on the other hand, the space is too large, then we must turn to heuristic methods to find a solution.

3.6.1 Characterizing the space of segmentations

The most complex, or worst case hypothesis graph is one where any combination of hypotheses is a legal and consistent segmentation. The set hypothesis graphs where any combination of hypotheses is legal is the set of graphs containing no loops.

One example of a graph with loops is the hypothesis graph of the two-spheres image in Figure 3.5. The hypotheses for the red region and the adjacent blue and green regions form a loop graph. If a merge edge connects the blue and green regions for a given segmentation, then this constrains the possible edges connecting the pair to the red region. Either merge edges must connect all three regions or discontinuity edges must connect the red region to the other two.

Figure 3.6, however, is a worst case graph because it contains no loops and any combination of hypotheses is possible. Given that the stop-sign and cup image is not an unusual image, examining the worst-case graph is a reasonable way to understand the complexity of the segmentation space. The worst-case scenario also gives an upper bound on the size of the segmentation space in the more general case of graphs with or without loops. The following theorem gives the upper bound on the size of the space of segmentations.

Theorem 1: The number of segmentations S for a graph without loops is given by

$$S = NE^{R-1} \tag{3}$$

where N is the number of hypotheses per region, E is the number of edges connecting a single hypothesis to all of the hypotheses in an adjacent region, and R is the total number of regions in the graph.

Proof: We can use induction to show Theorem 1. First, consider a hypothesis graph with a single region. As there are N hypotheses per region, there are N possible segmentations for this graph.

Now consider a hypothesis graph containing R regions $Q_1 ... Q_R$ and no loops. Let the number of segmentations contained in this graph be S_R. If we add a new region Q_{R+1} as a neighbor to any single existing region Q_i, then each of the S_R segmentations of the R regions attaches to E hypotheses in region Q_{R+1}. This multiplies S_R by E, giving $S_{R+1} = S_R * E$ as the number of distinct segmentations of the new graph with R+1 regions.

By combining these two observations we get (3) as the general formula for the

number of possible segmentations contained in a hypothesis graph with no loops. •

What Theorem 1 tells us is that testing all possible segmentations of an image is undesirable even for fairly simple images. With only nine initial regions, the stop-sign and cup image has $2\left(3^8\right) = 13122$ possible segmentations. While a computer could search all possible segmentations for this case in a reasonable amount of time, an image with just 25 regions would have 5.6×10^{11} possible segmentations. Using a computer that could test 50 million segmentations per second, this image would take about three hours to process. Simply put, it is beyond the capabilities of current computer technology to examine all possible segmentations of images with more than 25-30 initial regions.

3.6.2 Review of clustering techniques

Unable to test all possible cases, we turn to heuristic methods to extract segmentations from the hypothesis graph. Agglomerative clustering, or region merging is a common technique that resembles extracting segmentations from the hypothesis graph. In both cases we search for the best groupings of pixels, regions, or nodes.

Region merging algorithms calculate distance metrics between neighboring elements and then group similar elements using an iterative process. A number of researchers use clustering techniques for image segmentation.

The most relevant work is by LeValle and Hutchinson [30]. They developed a step-wise optimal algorithm, which they call the highest-probability-first algorithm, to extract segmentations from a single-layer graph of nodes and probabilities. Both they and Panjwani & Healey have used it to segment images containing texture [30][46]. At each step the algorithm merges the most likely two nodes until it reaches a threshold based on the number of regions or the likelihood of the best merge.

Roughly equivalent to the highest-probability-first algorithm is the use of dendograms. As described by Duda & Hart, a dendogram is a tree data structure used for unsupervised, hierarchical clustering [12]. The leaves of the tree are the individual regions, and at each level two regions merge to form a new node. Each layer of the dendogram is a different clustering, or segmentation, or the data. Which two regions merge at each layer depends upon the task. A common method is to merge the two nodes which most belong together, which is the same as the highest-probability-first algorithm. Which layer to choose as the final segmentation is the equivalent of deciding when to stop merging with the highest-probability-first algorithm. Both Wallace and Krumm used dendograms for the segmentation of data, Krumm for the segmentation of images with texture [58][27]. Note that neither the highest-probability first algorithm nor the dendogram approach are guaranteed to encounter the globally optimal solution.

3.6.3 Modified highest-probability first algorithm

Our segmentation extraction algorithm is basically the highest-probability first algorithm modified for the multi-layer hypothesis graphs. Although it may be a direction of future research, we do not maintain the different layers during the segmentation as with the dendogram approach. Figure 3.7 gives a graphic demonstration of the modified highest-probability first algorithm for the case of three regions all adjacent to one another.

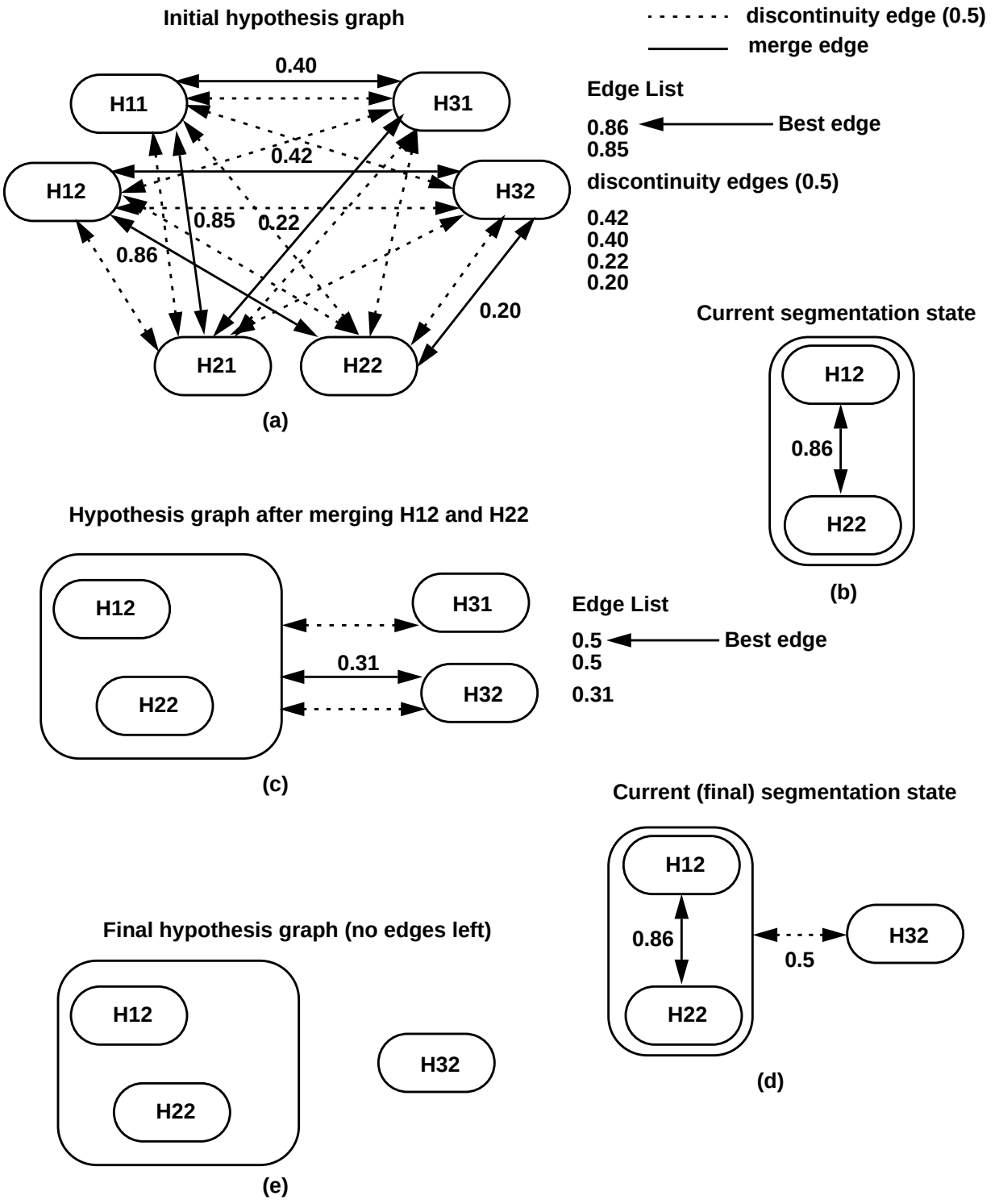


Figure 3.7: (a) Initial hypothesis graph, (b) segmentation state after merging H12 and H22, (c) hypothesis graph after merging H12 and H22 and updating the edges (d) segmentation state after adding H32 and the discontinuity edge, (e) final hypothesis graph state. After (e) there are no edges left so the algorithm terminates.

The segmentation extraction algorithm begins with an initialization step that sets up two lists: one contains all of the initial hypotheses, the other contains all of the edges in the initial hypothesis graph sorted in descending order. This step also initializes the segmentation map. This map contains one hypothesis opening for each initial image region and one edge opening for each adjacent hypothesis pair. These openings are initially empty. Figure 3.7(a) shows the initial hypothesis graph and edge list.

The main loop begins by taking the best edge from the edge list. In Figure 3.7(a) the best edge is a merge edge connecting H12 and H22 with a value of 0.86. If either or both of the hypotheses connected by the edge are not already in the segmentation map, then the algorithm puts them there. It also places the selected edge into its appropriate position in the segmentation map. Figure 3.7(b) shows the state of the segmentation map after this step.

Note that after placing one of the hypotheses for a region into the segmentation map the algorithm must remove all other hypotheses for that region from the hypothesis list. Furthermore, it must remove all edges associated with those alternative hypotheses from the edge list. This happens whether or not the edge indicates a merge or a discontinuity. For example, in Figure 3.7 we see that the hypothesis graph no longer contains H11 and H21 as alternative explanations for those regions are already in the current segmentation state.

In addition to the alternative hypotheses, if the selected edge is a merge edge, then the algorithm removes the individual hypotheses connected by that edge from the hypothesis list and replaces them with a single aggregate hypothesis. It then removes all edges connected to the individual hypotheses, recalculates the edge values between the aggregate hypothesis and its neighbors, and places the new edges into the edge list.

For example, in Figure 3.8 in the algorithm selects 0.86 the merge edge connecting H12 and H22. It then removes H12 and H22 from the graph as well as the merge and discontinuity edges connecting H31 and H32 to H12 and H22. The algorithm also removes H11 and H21 and all of their connecting edges from the graph as they are alternative explanations for the regions represented by H12 and H22. After these removals, the algorithm replaces H12 and H22 with a single aggregate region containing both and recalculates all of the edges connecting the new aggregate region to its neighbors. Figure 3.7(c) shows the state of the hypothesis graph after merging H12 and H22.

If, on the other hand, the edge is a discontinuity edge, then the algorithm leaves the individual hypotheses in the hypothesis list and only removes the discontinuity edge placed in the segmentation and any other edges connecting the same two hypotheses. In Figure 3.7(c), for example, the best edge is one of the two discontinuity edges connecting the aggregate hypothesis with H31 or H32. In this case we select the discontinuity edge connecting H32 with the aggregate hypothesis. Thus, the algorithm places H32 and the not-merge edge into the segmentation and removes the alternative merge edge. Figure 3.7(d) shows the state of the segmentation after this second pass.

One important side note is that we use averaging to combine edges together. To get the value of the merge edge between the aggregate region and H31 and H32 in Figure 3.7(c), for example, the algorithm averages the values of the merge edges connecting the aggregate region's component hypotheses to H32, in this case giving a value of 0.31. What is important to note is that this can alter the relationship between regions. For example, if a 0.60 edge connected H12 and H32, indicating a likely merge, the average value between H12 and H3 after the merge would be 0.4, indicating a likely discontinuity. In this way, hypothesis pairs with strong merge or discontinuity

values can influence more ambivalent hypothesis pairs.

This set of actions repeats until the edge list is empty. At that point the segmentation map may or may not be complete. In Figure 3.7(d) the segmentation is complete and, as shown in Figure 3.7(e) there are no edges left in the hypothesis graph, terminating the algorithm. An isolated region in an image, however, will have no neighbors and, therefore, no edges connected to its hypotheses. If the segmentation is incomplete, the algorithm completes it by randomly choosing one hypothesis from each isolated region's hypothesis list. Note, if the algorithm is given a rank-ordering for the hypotheses, then it could use that to make a selection instead of randomly choosing.

Running the algorithm once on the entire hypothesis graph returns a single segmentation. The sum of the values of all of the edges contained in the segmentation measures its quality. In most cases, this segmentation should be good. Given that the segmentation extraction algorithm is not globally optimal, however, this segmentation may or may not be the best, where the best segmentation is defined as the segmentation with the maximum possible sum of the edge values within it. Furthermore, this segmentation does not include a large percentage of the possible hypotheses.

3.6.4 Generating a set of representative segmentations

What we would like to do is extract a set of segmentations that provides a representative sample of the space of good segmentations. We also would like to have each hypothesis represented in at least one segmentation within this set. This guarantees that our set of good segmentations contains at least one data point for each interpretation of each region. The hope is that the best interpretation of a region will produce better global segmentations.

Our solution is to run the extraction algorithm on N different graphs, where N is the number of hypotheses in the image. For each hypothesis $h \in H$, we set up a complete hypothesis graph except that we modify it by only assigning the single hypothesis h to the region it represents. This forces each hypothesis to be included in at least one of the set of N segmentations. This set of segmentations may contain up to R duplicates, where R is the number of regions. This is because we can get the same segmentation from any two modified graphs so long as the lone hypothesis is one of the R hypotheses that would be chosen in a segmentation extracted from the complete hypothesis graph.

If we remove the duplicates and rank-order the segmentations according to the sum of the edge values, we end up with a rank-ordered list where each interpretation of each region is contained in at least one of the segmentations. Note, if all discontinuous region pairs have a likelihood of 0.5, there will almost always be multiple equally likely segmentations with the same grouping of regions but different hypotheses for the individual groups. For example, Figure 3.8, which shows the set of final segmentations extracted from the hypothesis graph of the two-sphere image in Figure 3.5, contains four possible segmentations of the two-sphere image. In all four cases each sphere is identified as a coherent surface as shown in Figure 3.9(a). The four segmentations are the possible interpretations of each surface being a disc or a sphere in the scene. Figure 3.9(b) shows the best region grouping for the stop-sign and cup image.

To summarize, the output of the system is a set of rank-ordered segmentations, where each segmentation specifies a hypothesis for each region and how those hypotheses group together. For most of the results shown herein we will show only the best segmentation for each image. It is

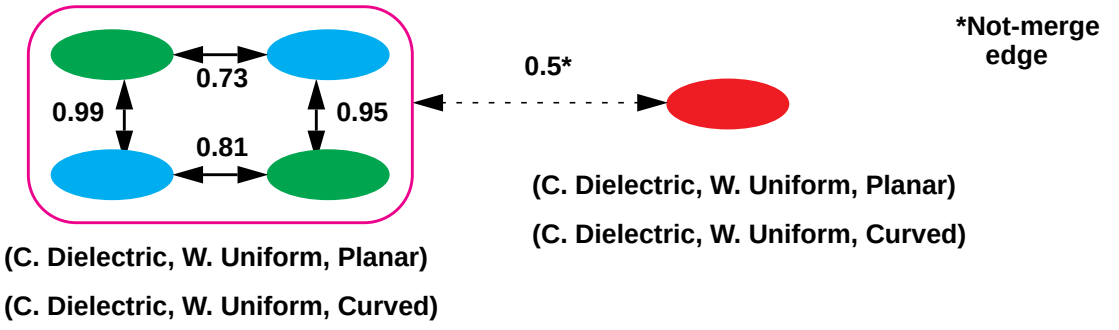
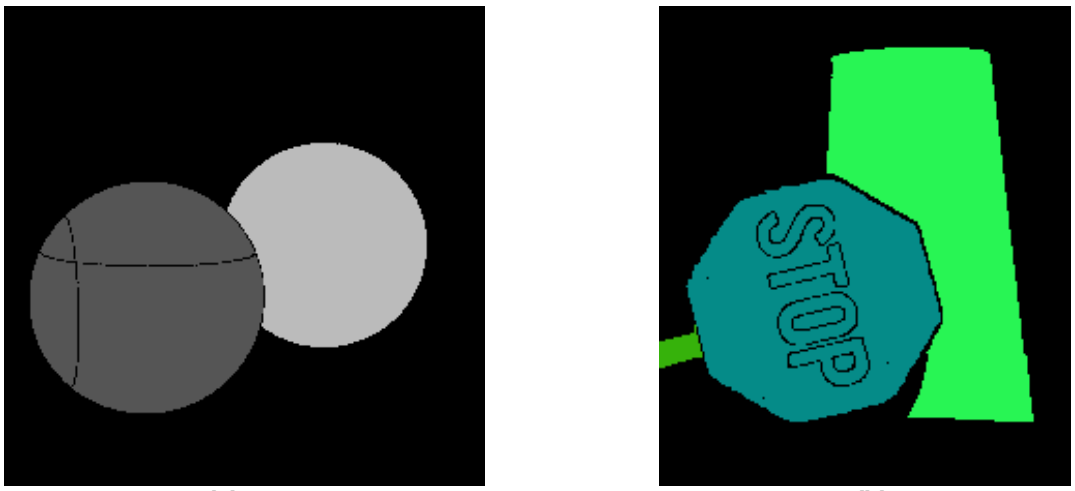


Figure 3.8: The best 4 segmentations of the two-sphere image. The green and blue regions form a coherent surface, discontinuous from the red region. Each region grouping can be either curved or planar.



(a) **(b)**

(size, dark, local NC, global NC) = (100, 6, 0.04, 0.15) **(25, 15, 0.04, 0.15)**

Figure 3.9: Best region groupings for the (a) two-sphere image, and (b) stop-sign and cup image.

important to remember, however, that the complete output of the algorithm is a rank-ordered set.

3.6.5 Avoiding and getting out of local minima

The biggest problem with the highest-probability first algorithm is that it tends to get stuck in local minima. This problem happens particularly in images with loops such as the image of two spheres. For example, if the best edge in pass one connects the curved hypotheses for the left two regions of the left-most sphere, then the algorithm places those two hypotheses in the segmentation map. If, however, in pass two the best edge connects the two planar hypotheses for the right two regions of the sphere, then the algorithm places those two hypotheses in the segmentation map. The only edges between the planar and curved hypotheses, however, are discontinuous edges. This produces a less than optimal segmentation, as shown in Figure 3.10(a).

We use two heuristics to avoid or get out of local minima. The first heuristic is that if there are several edges at the top of the edge list with close to the same likelihood, pick the edge that connects hypotheses that are most like the mandatory hypothesis in the graph. In the case of the two-

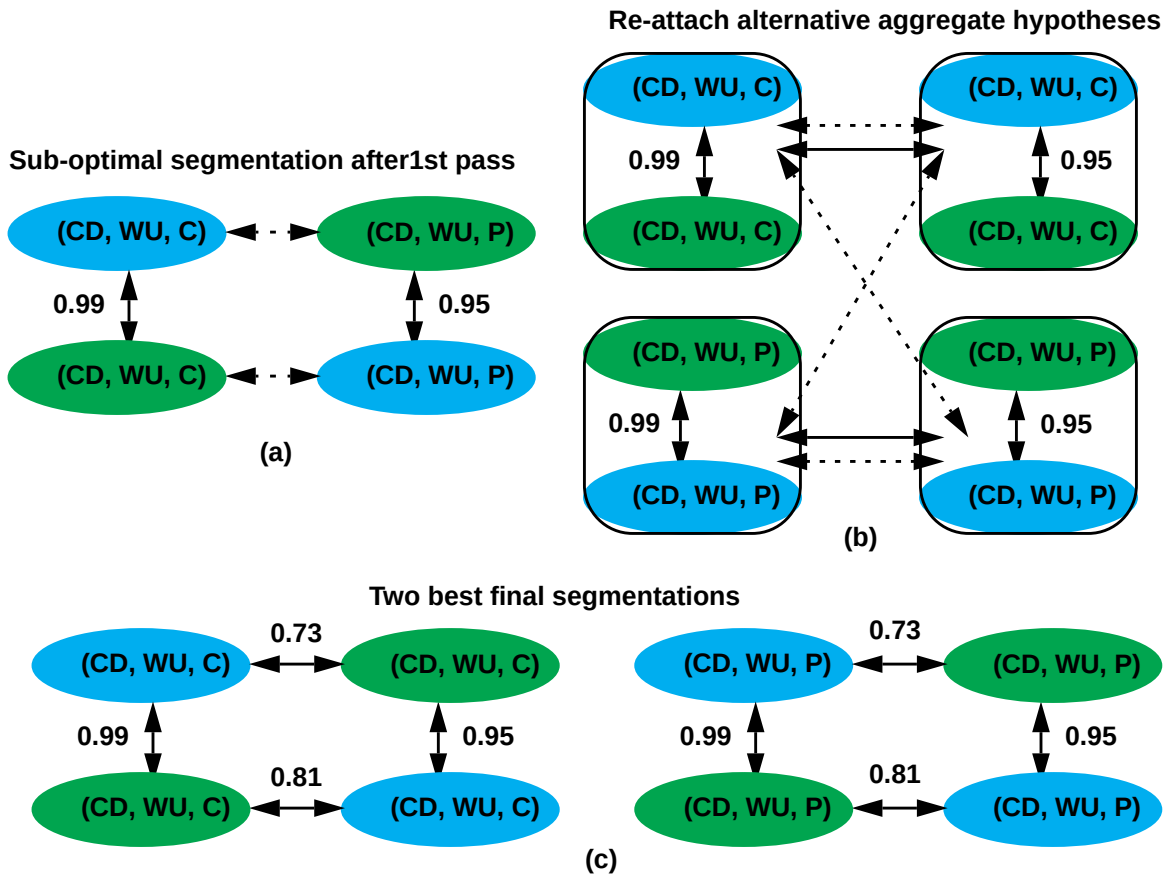


Figure 3.10: (a) Sub-optimal segmentation, (b) hypothesis graph after re-attaching alternative aggregate hypotheses, (c) two best segmentations given (b).

spheres image, if the upper left green region must be (Colored Dielectric, White Uniform Illumination, Curved), then prefer the curved hypotheses for other regions as well so long as the edge values are within epsilon of one another. Alternatively, if the one region must be planar, then prefer the planar hypotheses.

This heuristic works so long as the likelihoods of the different hypothesis pairs have similar values and we want all of the surfaces in an image to be like the mandatory hypothesis. However, it does not work well if there are differences in the likelihoods of the hypothesis pairs for the same two regions. In this case, we need a different approach as the first heuristic only tries to avoid local minima, not get out of them.

To get out of local minima somehow we have to perturb things and restart the search process, which motivates the second heuristic. First, run the algorithm to get an initial segmentation, or set of hypothesis groupings. These hypothesis groupings are also region groupings and tend to contain most of the strong edges in the hypothesis graph. The heuristic is based upon the observation that there are other, almost as likely interpretations for the aggregate regions. For example, in the two-spheres image the interpretation of the left sphere as four hypotheses proposing a curved surface has a similar likelihood as the interpretation of it as four hypotheses proposing a planar surface. However, any mixture of the planar and curved hypotheses is much less likely. In Figure 3.10(a), for example, there is a planar interpretation of the left side of the sphere that is equally as

likely as the curved interpretation. The same situation exists for the right side of the sphere. If we re-attach these alternative aggregate intepretations, then we have the new hypothesis graph shown in Figure 3.10(b). This perturbs the algorithm out of the local minimum and gives it larger groupings of hypotheses to work with. Now, when we run the modified highest-probability first algorithm again, we get one of the final segmentations shown in Figure 3.10(c). Both of these segmentations are better than the first segmentation shown in Figure 3.10(c).

It turns out that a third pass is superfluous for these cases. For all of the test images, the second pass found the optimal segmentation in the judgement of human observers.

This heuristic works for the two-layer hypothesis graphs because there will always be two possible interpretations for each region grouping, except the region grouping containing the mandatory hypothesis for that graph. When we begin to add more hypotheses per region, however, there will be more interpretations for the aggregate regions. In this case, we will need to enumerate the possible alternative interpretations for each aggregate region. This may, in itself, be a somewhat complex problem which we leave for future research.

It is interesting to note the approximate run-time of the modified highest-probability first algorithm in comparison to the complexity of the segmentation space. The complexity depends on the number of regions R, the number of hypotheses per region N_r, the number of adjacent region pairs A_r, and the total number of edges in the hypothesis graph E. If we assume the edge list is sorted on every pass through the main loop of the algorithm, then a single loop takes at most ElogE time to sort the edge list, linear time to update the segmentation map, and linear time to create the new edges. The algorithm goes through the main loop A_r times to fill the segmentation map, bounding the run-time by $O(A_r ElogE)$. If we run the extraction algorithm for each of the RN_r hypotheses, the total run time to extract the representative set of good segmentations is $O(RN_r A_r ElogE)$. Since R, A_r and E are related, as are N_r and E, the segmentation extraction algorithm is essentially polynomial in the number of regions and the number of hypotheses per region. However, this is significantly better than having to search the entire space of segmentations.

3.7. Summary

What this system represents is a specific implementation of the theory and framework laid out in Chapter 2. The system contains five basic sections. First, it finds an initial segmentation of the image using only color features. Second, it attaches a list of hypotheses to these initial regions. The list for this implementation represents a subset of the fundamental hypotheses identified in Chapter 2. Third, the system analyzes and compares these hypotheses for similarity, producing a likelihood of merging for each adjacent hypothesis pair. In the fourth step, the system uses this local information to generate a hypothesis graph, which represents the space of segmentations. Each point in the space is a different segmentation, and each segmentation possesses a measure of its likelihood. This measure depends upon which edges the particular segmentation contains. In the final step the system searches the space of likelihoods using a variant of a step-wise optimal algorithm to find a set of good segmentations.

The following figures show the best region groupings for a set of eight test images. For each of these figures, each region grouping has one explanation consisting of planar hypotheses and one consisting of curved hypotheses. Below each figure are the thresholds we used for each particular

image. Note that the only ones that change are the dark threshold and the region size threshold. Also listed with each figure are the number of initial regions.

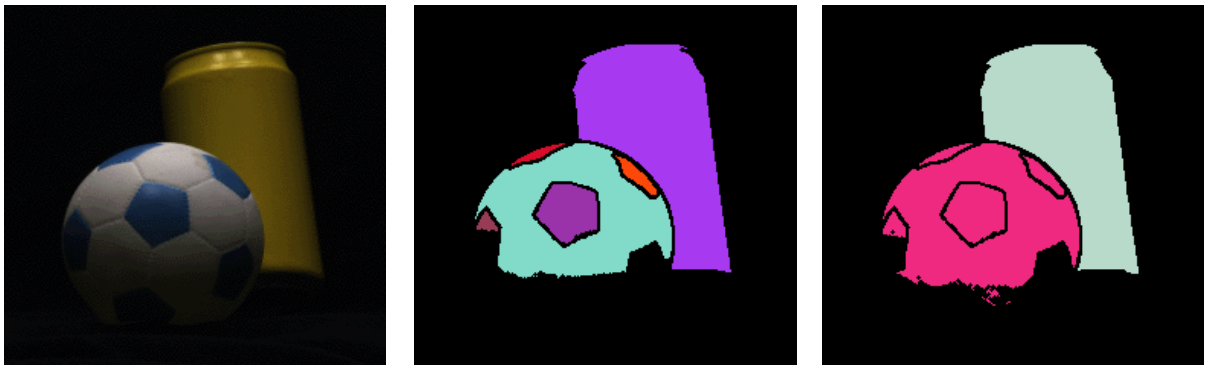
For all of the images, the complete segmentation time from the raw image data to the complete set of final segmentations takes on the order of minutes on a 66Mhz PowerPC. The most costly portion of the algorithm is the profile analysis, for which the system has to fit many polynomials to a large numbers of points. Given that little attempt was made to make the current system efficient, however, it is reasonable to expect that the run-time for any image in this test set could be reduced to under a minute from start to finish.

In all of the eight images the system successfully finds region groupings that correspond to the objects in the scene. However, the results presented in this chapter have one problem: they do not differentiate between the different interpretations of the region groupings. The ball and cylinder in Figure 3.11(a), for example, have the same chance of being planar objects as being curved. Intuitively, it seems that we can do better.

As noted previously, we can do better if we begin to rank-order the various hypotheses according to their likelihood given the image data. For example, since the stop-sign in Figure 3.9(b) has approximately uniform intensities, a hypothesis specifying a planar surface seems a more likely explanation it than a hypothesis specifying a curved surface. We explore this possibility in Chapter 7. Chapter 8 then presents how we can expand the initial hypothesis list to allow the system to handle images containing more complexity.

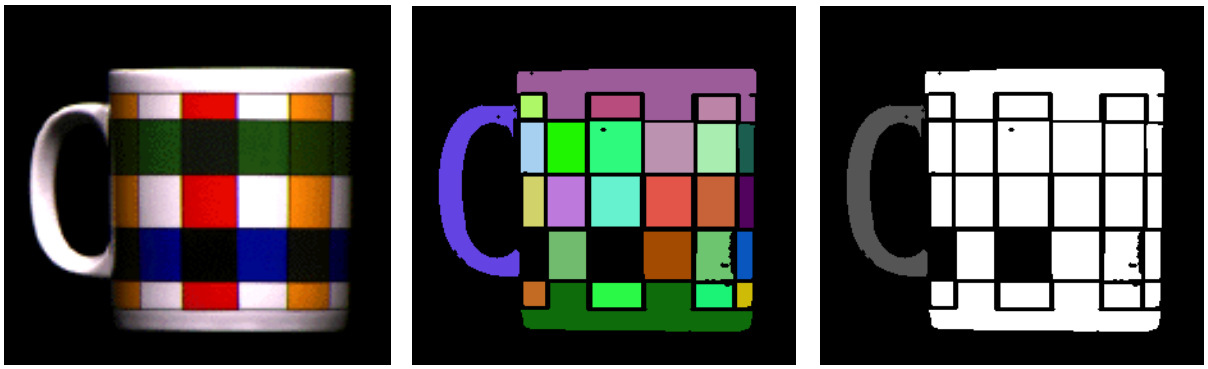
Prior to that, Chapters 4 and 5 probe deeper into the question of testing the compatibility of adjacent hypotheses. Chapter 5, in particular, presents an in-depth look at the compatibility tests the system used to obtain the results presented here. These compatibility tests provide fairly accurate estimates of whether two image regions are part of the same object, and they should find application more generally in the field of vision.

Chapter 6 then discusses how one image in particular, the stop-sign and cup image, made life difficult, but proved to be an excellent test image for this system.



(size, dark, local NC, global NC) = (100, 8, 0.04, 0.15)

(a)



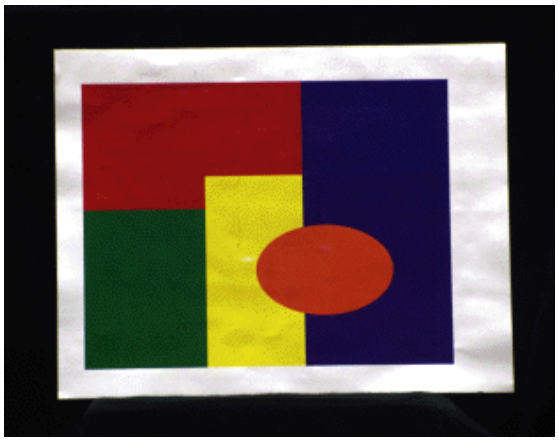
(100, 6, 0.04, 0.15)

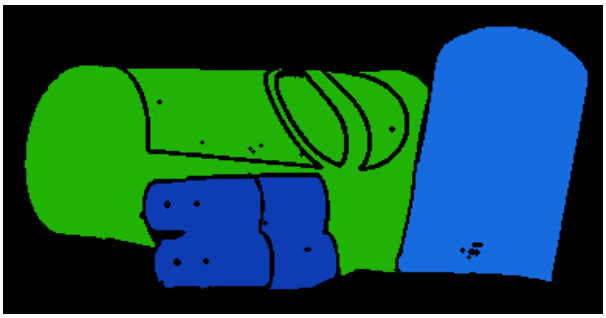
(b)

(100, 12, 0.04, 0.15)

(c)

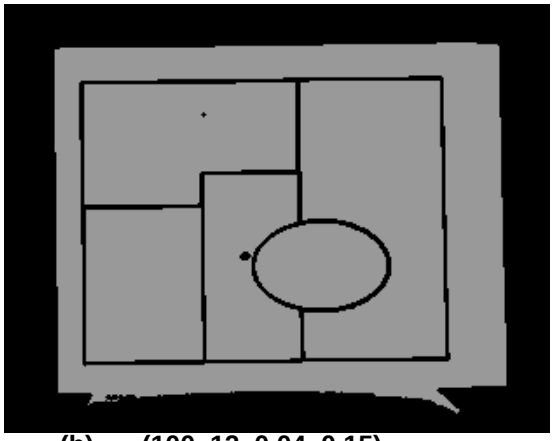
Figure 3.11: (a) ball-cylinder image, initial segmentation, and best region groupings, (b) mug image, initial segmentation, and best region groupings, (c) cup-plane image, initial segmentation, and best region groupings.



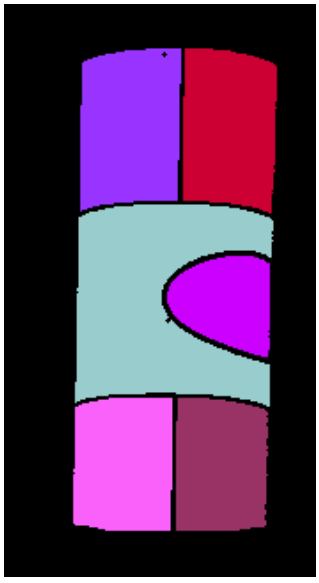


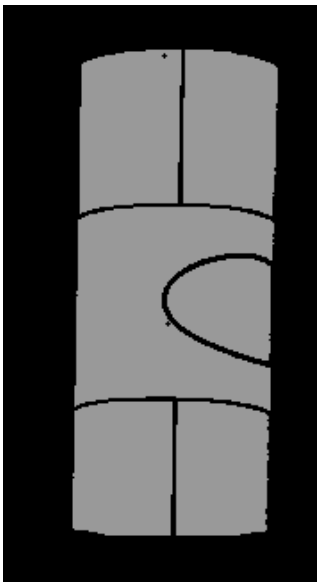
(100, 12, 0.04, 0.15)

(a)



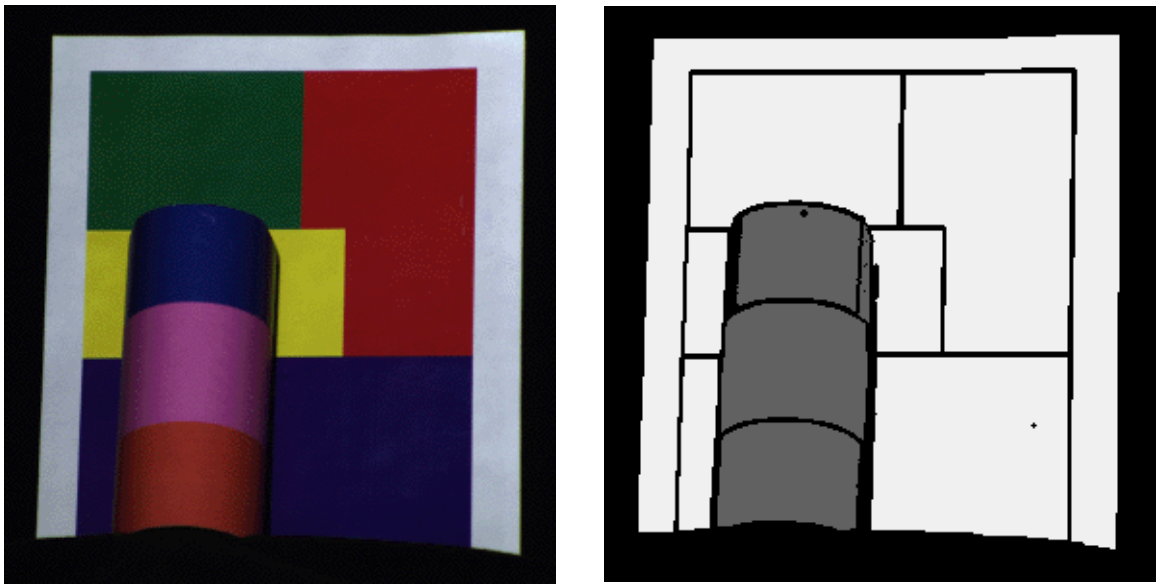
(b) (100, 12, 0.04, 0.15)





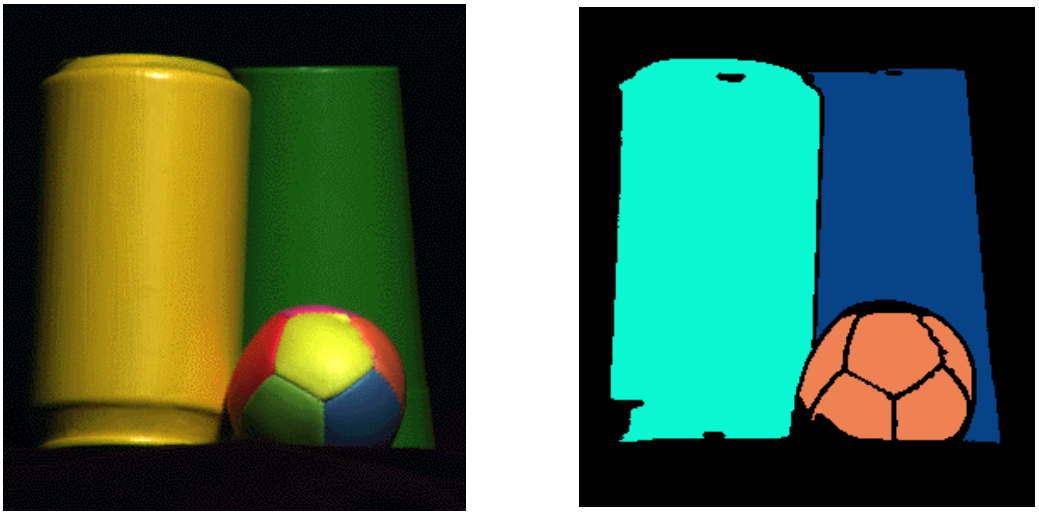
(c) (100, 12, 0.04, 0.15)

**Figure 3.12: (a) pepsi image and best region groupings,
(b) plane image and best region groupings,
(c) cylinder image, initial segmentation, and best region groupings.**



(100, 12, 0.04, 0.15)

(a)



(100, 12, 0.04, 0.15)

(b)

Figure 3.13: (a) plane-cylinder image and best region groupings, (b) two-cylinders image and best region groupings.

Chapter 4: Direct hypothesis compatibility testing

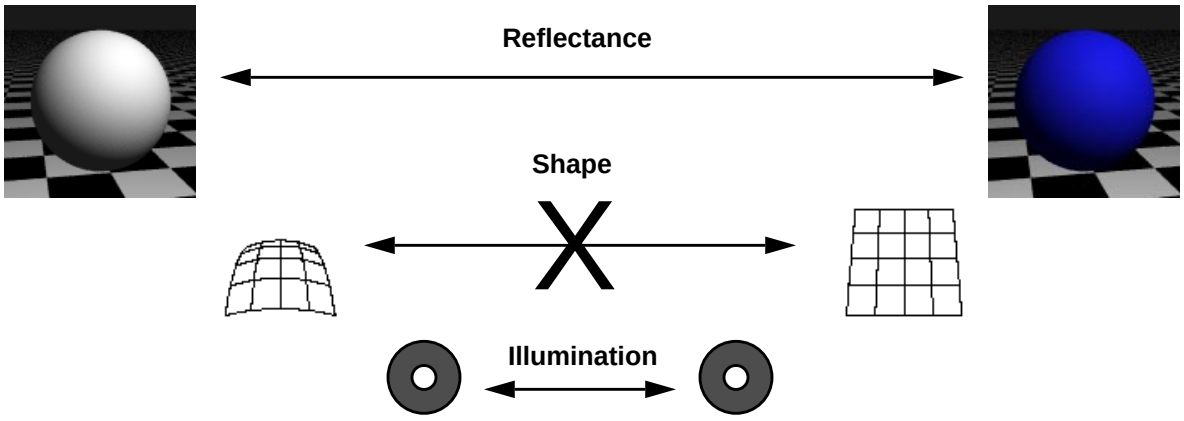


Figure 4.1: Visualization of direct hypothesis compatibility testing. In this case the shapes do not match though the illumination and material type (not color) do.

As discussed in Chapter 3, after finding the initial regions and specifying their hypothesis lists, the algorithm must determine which hypotheses of neighboring regions are compatible. This chapter explores one of the two approaches to answering the question, are these two regions part of the same surface?

This approach we call *direct instantiation*, or *direct hypothesis compatibility testing*. This method works with estimates and representations of the specific shape, illumination environment, and transfer function of each hypothesis. It estimates the similarity of two adjacent hypotheses by directly comparing their representations. In other words, direct hypothesis compatibility testing compares the intrinsic characteristics of the regions. This implies that we have to instantiate, or find estimates of all of the hypothesis elements prior to comparing them.

The hypotheses themselves provide broad direction for the region analysis. For example, if two adjacent hypotheses both specify colored plastic under white uniform illumination then the shape and illumination could potentially contain discontinuities. As in Figure 4.1 the surface representation of one hypothesis could be planar while the other is curved in an incompatible manner. Likewise, if the illumination environment does not change smoothly between the two regions then, while the surfaces may have compatible contours, there may be a depth discontinuity or other change between the two hypotheses indicating they are not part of the same surface.

With a set of rules specifying what constitutes compatibility we can make a compatibility decision. Note, however, that we can still make an incorrect decision. First, there may be inaccuracies in the hypothesis element instantiations. Second, since the mapping from the world to an image is

many to one, the particular instantiations the algorithm compares may not reflect the real world. However, if the instantiations are correct, then direct instantiation should give us the correct compatibility answer.

For the exploration of direct compatibility testing we considered only the hypothesis (Colored Dielectric, White Uniform, Curved) for colored regions and (White Dielectric, White Uniform, Curved) for white regions. To fit with the assumptions of shape-from-shading algorithms, we also assumed each region was lit by a single source. Note that this is different than assuming a single light source for the entire scene. Given these constraints, we could use existing tools for illuminant direction estimation to obtain an estimate of the illumination environment and shape-from-shading to obtain a relative depth map for each hypothesis.

The next two sections discuss the illumination direction estimation and shape-from-shading tools we implemented. Section 4.3 then presents a set of rules and tests that use this information to identify hypothesis compatibility. Finally, section 4.4 summarizes the results and discusses the strengths and weaknesses of this approach.

4.1. Illuminant direction estimation

Illuminant direction estimation [IDE] techniques grew out of a desire to make shape-from-shading [SFS] a more automatic process. Before IDE techniques, the user had to provide an SFS algorithm with the light source direction. Using IDE takes the user out of the loop, although the process then depends on the quality of the IDE technique.

IDE methods typically represent the illumination as two angles: the tilt τ and the slant σ. The tilt is defined as the angle the illuminant direction vector L makes with the x-z plane. The slant is the angle between L and the z-axis. Looking at an image, therefore, if (x,y) = (0,0) is located in the center of the image, then the tilt specifies the orientation of L with respect to the x-axis, and the slant specifies the distance away from the center of the image.

Trying to find the light source direction from a single image is an ill-posed problem. To solve it we need either knowledge of or strong assumptions about the surfaces or distribution of surfaces and their material properties. Having only a single intensity image forces us to make assumptions. A number of approaches, for example, assume that the image contains spherical objects, or a balanced distribution of surface normals, and that the objects display Lambertian reflection. How well an image fits these assumptions strongly affects the performance of IDE techniques. What this implies is that a general purpose IDE method for intensity images does not exist.

That said, there are several different IDE techniques available. Pentland was the first to propose an automatic IDE method [49]. It uses the derivative of image intensity along various directions to estimate the illuminant direction. The assumption is that there is a balanced distribution of surface normals in the image. This implies that imbalances in the distribution of the directional derivatives of image intensity are related to the light source direction.

Lee & Rosenfeld, on the other hand, base their method on the assumption that surface patches in the image are locally spherical [32]. They derived formulas for the illuminant direction angles using a probability density function for a Gaussian sphere and the expected values of the square of the image intensity, the image intensity, and its directional derivatives. In addition to the spherical patch assumption, they assume that the probability density function reflects the distribution of

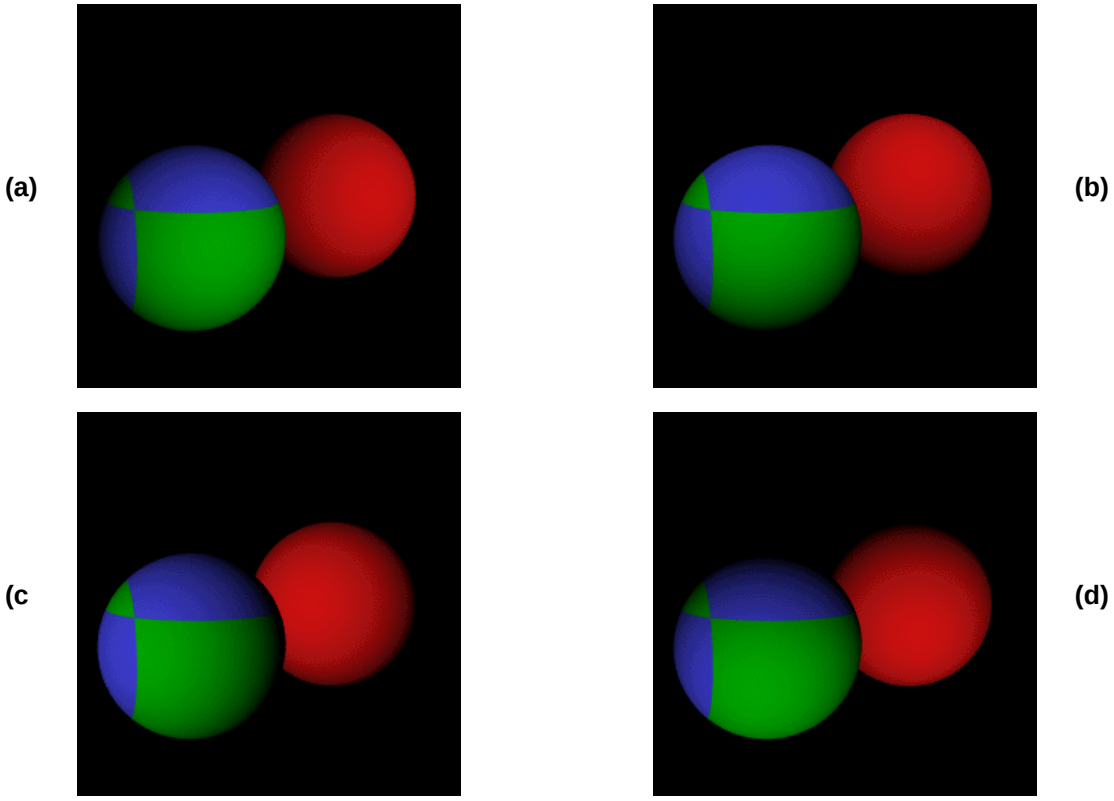


Figure 4.2: Four of the synthetic test images. (a) two-spheres 0-20, (b) two-spheres 90-20, (c) two-spheres 180-20, and (d) two-spheres -90-20.

patch orientations.

Zheng & Chellappa also assume that a spherical patch closely approximates the neighbors of any point in an image [63]. They use a local voting method to calculate the azimuth angle, or the tilt, and then derive a relationship between the first two moments of image intensity and the slant and albedo of the object. Based on the results in Zheng & Chellappa's paper, who directly compared their method to the other two, we decided to use their IDE approach.

To test the algorithm we used multiple versions of the two-sphere image with different light source locations. Figure 4.2 shows four of the test images. Table 4.1 gives the results of the IDE when applied to the entire image and to each region separately. Note that the two numbers at the end of the name of each test image indicate the light source position. For example, two-spheres 0-0 specifies that the light source is at tilt 0 and slant 0 . This makes the light source and viewing directions the same. Likewise, two-spheres 90-10 specifies that the light source is at tilt 90 and slant 10 , indicating the light source is slightly above the viewer.

The two-spheres image is almost a perfect image for Zheng &Chellappa's IDE except for two problems. First, the green and blue sphere occludes a portion of the red sphere. Second, the image contains regions with differing albedos, breaking the IDE assumption that the image contains only objects with the same albedo. As discussed below, the results clearly reflect the effect of the occlusion. The effect of the differing albedos, however, is more difficult to interpret. As the tilt calcula-

Table 4.1: Illuminant direction estimation results for the entire image and for each region individually. Angles shown are the estimated tilt and slant, respectively.

Image	IDE Image	IDE Region 1	IDE Region 2	IDE Region 3	IDE Region 4	IDE Region 5
two-spheres (0,0)	(-154 , 0)	(-90 , 0)	(-102 , 0)	(-38 , 0)	(7 , 0)	(134 , 0)
two-spheres (0,5)	(-75 , 0)	(-17 , 0)	(-97 , 0)	(-36 , 0)	(6.8 , 0)	(124 , 0)
two-spheres (0,10)	(-21 , 0)	(-8 , 0)	(-92 , 0)	(-34 , 0)	(7 , 0)	(110 , 0)
two-spheres (0,15)	(-12 , 0)	(-5 , 0)	(-87 , 0)	(-33 , 0)	(6 , 0)	(95 , 0)
two-spheres (0,20)	(-7 , 0)	(-3 , 0)	(-82 , 0)	(-32 , 20)	(-6 , 11)	(80 , 0)
two-spheres (0,25)	(-5 , 4)	(-2 , 10)	(-77 , 0)	(-31 , 25)	(6 , 22)	(64 , 0)
two-spheres (0,30)	(-4 , 16)	(-1 , 18)	(-72 , 3)	(-29 , 30)	(52 , 0)	(5.7 , 27)
two-spheres (0,35)	(-3 , 22)	(-0.7 , 24)	(-68 , 14)	(-28 , 30)	(43 , 0)	(5 , 31)
two-spheres (90,5)	(150 , 0)	(88 , 0)	(-103 , 0)	(-35 , 0)	(11 , 0)	(126 , 0)
two-spheres (90,10)	(120 , 0)	(88 , 0)	(-104 , 0)	(-33 , 0)	(15 , 0)	(120 , 0)
two-spheres (90,20)	(104 , 20)	(90 , 0)	(-109 , 0)	(-27 , 0)	(23 , 0)	(113 , 21)
two-spheres (90,35)	(97 , 31)	(90 , 19)	(-141 , 0)	(-14 , 0)	(34 , 18)	(108 , 33)
two-spheres (180,5)	(-166 , 0)	(-162 , 0)	(-108 , 7)	(-39 , 0)	(7 , 0)	(142 , 0)
two-spheres (180,10)	(-171 , 4)	(-170 , 0)	(-111 , 16)	(-41 , 0)	(8 , 0)	(148 , 7)
two-spheres (180,20)	(-175 , 22)	(-174 , 0)	(-121 , 26)	(-47 , 0)	(10 , 0)	(155 , 24)
two-spheres (180,35)	(-176 , 34)	(-176 , 24)	(-135 , 34)	(-67 , 0)	(32 , 0)	(162 , 34)
two-spheres (-90,5)	(-122 , 0)	(-90 , 0)	(-101 , 14)	(-40 , 0)	(3 , 0)	(146 , 0)
two-spheres (-90,10)	(-110 , 0)	(-91 , 0)	(-100 , 22)	(-41 , 0)	(-1 , 0)	(161 , 0)
two-spheres (-90,20)	(-102 , 13)	(-91 , 9)	(-99 , 31)	(-46 , 3)	(-11 , 0)	(-159 , 0)
two-spheres (-90,35)	(-97 , 27)	(-93 , 24)	(-97 , 37)	(-53 , 28)	(-26 , 0)	(-122 , 0)

tion is a local voting method, having piece-wise uniform regions should not strongly affect the tilt angle. On the other hand, the relative distribution of dark and light albedo regions may affect the slant estimation as it depends upon the expected intensity value of the image.

When the algorithm uses the entire image to obtain its estimates it obtains reasonable results for most cases. When the light source moves to the right, cases 0-5 through 0-35, the tilt estimate improves as the actual slant increases. This relationship occurs because as the light source moves right, there is more evidence for the calculations of the tilt estimator. Also, for this case, as the angle of the light source increases the occluded regions of the red sphere have less influence on the tilt calculation. In fact, this pattern occurs for each light source motion; the larger the actual slant angle gets, the more accurate the tilt estimation becomes.

The slant estimation shows a similar pattern, getting more accurate as the actual slant increases. In part this is because of the dark threshold. The slant depends upon the distribution of intensities in the image, and the darkest pixels have the greatest influence. For the results in Table 4.1 we used a dark threshold of only 5 pixel intensities. A larger threshold strongly decreased the slant esti-

mates. In all except the 0 tilt case, the results for the 20 and 35 actual slant angles are pretty good. The 0 tilt case reflects the effect of the occlusion, which hides a portion of the dark pixels as the light source moves right.

For the 5 and 10 slant angles, however, except for the two-spheres 180-10 image the algorithm is unable to differentiate them from a 0 slant angle. This is due in part to the dark threshold and in part to the occlusion, which covers some of the darkest pixels in the image. Even dropping the dark threshold to 1 pixel value, which causes problems for the normalized color initial segmentation, the algorithm cannot distinguish between the 0 slant and 5 slant cases. Overall, however, the IDE on the entire image is good enough to obtain reasonable SFS results.

On the other hand, when the algorithm uses only the individual regions, the results are sporadic and depend upon the particulars of the region. Region 1, for example, is almost an ideal case for IDE: a single sphere. The tilt estimates for region 1 are within 10, and most are within 5, except for the two-spheres 0-5 case where the occlusion strongly affects the results. However, even the small occlusion has a strong effect on the slant estimate. Because the occlusion removes a sizable percentage of the darkest pixels, which have the most influence on the slant calculation, the slant estimates are all significantly smaller than the actual angles.

Regions 2-5, on the other hand, are anything but ideal. Given that they only cover a portion of a sphere, they immediately break the assumption of a good distribution of surface orientations. The tilt estimates each reflect the particular bias of the region. For example, region 2, on the north side of the sphere, says the illumination always comes from the south. Region 3, however, on the northwest side of the sphere, says the illumination always comes from the southeast. We can see similar patterns for regions 4 and 5. The conclusion from the data is that we cannot accurately estimate the tilt from individual regions, especially in an image containing occlusions or multi-colored objects.

The slant estimates, like the tilt, also depend heavily on the orientation of the region. The slant estimates are reasonable when the region is on the dark side of the object as the light source moves away. For example, region 3 gets the correct slant angles for two spheres 0-20, 0-25, and 0-30. However, if the light source is moving in the direction of the region, then there are no dark pixels and the slant estimates all remain at 0. Again, we have to conclude that we cannot accurately estimate the slant from individual regions.

A valid question is why not use the results of IDE on the entire image? Why do we have to analyze each region independently? If we limit the segmentation algorithm to images under a single illuminant, for example, then we can use the IDE results from the entire image for all regions and their hypotheses.

However, for the framework to be more general, we have to apply IDE to each region individually. To move beyond the simplest cases we cannot make assumptions about the global scene illumination. Only the broad classes of the individual hypotheses limit the form of the illumination, and only for the region they specifically explain. This exploration of direct compatibility testing aims not at finding out whether IDE and SFS are useful for an extremely limited scenario, but at how effective direct compatibility testing is within the more general framework for segmentation.

4.2. Shape-from-shading

Shape-from-shading [SFS] algorithms fall into three categories: global minimization, global propagation, and local methods. Global minimization techniques use an error function based on the reconstructed appearance of the scene and constraints such as smoothness in the gradient or surface normal. Using gradient descent or some form of relaxation, they attempt to minimize the error function while conforming to the specific constraints of the technique. Global minimization techniques often require an initial estimate of the depth map and do not run quickly. Examples of global minimization include the work of Ikeuchi & Horn and Brooks & Horn [21] [9].

Global propagation techniques, on the other hand, use singularity points and occluding boundaries to get surface gradient estimates for a few points in the scene. Then they propagate this information throughout the image, again using smoothness or integrability constraints to control the process. Global propagation is not in general as accurate as global minimization, but techniques that use global propagation run faster and require initial estimates for only a few well-defined points. Some global propagation algorithms include the characteristic strip method of Horn, and the minimum downhill approach of Bichsel & Pentland [19][4].

Finally, local techniques make assumptions about the surfaces they work with and use these assumptions to calculate surface gradients based on only a local area of image intensities. Local techniques include two approaches by Pentland; one assumes that the surfaces are locally planar, the other that they are local spherical [47][48]. Local methods are the fastest of the three approaches, but give the worst results in most cases because of the restrictive assumptions about the surfaces.

To directly instantiate the shape and illumination of the hypotheses, we implemented Bichsel & Pentland's SFS algorithm [4]. We chose this method because it is a global propagation method, and, according to the survey by Zhang *et. al.*, is the best method when the illumination comes from the side [62]. This is an important characteristic given that the algorithm often examines regions of objects that are oblique to the illuminant. The global propagation methods also appeared, at the time, to offer the most accurate depth maps with the fewest assumptions. In retrospect, a local method might have been better because small regions of an object are not guaranteed to contain good initial starting points.

Bichsel & Pentland's method works by calculating gradients and assign initial depths for singular points, or maxima of the image intensity. Given knowledge about the illuminant direction, the algorithm can calculate the gradients for these points. The algorithm then propagates the information globally through the image in an iterative fashion, using the gradient of the image intensity to constrain the propagation of the surface gradients and depth.

For this test, we represent the shape as a depth map. Figure 4.3 shows the SFS results for the regions in the two-spheres 0-0 test image when the illuminant and viewing directions are the same. Figure 4.4 shows the SFS results on the four test images from Figure 4.2. All of these results are completely automatic, since the illuminant direction estimator finds the direction of the illumination independently for each region.

4.3. Comparing the intrinsic characteristics

After obtaining estimates of the shape and illuminant direction we must compare the hypothesis

Figure 4.3: Border errors and depth map for the two-spheres 0-0 image. For the border errors, darker pixels show larger errors, and the blue pixels indicate no adjacent region. The numbers are the sum-squared error along the border. For the depth map, dark pixels are further away. For this image the depth values ranged from [-78, 55], or [0, 133].

elements of adjacent pairs. We use a two-step algorithm to compare their shapes. First, the algorithm finds the optimal offset, in a least-squares sense, of the two regions by comparing their depth values along the border and minimizing the square of the error between them. Equation (1) gives optimal offset Δ_{ij} in terms of the region border height values h_{1i} and h_{2j}, and the number of border pairs N.

$$\Delta_{12} = \frac{\sum_N h_{1i} - \sum_N h_{2j}}{N} \tag{1}$$

Second, the algorithm uses the optimal offset to find the minimum sum-squared error of neighboring border pixels as described by (2).

$$SSE = \sum_{i=1}^{N} \left(h_{1i} - \left(h_{2i} + \Delta_{12}\right)\right)^2 \tag{2}$$

By then dividing by the number of border pixels less one, as in (3), the algorithm gets the sample variance of the height difference of border point pairs along the boundary [29].

$$\sigma_{12}^2 = \frac{SSE}{N-1} \tag{3}$$

Given the variance of the difference in the depths of adjacent border pixels, we now want to quantify it for each region pair. The variance should be larger for adjacent region pairs with differing shape, and smaller for regions whose shapes match. Somehow we have to convert this intuition into a measure of similarity.

First, we select a threshold variance for the surface depths. Based on the range of depth values returned by B&P's algorithm, given in Figure 4.3, we used a variance of 64 depth values, which corresponds to a standard deviation of 8 depth values. If two regions' shapes match, then the calculated variance in the border pixel depths should be close to or less than the threshold variance. We can compare the variances using a chi-square test [29]. The chi-square test returns a likelihood

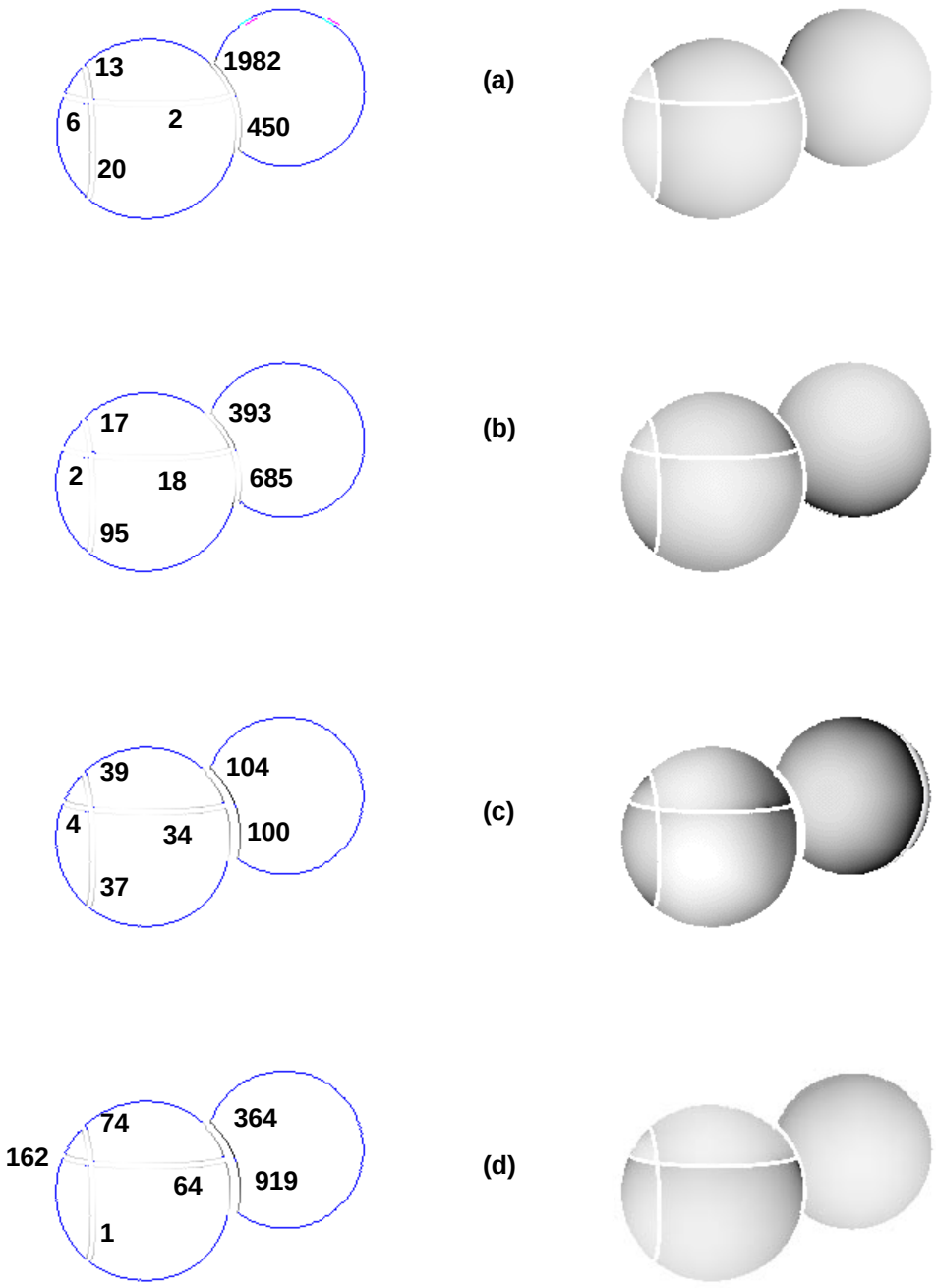


Figure 4.4: Border depth errors and depth maps for (a) two-spheres 0-20, (b) two-spheres 90-20, (c) two-spheres 180-20, and (d) two-spheres -90-20. On the border error displays, larger border errors show up as darker points, and the blue points show borders with no adjacent regions. The numbers indicate the sum-squared error along the border. For the depth map, lighter points are closer, darker points are further away.

Table 4.2: Results of shape and illumination comparisons for all region pairs. The cells indicate whether the combination of the illumination and shape comparisons are less than 0.5 (discontinuity) or greater than 0.5 (merge).

Image	SSE / P(z Z) Regions 1-2	SSE / P(z Z) Regions 1-5	SSE / P(z Z) Regions 2-3	SSE / P(z Z) Regions 3-4	SSE / P(z Z) Regions 4-5	SSE / P(z Z) Regions 2-5
two-spheres (0,0)	discontinuity	discontinuity	merge	merge	merge	merge
two-spheres (0,5)	discontinuity	discontinuity	merge	merge	merge	merge
two-spheres (0,10)	discontinuity	discontinuity	merge	merge	merge	merge
two-spheres (0,15)	discontinuity	discontinuity	merge	merge	merge	merge
two-spheres (0,20)	discontinuity	discontinuity	merge	merge	merge	merge
two-spheres (0,25)	merge	discontinuity	discontinuity	merge	merge	merge
two-spheres (0,30)	merge	discontinuity	discontinuity	discontinuity	discontinuity	discontinuity
two-spheres (0,35)	merge	discontinuity	discontinuity	discontinuity	discontinuity	discontinuity
two-spheres (90,5)	discontinuity	discontinuity	discontinuity	merge	merge	merge
two-spheres (90,10)	discontinuity	discontinuity	merge	merge	merge	merge
two-spheres (90,20)	discontinuity	discontinuity	merge	merge	discontinuity	merge
two-spheres (90,35)	discontinuity	discontinuity	merge	merge	discontinuity	discontinuity
two-spheres (180,5)	discontinuity	discontinuity	merge	merge	merge	merge
two-spheres (180,10)	discontinuity	discontinuity	merge	merge	merge	merge
two-spheres (180,20)	discontinuity	discontinuity	discontinuity	merge	merge	discontinuity
two-spheres (180,35)	discontinuity	discontinuity	discontinuity	merge	discontinuity	discontinuity
two-spheres (-90,5)	discontinuity	discontinuity	merge	merge	merge	merge
two-spheres (-90,10)	discontinuity	discontinuity	discontinuity	discontinuity	merge	discontinuity
two-spheres (-90,20)	discontinuity	discontinuity	discontinuity	discontinuity	merge	discontinuity
two-spheres (-90,35)	discontinuity	discontinuity	discontinuity	discontinuity	merge	discontinuity

that a sample of a population, in this case a single border from the population of all matching borders, parameterized by the threshold variance would have a sample variance equal to the calculated one. Since the threshold variance is based upon the noise and random variation in the depth map, the chi-square test result indicates the likelihood that the depth map noise would cause the sample variance.

Ultimately, the chi-square result represents how well the region borders match. For example, if there is a 99% likelihood that the variance, or error, is due to random variation, then there is only a 1% likelihood that the error is due to a discontinuity in the shape of the regions. Figure 4.3 shows the sum squared error for each region pair border in the two-spheres 0-0 test image. The sum-squared error [SSE] is more informative in this case because the results of the chi-square test are likelihoods of 1 for the small errors and 0 for the large errors for a wide range of threshold variances. For this image direct instantiation gives a clear indication of which regions' shapes match.

Table 4.2 shows the merge decision for all of the border pairs for all 20 test cases. To obtain these results, the algorithm first applied IDE to each region and then used the region illuminant direction estimates as the input to Bichsel & Pentland's SFS algorithm. The algorithm found the shapes for each region independently. The cell values in Table 4.2 indicate whether the combination of the shape comparison and illumination comparison indicates a likely merge. The first two columns correspond to the borders between the green-blue sphere and the red sphere. The latter four correspond to the borders between the regions of the green-blue sphere. A good result does not match the red sphere with the green-blue sphere but does match all of the green-blue regions. The two-spheres 0-0 case, for example, is a good result.

Comparing the illumination and transfer functions for this test case is straightforward. The transfer functions are necessarily discontinuous at the borders because of the hypotheses being considered and the initial segmentation method. Therefore, no comparison of the transfer functions needs to be made.

As the illumination should be coherent over both regions, however, we do need to estimate its similarity. One method of comparing the illuminant direction estimates of adjacent regions is to first convert the tilt and slant angles for each region into 3-D illuminant direction vectors and compute the angle between them. Then we can treat this angle as a random variable with a normal distribution, a mean of 0, and a threshold variance. Using this model we can convert the angle difference between the illuminant direction vectors to a likelihood in the range [0,1] indicating the similarity of the two illumination environments. Equation (4) gives the relationship between the standard deviation σ, the angle α, and the likelihood L_I of that angle given that the two regions share the same illumination.

$$L_I = e^{-\frac{\alpha^2}{2\sigma^2}} \tag{4}$$

The algorithm then multiplies this likelihood by the shape-comparison results to obtain a single match estimate [29]. The results in Table 4.2 use a standard deviation of 20 to calculate the probability that the two illuminant directions are the same.

As specified in Table 4.1, for the two-spheres 0-0 image the illuminant directions are all identical. For this case, therefore, the illumination environment comparison does not affect the SFS comparison results shown in Figure 4.3.

In fact, the illuminant direction comparison does not strongly affect the overall results for a majority of the test images. In particular, it does not affect the cases where the slant is small. However, the illuminant direction does affect the compatibility results when the slant gets larger. As is clear from Table 4.2, the combination of the illumination direction comparison and the shape comparison are more likely to provide incorrect results as the slant get large.

4.4. Analysis of results

What this chapter presents is one method of comparing the compatibility of two image regions. For this task we selected particular solutions to particular problems, namely a method for shape-from-shading and one for illuminant direction estimation. Analyzing these particular results we can focus on their faults and characterize when they will work. The previous sections presented

most of this focused analysis. This in itself is useful because it reveals information about how these particular algorithms work and break.

But we are also searching for the answer to a broader question: is direct instantiation a feasible approach to comparing the compatibility of image regions? To answer this question we must extrapolate from the results in Table 4.1 and Table 4.2, and ask ourselves whether they predict anything about other shape-from-shading and illuminant direction estimation techniques. There seem to be three interesting issues to explore. First, is there an inherent difficulty in region-based processing? Put differently, is there a limit to our ability to make good decisions about the intrinsic characteristics of a region using only local information? Second, does direct instantiation make too many decisions too early in the process? Finally, given the performance of these algorithms, deemed to be of high quality by other researchers, is it reasonable to expect good performance on more general real images which magnify the negative effects present in the test images and which add new problems such as noise and camera limitations?

To answer the first question, we look first to the task of illuminant direction estimation. Given that the illuminant direction is a necessary prerequisite for SFS, if we cannot achieve good IDE, then we cannot adequately estimate the shape. As noted by Zheng & Chellappa, IDE is an ill-posed problem and requires strict assumptions about the scene such as locally spherical objects, a balanced distribution of surface orientations, and uniform albedos. In almost any random scene these assumptions will be bent or broken. As shown by the results for the two-spheres image, however, which has a slightly unbalanced distribution of surface orientations and whose objects do not exhibit uniform albedo, IDE does not suffer catastrophic failure when the scene mildly breaks the assumptions. In other words, an entire scene can contain sufficient redundant information that overcomes slight mismatches between the scene and the algorithm's assumptions.

As shown by the IDE results on the individual regions, however, the combination of strongly breaking the algorithm's assumptions and the lack of redundant information do cause the algorithm to fail. The assumptions break because they are geared towards an entire scene, not a portion of an object. To make sufficiently strict assumptions about arbitrarily shaped and sized image regions is a difficult task at best and would strongly limit the complexity of the objects permitted in a scene.

This analysis is strengthened by the SFS results presented by Zhang *et. al.* in their survey paper on SFS techniques [62]. As noted previously, the local SFS methods they examined require strong assumptions about the types of surfaces in the scene. Thus, while they are fast and work on small image regions, they produce the poorest estimates on general test images. Both global methods, propagation and minimization, in general return better results.

The conclusion is that to make a firm estimate of the illumination environment, and therefore the shape, requires some form of global scene analysis. The approach to illuminant estimation and shape-from shading of Breton *et. al.*, for example, is one framework for applying global information to local analysis [5]. They first undertake multiple local SFS analyses for small image regions using multiple light source locations. Then they choose a single analysis from each small region based on how well the patches fit together globally.

To adequately estimate the illumination environment and the shape of particular hypotheses, therefore, probably requires this form of analysis. In other words, each hypothesis for an image region would have to contain multiple estimates of the shape and illumination. Then, our segmen-

tation algorithm not only would have to select a broad class hypotheses for each region, but also the particular shape and illumination instantiations within those hypotheses.

In theory, this is an attractive method. However, in practice it is not currently feasible because a number of the broad classes are high-dimensional. For example, even the uniform illumination environment allows for multiple arbitrarily shaped light sources, ambient lighting, and differently colored illumination. In other words, to adequately cover the space of possible shapes and illumination environments would require an exceptionally large amount of memory and time. While it may be possible to compress this information, that is a question beyond the bounds of this work.

To answer the first question posed at the beginning of this section, our ability to make good decisions about the intrinsic characteristics of a region based on local information is limited and some form of global analysis is necessary.

The answer to the second question, about making decisions too early in the process, relates to the first answer. While it is appropriate to make guesses about the shape and illumination of a region early in the process, making a final decision requires more global information if we want to make good decisions. Both the framework for segmentation presented herein and the work of Breton *et. al.* work on this premise. The process makes final decisions about the intrinsic characteristics of a region as late as possible when it has the most information, and the most global information to work with.

The final question is whether the IDE and SFS methods are flexible enough to work on real images. Simply put, every test image contains something that would cause the analysis to break. The scenes either were taken under multiple light sources, contain interreflection, contain shadows, contain specularities, or contain planar objects. Furthermore, all of them contain some form of noise or texture at a small scale, which strongly affects the results of the Bichsel & Pentland SFS algorithm [62]. These aspects of real scenes not only affect these particular algorithms, but also affect other SFS algorithms which assume Lambertian reflection and single light sources.

The conclusion of this discussion is that direct compatibility testing is difficult at best. Not only are there insufficient tools for analyzing more complex hypotheses, but the need for global analysis greatly increases the resources required for the method to be effective. For the segmentation framework to be effective, it must use a different form of hypothesis compatibility testing.

Chapter 5: Weak hypothesis compatibility testing

An alternative to direct hypotheses compatibility testing is *weak hypothesis compatibility testing*. This method draws upon the idea that it is easier to tell if regions *should not* be connected then if they *are* connected. To accomplish this we search for physical attributes of an object's appearance that have predictable relationships between different regions of the same object. By looking for these predictable relationships we can differentiate between regions that *may be* part of the same object and those that are not. As these physical characteristics are generally local, they are more appropriate for region-based analysis than the aforementioned direct comparison techniques.

Like direct compatibility testing, we cannot guarantee that each compatibility test will correctly identify regions from different objects; the measured physical characteristics may match between two regions that are not part of the same surface.

Our solution to this problem is to test multiple physical characteristics with the expectation that not all of them will match given two regions that are not part of the same surface. The three characteristics we examine are the reflectance ratio of nearby border pixels, the gradient direction of nearby border pixels, and the smoothness of the adjusted intensity values across the region boundaries.

The next three sections present these characteristics and the compatibility tests based upon them. Section 5.4 then talks about how to merge the individual test results and presents and analyzes the results on a series of test images and objects.

5.1. Reflectance ratio

The reflectance ratio is a measure of the difference in transfer function between two pixels that is invariant to illumination and shape so long as the latter two elements are similar. Nayar and Bolle developed the concept and have shown it to be effective for both segmentation and object recognition [41]. The reflectance ratio was originally defined for intensity images and measures the ratio in albedo between two points. The albedo of a point is the percentage of light reflected by the surface.

The principle underlying the reflectance ratio is that two nearby points in an image are likely to be nearby points in the scene. Therefore, they most likely possess similar illumination environments and geometric characteristics as shown in Figure 5.1(a).

Nayar and Bolle represent the intensity I of a pixel in an image as

$$I_i = k\rho_i R(s, v, n) \quad (1)$$

where k represents the sensor response and light source intensity, ρ is the albedo of the surface, and $R(s, v, n)$ is a scattering function representing the geometry dependent aspects of the transfer

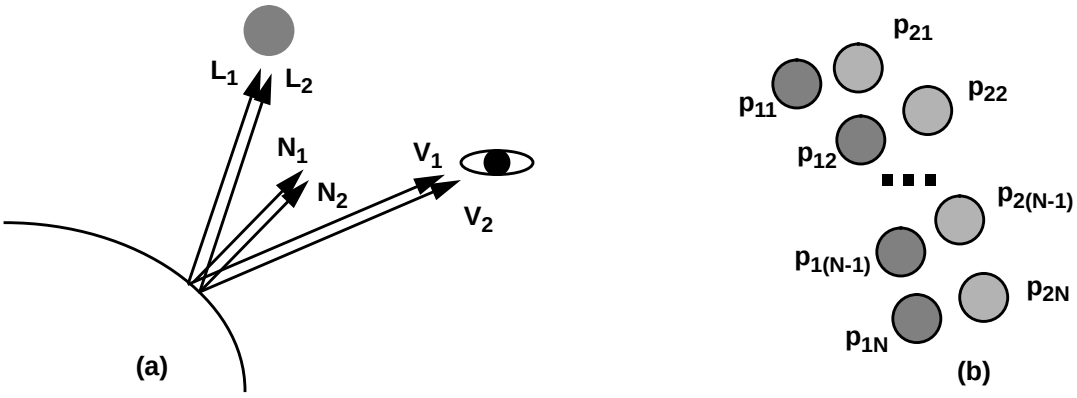


Figure 5.1: (a) Two nearby points. Note that the geometry is approximately the same for both points. The ratio of the intensities is, therefore, a function of the albedos of the two points. (b) Pixel pairs along the border of two regions. Two regions that are part of the same object should have a constant reflectance ratio along their boundary.

function and exitant illumination field. If p_1 and p_2 are nearby points in a scene, then the geometry dependent functions should be similar, as will the light source brightness and sensor response. Therefore, if we take the ratio of the intensities of two nearby pixels in an image, then we obtain the ratio of albedos, or the reflectance ratio π as shown in (2).

$$\pi = \frac{I_1}{I_2} = \frac{k\rho_1 R(s, v, n)}{k\rho_2 R(s, v, n)} = \frac{\rho_1}{\rho_2} \tag{2}$$

Nayar and Bolle go on to show that multiple light sources do not affect the reflectance ratio so long as the assumption of similar geometries and illumination environments holds [41].

Unfortunately, π is unbounded and ranges from 0 to infinity. A well-behaved version of the reflectance ratio is the difference in intensities divided by their sum, as in (3) [41].

$$r = \left(\frac{I_1 - I_2}{I_1 + I_2} \right) \tag{3}$$

Note that the geometry and sensor dependent terms still cancel out. Unlike the ratio in (2), however, this measure of the reflectance ratio ranges from [-1,1].

Returning to the matter at hand, given this measure of the difference between the transfer functions of adjacent pixels, how do we use it to measure hypothesis compatibility? The basic idea is to look for constant reflectance ratios along region boundaries. If the reflectance ratio along the boundary connecting two regions of different intensity is not constant, then either the shape or illumination are incompatible in addition to the transfer function.

The algorithm is as follows. As shown in Figure 5.1(b), for each border pixel p_{1i} in h_1 that borders on h_2 the algorithm finds the nearest pixel p_{2i} in h_2 using the local tangent method described in Chapter 3 and calculates the reflectance ratio between them if both pixels are equal to or brighter than a dark threshold d_{rr}. Since the reflectance ratio is designed to work on intensity images, not color images, the algorithm uses the pixel intensity values as calculated by (4) from the R, G, B values.

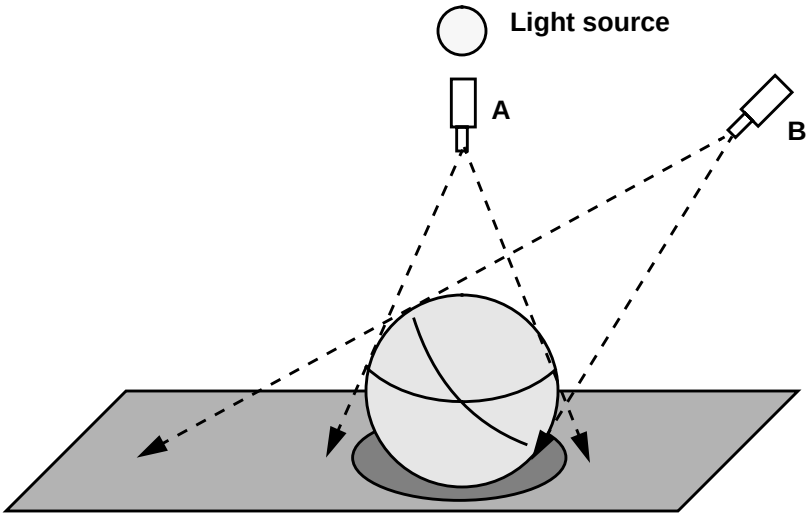


Figure 5.2: Scene of a sphere and a plane lit by a single light source. From position A the boundary of the sphere has constant brightness, as does the plane. From position B, however, the boundary of the sphere has varying brightness and the plane is partially in shadow

$$I_i = \frac{\sqrt{R_i^2 + G_i^2 + B_i^2}}{\sqrt{3}} \tag{4}$$

Summing up the ratios of all of the sufficiently bright pixel pairs the algorithm calculates the mean reflectance ratio r_{avg}. It then finds the variance of the reflectance ratio along the border using

$$Var = \sum_{i=1}^{N} \frac{(r_i - r_{avg})}{N-1} \tag{5}$$

where r_{avg} is the mean reflectance ratio along the border and N is the number of sufficiently bright border pixel pairs.

If the regions belong to the same object, the reflectance ratio should be the same for all pixel pairs (p_{1i},p_{2i}) along the h_1,h_2 border, regardless of the shape or illumination. The variance as calculated above is a simple measure of constancy. If h_1 and h_2 are part of the same object, this variance should be small, although there will be some variation because of the quantization of pixels, noise in the image, and small-scale texture in the scene.

If, however, h_1 and h_2 are not part of the same object, then the illumination and shape are not guaranteed to be similar for each pixel pair, violating the assumption underlying the reflectance ratio. This should result in a larger variance. We can differentiate between these two cases using a threshold variance and chi-square test as we did when comparing the border depths in Chapter 4. Region pairs with a small variance may be compatible hypotheses; region pairs with large variances are not compatible. This test, therefore, will rule out merging some hypothesis pairs and build confidence for merging others.

It is important to note that not all discontinuous surfaces will produce a significant variance in the reflectance ratio. For example, consider a sphere sitting over a plane as in Figure 5.2. If the cam-

Table 5.1: Reflectance Ratio Results for Var_N = 0.008. The last column indicates whether the two regions are possibly compatible.

Region A	Region B	Average Reflectance Ratio	Refl. Ratio Variance	Compatible?
Red region	S region	.4463	.0004	Possibly
Red region	T region	.4449	.0005	Possibly
Red region	O region	.4503	.0004	Possibly
Red region	P region	.4541	.0006	Possibly
Red region	Cup region	.2107	.0125	No
O hole	O region	-.4358	.0008	Possibly
P hole	P region	-.4562	.0004	Possibly
While pole	Red region	.1709	.0710	No

era is in position A, then the border pixel pairs all have similar reflectance ratios as the boundary of the sphere from the camera's viewpoint has constant intensity. From position B, however, the boundary of the sphere varies in intensity and part of the plane is in shadow. Therefore, from position B the reflectance ratio along the boundary of the sphere will vary significantly.

For any two objects and a given scene geometry there are probably camera positions from which the reflectance ratio along the boundary will be constant, and other camera positions from which it will vary. For two regions that are part of the same surface, however, the reflectance ratio along the boundary should be constant independent of the camera position. The expectation is that for most camera positions the reflectance ratio between two different surfaces will vary. For example, for the scene in Figure 5.2 there is only one camera position from which the sphere's boundary has constant intensity, position A. In this position, the two reflectance ratio test will not rule out merging the two regions. For other camera positions the sphere's boundary in the image does not follow an intensity contour, resulting in a large variance and ruling out a merger between the two regions. Note that as the camera approaches position A the variation in the reflectance ratio along the boundary decreases and at some position will fall below any reasonable threshold. Therefore, the space of camera positions from which the boundary will have an approximately constant reflectance ratio is actually a 3D volume. What this means is that we cannot simply dismiss this set of camera positions as insignificant compared to the entire space of camera positions. This in turn backs up the statement that a large reflectance ratio variance allows us to rule out a merger, but a small variance only tells us that a merger is possible.

As noted previously, we can use a chi-square test to compare the sample variance of the reflectance ratio along a border to a threshold variance [29]. The algorithm uses a threshold variance of 0.008, or a standard deviation of 0.09 for all of the test images. This is approximately 4.5% of the range [-1,1].

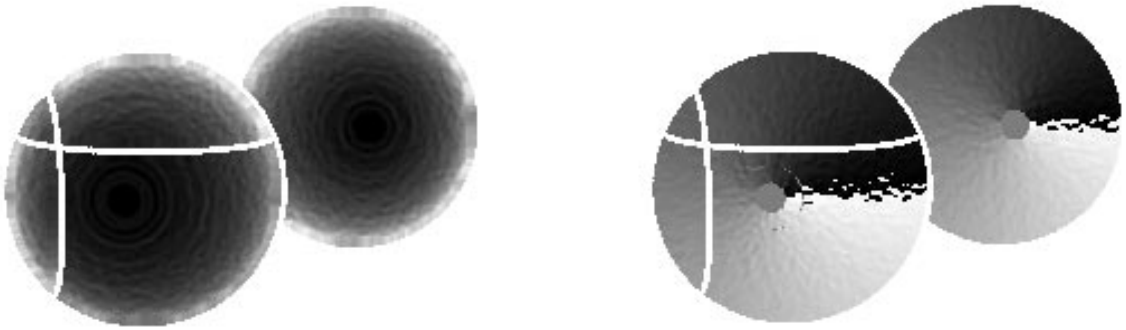


Figure 5.3: (a) Image gradient intensity values for the two-spheres image. Darker pixels represent smaller gradients. (b) Image gradient direction values for the two-spheres image. From darkest to lightest, the intensity values represent from -π to +π radians.

There are two main issues driving the selection of a threshold. The first is the natural variation in the reflectance ratio due to small-scale texture and noise in an image. The second is the quantization effects of using integral pixel values. This becomes especially important as pixels get darker. For example, the variance in the pixel pairs (20, 30) and (21, 30) is much larger than the variance in the pixel pairs (100, 150) and (101, 150). This is the reason the algorithm uses the dark threshold d_{rr} and only considers pixel pairs where both pixels are equal to or brighter than d_{rr}. For all of the test images $d_{rr} = 20$. Given that a typical dark threshold for the initial region growing algorithm is 35, d_{rr} rarely affects the results.

Table 5.1 shows the variances in the border reflectance ratios of the region pairs for the test image of the stop-sign and cup. This example shows an order of magnitude or more difference in the reflectance ratio variances for region pairs that belong to the same object versus region pairs that do not.

5.2. Gradient direction

The direction of the gradient of image intensity is another characteristic that reflects the geometry and illumination of a scene. For piecewise uniform objects, the image gradient direction is invariant to the transfer function except at region boundaries. Therefore, the similarity in the gradient direction along the borders of two adjacent regions gives us a measure of the similarity of the shape and illumination of the corresponding surface patches in the scene.

The biggest drawback to using the gradient direction is its sensitivity to noise and the effects of the region boundaries. Furthermore, pixels with small gradient intensities do not have reliable gradient directions as small changes in the intensity can cause large changes in the gradient direction. These problems make it a less robust measure overall than the reflectance ratio. However, with appropriate precautions it makes an effective compatibility test.

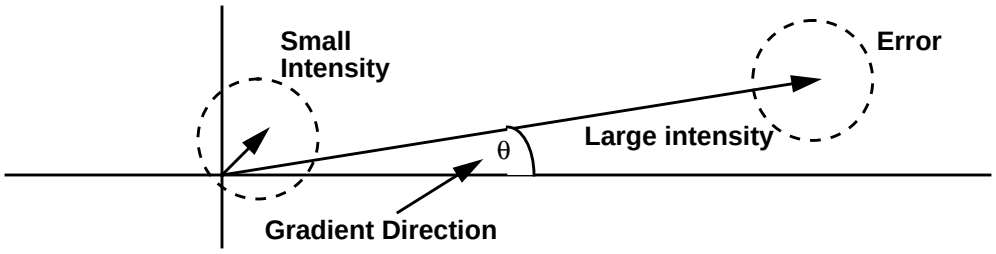


Figure 5.4: Example of how a small gradient intensity makes the gradient direction estimate unreliable in the presence of noise.

The first issue is how to calculate the gradient direction. For this task we want to smooth over small-scale texture in an image but avoid the effects of the region boundaries. To balance these competing interests we use a 5x5 Sobel operator to calculate the gradient intensity in the horizontal (x) and vertical (y) directions. The 5x5 Sobel combines a Gaussian smoothing operator with a derivative calculation. It provides smooth gradients while still calculating them within a few pixels of the region boundary. Like the reflectance ratio, we apply the gradient direction operator to the intensity image rather than a single color band.

Using the horizontal and vertical gradient values returned by the Sobel operator we can calculate the gradient intensity and direction using the polar coordinate transformation given by (6) and (7).

$$|\Delta I| = \sqrt{\Delta I_x^2 + \Delta I_y^2} \tag{6}$$

$$\angle \Delta I = \operatorname{atan} \frac{\Delta I_y}{\Delta I_x} \tag{7}$$

To avoid the effect of the region boundaries, which modify the gradient direction because of the change in albedo, the algorithm does not calculate a gradient value for any pixel unless all of the pixels underlying the 5x5 Sobel centered on that pixel fall within the region of interest. The algorithm then iteratively grows the x and y Sobel results. For each iteration through the region, any pixel without a Sobel result that is 4-connected to at least one pixel with a Sobel result receives the average x and y Sobel values of all of the 4-connected pixels it borders. This continues until all pixels in the region possess an x and y Sobel value from which the algorithm can calculate the gradient intensity and direction. Figure 5.3 shows the calculated values for the two-spheres 0-0 image.

As noted previously, when the gradient intensity is small compared to noise or small-scale texture in an image then the gradient direction estimate is unreliable. Figure 5.4 shows graphically how noise in the image, simplistically modeled here as a circular error, causes the gradient to fall somewhere within a circle around the true gradient position. When the gradient intensity is large compared to the noise then it does not strongly affect the gradient direction. However, as the intensity decreases, the effects of noise and texture on the gradient direction increases.

To deal with this problem we have to use a gradient intensity threshold $T_{\Delta I}$. We would like to choose a large threshold which minimizes the error in the direction estimation. Unfortunately, we are generally comparing smooth surfaces with relatively small gradients. For the compatibility test to be useful we have to keep the threshold as small as possible. The algorithm uses a gradient

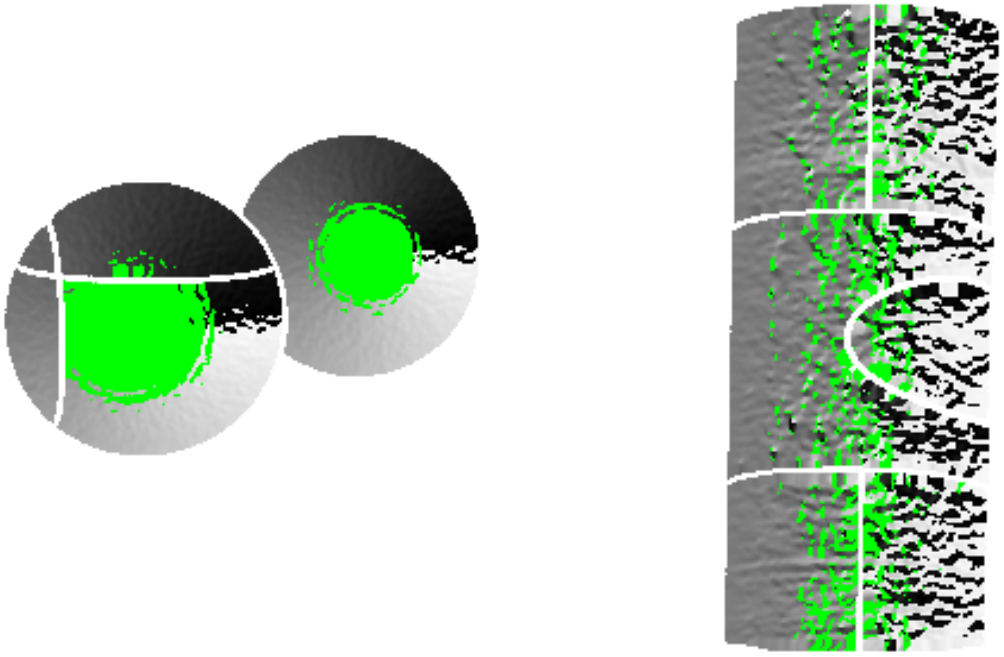


Figure 5.5: Thresholded gradient direction image of (a) two-spheres 0-0 and (b) cylinder. Pixels with gradient intensities less than 2 are shown in green.

intensity threshold $T_{\Delta I} = 2$ for all of the test images. Figure 5.5 shows the thresholded gradient direction for the two-spheres 0-0 image and the cylinder image. Note that in the cylinder image, taken in the CIL, the texture and noise is somewhat larger than the threshold. However, for the synthetic image large chunks of the surface have gradients less than the threshold. The same is true for the ball and cylinder image in Figure 5.6(a). For all of the results presented herein the threshold remains constant at a value of 2. Adapting the threshold to the image of interest might produce more robust results, but this is a topic for future work.

One problem not yet mentioned occurs when an intensity peak or minimum is near a region border. For example, in the cylinder image, the intensity peak follows a line down the center of the cylinder where the top two and bottom two regions happen to meet. At an intensity peak or minimum two things happen. First, the gradient intensity is usually quite small, meaning that noise and texture strongly affect the gradient direction estimation. Second, the gradient directions flip approximately 180 .

We use a simple adjustment to compensate for this phenomenon. The algorithm flips one of the gradient directions whenever the angular difference between them is close to 180 . For each gradient direction pair, if the difference between the angles is greater than a threshold angle then the algorithm flips one of the gradient direction values by 180 and recalculates the angle difference. The algorithm uses an angle threshold of 148 for all cases. Note that it always uses the minimum angle between two gradient vectors, which means the angle differences are always less than or equal to 180 .

We use one final step to smooth out errors in the gradient direction process. After finding the angle

difference of two gradient vectors we put the resulting angle difference into one of ten 18 buckets labeled zero to nine. Therefore, if two angles are within 18 of one another then the counter for bucket zero increases by one. Likewise, if the angle difference is between 19 and 36 then the algorithm increments the counter for bucket one. We then use this histogram to calculate the similarity of the gradient directions for two adjacent regions.

As with the reflectance ratio, we can use the variance of the gradient direction differences to compute similarity. We assume the angle differences follow a normal distribution with a mean of zero. The variance is the sum over all buckets $b \in B$ of the square of the bucket's number times the number of items in the bucket C_b divided by the total number of valid border pixel pairs N less one. This is given explicitly in (8).

$$\sigma^2 = \frac{\sum_{b=0}^{B-1} C_b b^2}{N-1} \tag{8}$$

Given the variance in the gradient direction along the border, we can use the chi-squared test to compare the sample variance to a threshold variance [29]. Because of the conditions required for the gradient directions of adjacent borders to be similar, we can interpret the result as a likelihood that the illumination and shape are similar along the border of two regions. The gradient direction threshold variance $\Sigma_{\Delta I}$ equals 10 for all of the test images. This corresponds to a standard deviation of just over three buckets, or 66 . At first glance this may appear to be a large threshold. However, for the test set images the distributions of bucket values for adjacent regions of different surfaces tend towards uniformity. On the other hand, the distributions for adjacent regions of the same surface are strongly weighted towards the zero bucket with a few outliers. The relatively large threshold variance is necessary because of the effect of these outliers on the variance. However, because regions from different surfaces tend to have more of these "outliers" the threshold effectively differentiates between them in most cases.

Figure 5.6 is a visualization of the differences in gradient direction along the region borders for two real and one synthetic image. Figure 5.6(a) shows the thresholded gradient for the ball-cylinder image. Figure 5.6(b) shows the resulting border comparisons. For the most part, the borders between the regions of different objects in this image indicate larger angle differences than the borders between regions of the same object.

The one exception is the top left region of the ball in Figure 5.6(b), which is such a small region that it is difficult for the algorithm to calculate the gradient at more than a few points in the region. When the algorithm then grows these gradient values, the entire region ends up with approximately constant gradient directions calculated from only a few points. We see this effect in Figure 5.6(a) where the gradient direction is constant over the upper left region.

Figure 5.6(c), the border comparison results for the two-spheres image, also shows large gradient direction differences between the two different objects. The four regions of the left sphere, on the other hand, show very small differences in the gradient direction. Note that the pixels on the flatter area at the center of the sphere fall below the gradient intensity threshold. This image also shows the effect of a local minimum in image intensity at a border. Between the two spheres there is a portion of the border where the gradient directions are approximately 180 apart. Because of

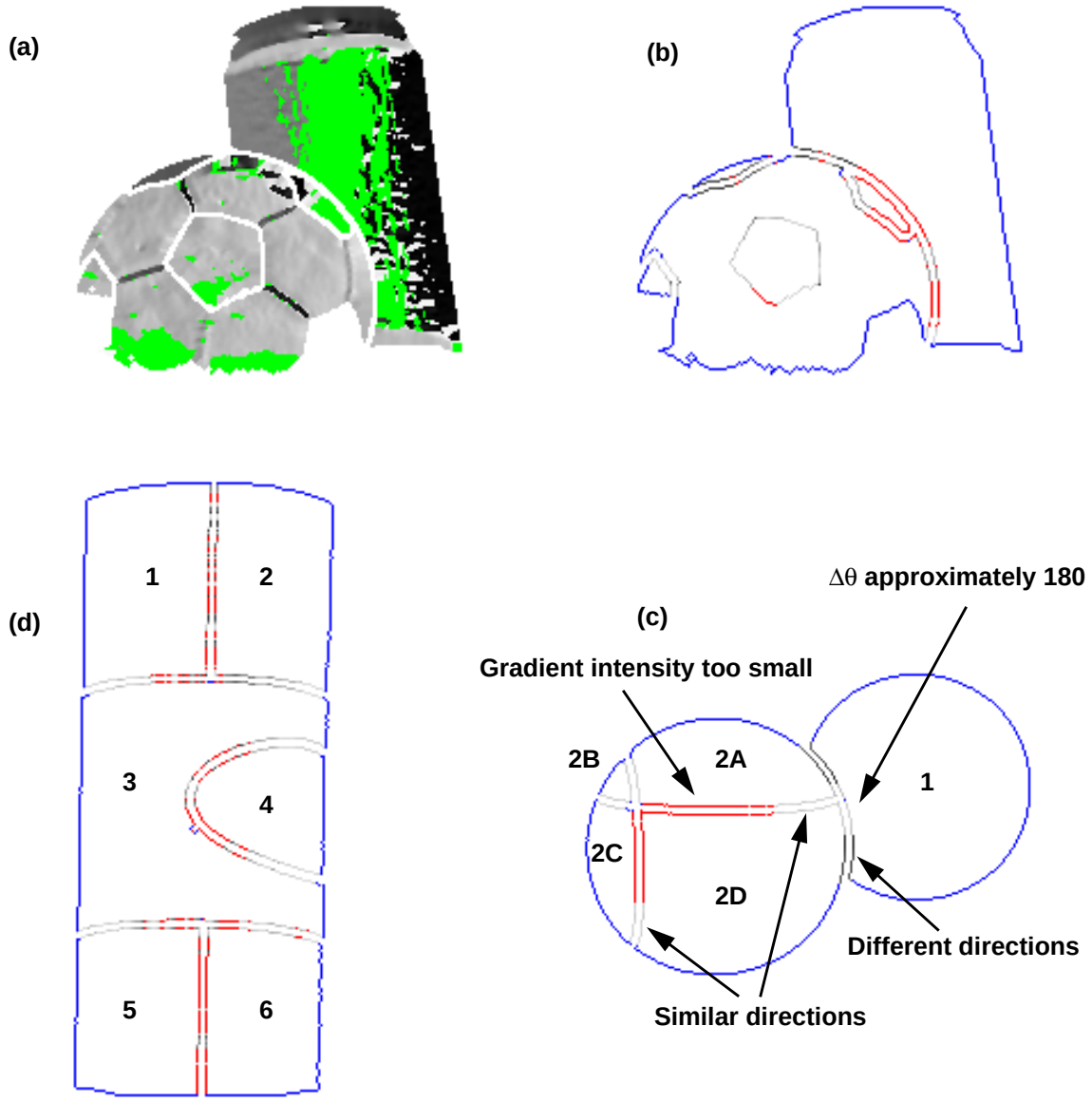


Figure 5.6: (a) Thresholded gradient direction image of the ball and cylinder scene. (b) Comparing the gradient directions for the ball and cylinder scene. (c) Comparing the gradient directions for the two-spheres 0-0 image. (d) Comparing the gradient directions for the cylinder image. Light pixels indicate similar directions. Darker pixels indicate larger direction disparities.

the heuristic where we flip one of these direction vectors, for this portion of the border the gradient direction differences are small.

Figure 5.6(d) demonstrates the effects of having a local maximum in image intensity near a region boundary. The gradient intensities of the pixels along the border of cylinder regions 1 and 2 for the most part fall below the gradient threshold. These pixels appear red in Figure 5.6(d). Most of the pixels along these borders that are above the threshold show large disparities in the gradient directions. However, the borders further away from the center of the cylinder, such as between

Table 5.2: Gradient direction comparison results for the two-spheres and cylinder images. The shaded boxes indicate incorrect results.

Region A	Region B	Variance	Compatible?
two-spheres 1	two-spheres 2	38.8	No
two-spheres 1	two-spheres 5	30.1	No
two-spheres 2	two-spheres 3	0.71	Possibly
two-spheres 3	two-spheres 4	0.66	Possibly
two-spheres 4	two-spheres 5	0.52	Possibly
two-spheres 2	two-spheres 5	0.45	Possibly
cylinder 1	cylinder 2	23.6	No
cylinder 1	cylinder 3	6.4	Possibly
cylinder 2	cylinder 3	9.3	Possibly
cylinder 3	cylinder 4	6.1	Possibly
cylinder 3	cylinder 5	5.2	Possibly
cylinder 3	cylinder 6	2.9	Possibly
cylinder 5	cylinder 6	17.1	No

regions 1 and 3, exhibit similar gradient directions because they are not near an intensity maximum and they are part of the same object.

Table 5.2 shows the calculated variances and compatibility results for the cylinder and two-spheres images. For the two-spheres image the test gives excellent results, showing more than an order of magnitude difference between the region pairs that match and those that do not. For the cylinder image, all of whose regions should match, we see that for the region pairs (1, 2) and (5, 6) the intensity maximum effects cause the test to return a false negative result.

To try and avoid false results, we want to identify borders where most of the pixel's gradient intensities fall below the threshold. For these borders it does not make sense to use the gradient direction compatibility test because there is simply not enough information to make an accurate compatibility determination. When most of the border pixels fall below the intensity threshold, it means one of two things. First, like the case of the cylinder, the border may be at a local intensity maximum or minimum, indicating that the gradient directions are unreliable. Second, the two regions may be planar, which means in most cases that the gradient direction values are due to small scale texture and noise in the image.

We filter out these cases by returning a compatibility result only when at least 20% of the pixel pairs along the border of two regions have gradient intensities above the threshold. For any pair of regions that fall below the 20% requirement, the gradient direction compatibility test returns a

null result, indicating that it does not have enough information to make an informed decision.

Despite the fact that the gradient direction is the least robust of the three tests, it does have one advantage over the reflectance ratio: it is not particularly sensitive to absolute magnitude. So long as the gradient intensity is not small and the algorithm can accurately calculate the gradient direction, the absolute magnitude of a given pixel is irrelevant.

To summarize, the gradient direction compatibility test uses the following stages. To calculate the gradient, horizontal and vertical 5x5 Sobel operators first calculate the smoothed gradient for the interior pixels of a region. Next, the x and y Sobel results iteratively grow to fill out the region by averaging the values of previously calculated 4-connected neighbors. Then the polar coordinate transformation provides a gradient intensity and direction for each pixel in the region.

To test for compatibility, the algorithm first calculates the angle differences between pairs of border pixels. It filters these pixel pairs using a gradient intensity threshold. Both pixels in each border pair must have a gradient intensity equal to or larger than the threshold. To account for the effects of intensity minima and maxima, if the angle difference is close to 180 then one of the gradient directions is flipped by 180 . The algorithm then increments the values of 18 buckets based on the angle differences. If less than 20% of the applicable border pixels contribute because of the gradient intensity threshold, then the compatibility test returns a null result. Otherwise, it calculates the variance in the angle differences based on the number of items in each bucket and compares it to a threshold variance. The chi-square test used to make this comparison returns a value in the range [0,1] which indicates the likelihood that the gradient directions along the border of the two regions are similar. If the likelihood is high, then this builds our confidence that the two regions may be compatible. If the likelihood is low, however, then it is unlikely the two regions are compatible and we can rule out a merger.

5.3. Profile analysis

Both the reflectance ratio and the gradient direction compatibility tests share one drawback: they look only at border pixels to compare two regions. The bodies of the two regions contain a lot of information about their relative geometry and illumination characteristics. Intuitively, when we look at a multi-colored object we do more than look at points near the borders; we also look at the overall smoothness, or continuity of the shading patterns, automatically normalizing for changes in the illumination and transfer function. This intuition motivates the profile analysis compatibility test.

The test works on the following assertion: if two adjacent hypothesis regions are part of the same surface, then, if we take into account the brightness scale change due to the change in transfer function, the intensity profiles of the two regions should be smooth and continuous according to some criteria. As with the other two tests, we cannot claim that hypothesis region pairs that are not part of the same surface will not be smooth and continuous.

Rather than look for discontinuities or breaks in the intensity profiles, we prefer to take a more general approach that maximizes the amount of information we use. We can think about the previous assertion from the standpoint of fitting a smooth model to the intensities of each hypothesis region. If they are part of the same surface, then a single profile model should adequately fit the intensities across both hypothesis regions. On the other hand, if their combined profile contains discontinuities or they have differently shaped profiles, then a single model will not adequately fit

the data. In this case it would be better to use two models.

To demonstrate this assertion, consider Figure 5.7(a), which shows two highlighted scanlines. The scanline A-A' crosses regions 4 and 5, which are part of the same object. Scanline B-B', however, crosses two regions that are part of separate objects. The top graph in Figure 5.7(b) and Figure 5.7(c) both show the raw intensity data and a low-order polynomial fit to the intensity data for each individual hypothesis region. For both of these cases the squared error between the model and the intensities, shown in red, is small.

To fit a model to the intensity data for both hypothesis regions, we must first calculate the average reflectance ratio along the border to adjust the intensities to account for the change in albedo. We have to use the reflectance ratio and not an optimal offset because the change in the transfer function between two regions is a scale change, not a translation of the intensity values. Note that the reflectance ratio in this case is the ratio of adjacent pixels, not the difference of two pixels divided by their sum as in (3). The average reflectance ratio $\bar{r}$ along the border of regions A and B, with N border pixel pairs is given by (9).

$$\bar{r} = \frac{\sum_{i=1}^{N} \frac{I_{Ai}}{I_{Bi}}}{N} \tag{9}$$

If we let region 4 of Figure 5.7(a) be region A and region 5 be region B with respect to (9), then by multiplying the intensities from A'' to A' by the average reflectance ratio we adjust for the difference in albedo. For this particular case a single polynomial is a good model for the intensity profile across both hypothesis regions as indicated by the small errors in the bottom graph of Figure 5.7(b).

We can use the same procedure for the scanline B to B'. For this case a single model is not a good fit for the data as shown by the large errors in the bottom graph of Figure 5.7(c).

In order to use this intuitive procedure as a test of compatibility we have to A) choose what models to use, B) choose which scanlines to use, or whether we want to fit models to the entire region, C) choose the order of the model to use for each region and for their combination, C) choose how to decide when using two models is better than one, and D) convert this into a probability we can combine with the other two tests.

For this task we decided to use low-order polynomials for a number of reasons. First, they are simple to fit to both scanlines and surface patches [50]. Second, they are general models and do not assume anything about the underlying surfaces except that they are continuous. Finally, we can flexibly select the order of the polynomial based upon each individual case.

5.3.1 Choosing which data to model

The next decision we had to make was whether to fit models to scanlines or areas, and which scanlines to use. If we want to maximize the amount of information incorporated into each model, then fitting surface patches to the intensity values is the more attractive option. We implemented the surface fitting option and found that it has three drawbacks compared to fitting scanline intensity profiles.

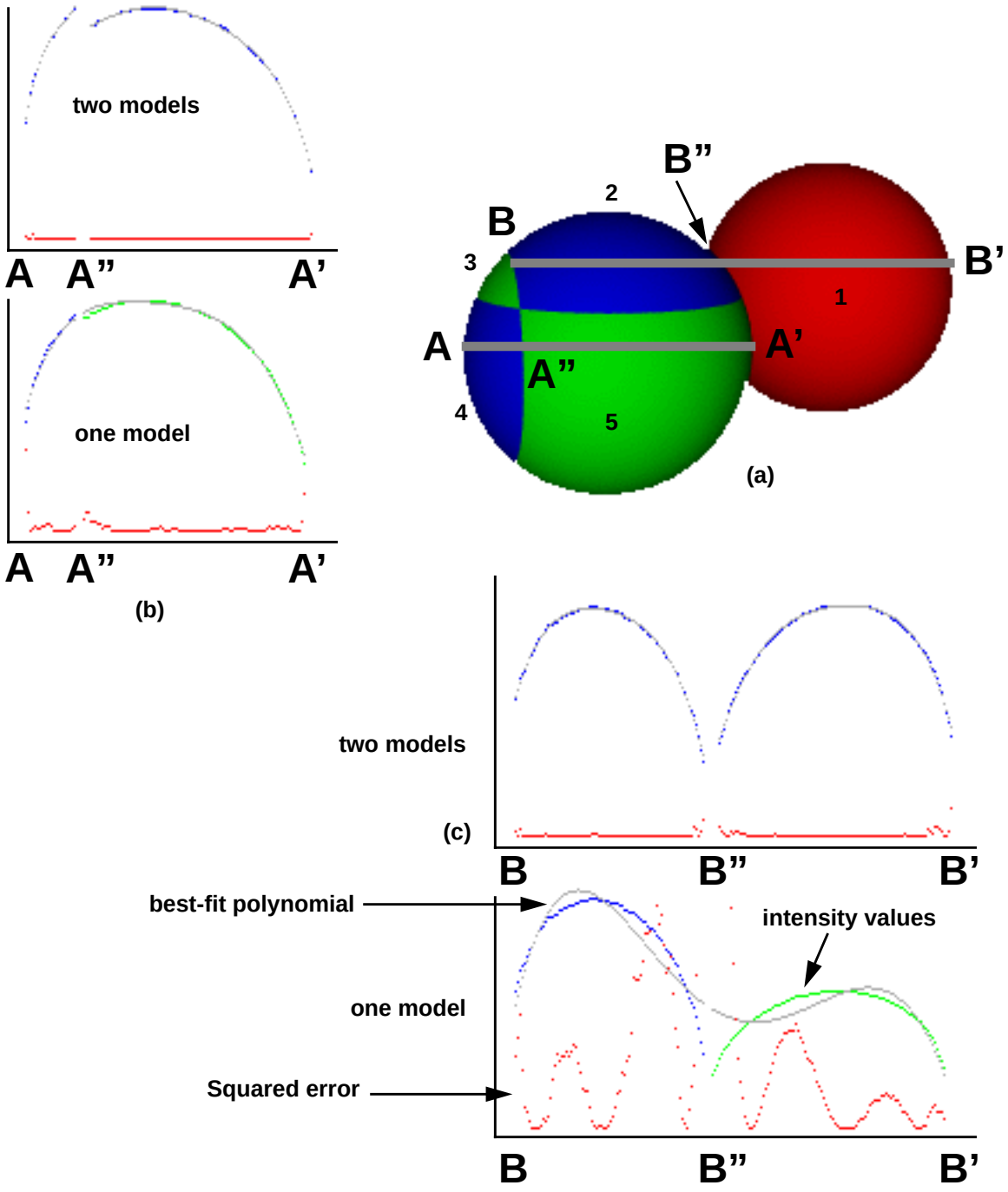


Figure 5.7: Profile analysis for the two-spheres image. (a) two-spheres 0-0 image with two horizontal scanlines highlighted. A-A’ crosses two regions of the same object, B-B’ crosses two different objects. (b) Intensity data, best-fit polynomial, and squared error for A-A’. (c) Intensity data, best-fit polynomial and squared error for B-B’. The intensity data is in blue and green. The polynomial is shown in grey, and the squared error is shown in red.

- First, it takes significantly longer to fit a surface to the data than it does to fit a curve to a single scanline. We used singular value decomposition [SVD] to find the least-squares polynomial, which is an $O(N^3)$ algorithm. Because of the relative run-times, it is a lot faster to fit individual models to each scanline than one model to the entire region.
- Second, the surface patches need to be higher order, in general, than their scanline counterparts. This is because it is not uncommon for a few pixels within a region to display some specularity or small-scale texture. For example, the green cup in Figure 5.8(a) has some specular points near its top edge. When we fit a low-order surface to the cup region these pixels act as outliers and pull the surface away from the other intensities. This causes large squared errors for the individual region fits which biases the results towards using a single model for both regions. Using a higher-order surface, however, takes even more time and gives us a less general surface. On the flip-side, even if we still fit every pixel in the region using a scanline method, the specular or textured pixels only show up in a few of the scanlines, while the majority of them use less noisy information.
- Finally, the surface fitting method is less robust because it bases its decision on only one comparison. Alternatively, using multiple scanlines we get more data points with which to make a decision. As in the case of specularity or small-scale texture identified above, this allows us to filter or average the data to remove outliers and get a more robust measure of compatibility. For these reasons we decided to use scanlines as the basis for the profile analysis test.

The question then becomes, which scanlines do we use? Ultimately, we would like to use those scanlines with the most information. We also want to use as many as possible. It does not make sense, however, to use scanlines that do not contain adjacent border pixels. If two regions are not adjacent on a given scanline, we cannot assume they will have coherent intensity values even if they are part of the same surface. Given these factors, we chose to follow along the border of two adjacent regions and look at one scanline for each border pixel.

For each border pixel we then have to select the scanline direction. To keep things simple, we look only vertically or horizontally in the image. Two factors affect the decision of which direction to use. The primary consideration is whether there is enough information to fit a model to the data. If both directions contain a sufficient number of data points then the local border tangent determines which direction to use. The algorithm defines sufficiency as 20 data points in each of the adjacent regions. If either direction contains fewer data points than this threshold, then the algorithm chooses the direction with more information.

To find the local tangent, the algorithm looks at the relative position of the nearest border point in the adjacent region as found by the initial border analysis algorithm described in Chapter 3. If the difference in rows is greater than the difference in columns, the algorithm uses the vertical scanline. Otherwise it uses the horizontal scanline.

We also experimented with using the direction with greater variation in the intensities. The idea was that the directions with more variation would contain more information with which to make a decision. While this is true in some cases, what happened with the actual images is that the specularities, areas of small-scale texture, and very dark patches “captured” a large portion of the scanlines because they contained the most variation. Unfortunately, these areas also gave the least reliable results, forcing us to discard this method.

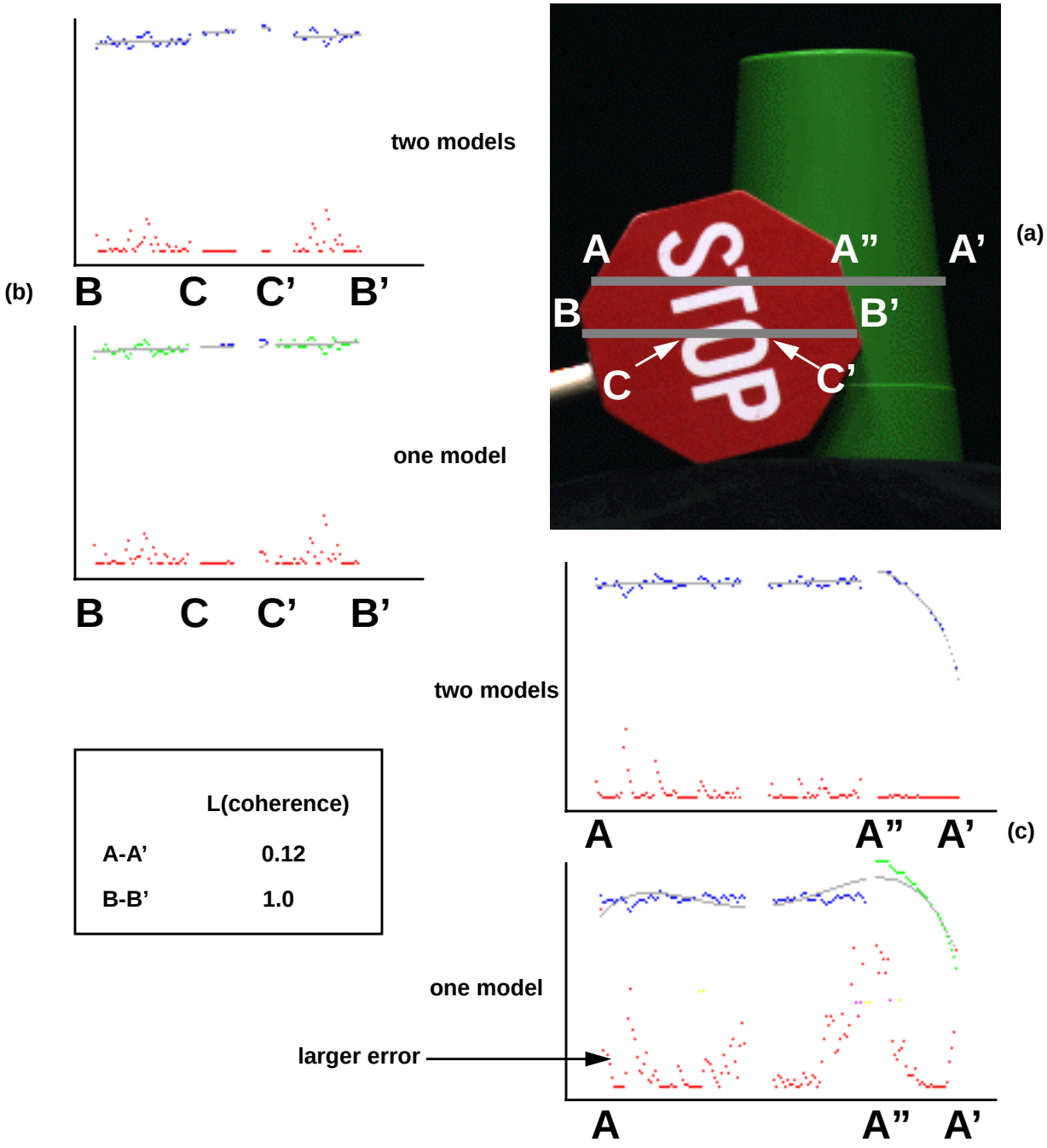


Figure 5.8: Profile analysis for the stop-sign and cup image. (a) Stop-sign and cup image with two scanlines highlighted. A-A’ crosses two different objects. B-B’ crosses differently colored regions of one object. (b) Intensity profiles, best-fit polynomials, and squared error values for B-B’. Note that the error is about the same in both cases. (c) Intensity profiles, best-fit polynomials, and squared error values for A-A’. Note that the error is much larger using a single model than it is using two.

5.3.2 Modeling the data

Having chosen to use scanlines, and knowing which scanlines we want to use, we now turn our attention to the examination of a single scanline. Our first task is to find the best-fit polynomial of a given order. SVD is a simple and robust method of finding the least-squares polynomial. We can set up the problem as a set of linear equations

$$\begin{bmatrix} {c_1}^0 & {c_1}^1 & \dots & {c_1}^k \\ {c_2}^0 & {c_2}^1 & \dots & {c_2}^k \\ \dots & \dots & \dots & \dots \\ {c_N}^0 & {c_N}^1 & \dots & {c_N}^k \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \dots \\ a_k \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_N \end{bmatrix} \tag{10}$$

where the polynomial coefficients $a_0 ... a_k$ are the unknowns of the kth order polynomial, the c_i represent the column or row numbers of the corresponding x_i data points, and N is the total number of intensity values along the profile.

Because SVD is a least-squares approach, outliers can draw the resulting polynomial away from the actual surface. To minimize this problem the algorithm makes one smoothing pass through the intensity data for each individual region and averages each pixel with its two closest neighbors. The combined data is simply the union of the smoothed values for the individual regions; the algorithm does not make a second smoothing pass.

Given a least-squares polynomial for a given order, we now want to find the best order for the profiles of each individual region and for their combined profile. We would like to minimize the error between the model and the data, while at the same time keeping the order reasonably low so we maintain the overall shape of the intensity profile. For guidance we turned to the minimum description length [MDL] and the process for choosing a model's order described by Rissanen [51].

Rissanen argues that the best model for a set of data is the one that describes the data in the shortest length. This length includes not only the number of model parameters, but also a term based upon the error between the model and the data, which any complete description of the data must include. The formula for calculating the description length of a polynomial model is given in (11), where x^n is the data, θ is the set of model parameters, k is the number of model parameters, and N is the number of data points [51].

$$DL = -\log P\left(x^n \middle| \theta \right) + \frac{k}{2} \log N \tag{11}$$

The first term in (11) reflects the error between the model and the data. If we model the error at each pixel as a Gaussian, then the probability of a set of data given the model is given by

$$P\left(x^n \middle| \theta \right) = \prod_{i=1}^{N} e^{-\frac{(m_i - x_i)^2}{\sigma^2}} \tag{12}$$

where N is the number of data points, σ is the standard deviation of the error, the m_i are the model values, and the x_i are the actual values. If we substitute (12) into (11) then the first term becomes the sum squared error divided by the variance. As the second term depends upon the order of the model and the number of data points described by it, the MDL thus balances the error and the model's order. Note that for all cases the algorithm uses a variance of 4 pixel values, or a standard deviation of 2 pixel values, which is on the order of the noise inherent in the camera used to take the test images [25].

Given that we can calculate the MDL for a polynomial of a given order, we are now able to find the best order model for a given data set. We simply pick the order of the model whose best-fit polynomial produces the minimum description length. When choosing a model for the individual regions, we limit our search to orders one through five. We need to go as high as order five because of the sharp features present in some regions. Beyond order five, however, the polynomials begin to lose the overall shape of the intensity profile and conform to outlier values.

To find the best order for the combined intensity profile, we limit the search to order one through the maximum of the best orders chosen for the two individual intensity profiles. For example, if the best order for the profile A-A" in Figure 5.7(b) is two and best order for the profile A"-A' is three, then the algorithm searches only orders one to three for the profile from A-A'. If we allow the algorithm to test higher order polynomials, then the additional degrees of freedom allow the polynomial to conform to any discontinuities between the two regions. For example, in Figure 5.7(c) a higher order polynomial would be better able to fit the discontinuity between the two regions, whereas a polynomial sufficient to describe one of the two regions cannot. This biases the compatibility test towards returning a negative response for region pairs whose combined intensity profile has significantly more complexity than the individual regions.

5.3.3 Comparing two models to one

Now that we can find the best-fit best-order polynomial for each individual and combined intensity profile, we are ready to test whether using two models for the combined profile is better than one for a particular scanline. To make this comparison we again turn to the MDL principle. If the MDL of the single model for the combined intensity profile is similar to or less than the sum of the MDLs of the models for the individual intensity profiles, then it is better to use one model and the two regions are potentially part of the same surface.

Defining what we mean by "similar to or less than" turns out to be a sticky problem, however. From extensive experimentation, the MDL of the combined profile is rarely less than that of the sum of the MDLs of the individual profiles. Therefore, we need some non-zero threshold within which we can declare the two values as equal. This threshold cannot be static, however, because some scanlines have large MDLs, while other scanlines's MDLs are quite small. It must reflect the relative sizes of the MDLs as well as the relative difference between them.

Inspiration for how to solve this problem came from the bounded reflectance ratio function defined by Nayar & Bolle [41]. This function uses the difference in two values divided by their sum, which provides a well-behaved, bounded measure of the difference between two values expressed as a percentage of their combined value. This turned out to be a useful measure of equality for the profile analysis.

We define the profile analysis measure of merit m_{AB} for a scanline crossing two regions A and B

as

$$m_{AB} = \frac{MDL_A + MDL_B - MDL_{AB}}{MDL_A + MDL_B + MDL_{AB}} \tag{13}$$

where MDL_A and MDL_B are the individual minimum description lengths of the best-order, best-fit polynomials for the scanline profiles of regions A and B, respectively. MDL_{AB} is the minimum description length of the best-order, best-fit polynomial for the combined scanline profile. Technically, this measure can have any value in the range [-1,1], however, as MDL_{AB} is rarely less than $MDL_A + MDL_B$, in practice this function is generally in the range [-1,0]. Values near -1 indicate that using two models is much better than one, while values near zero indicate that using one model is as good or better than using two.

We convert (13) into a likelihood in the range [0,1] according to the rule given in (14).

$$L(coherent) = \begin{cases} 1 & m_{AB} \geq 0 \\ 1 + m_{AB} & m_{AB} < 0 \end{cases} \tag{14}$$

The resulting likelihood function indicates whether two surfaces are coherent and smoothly decreases as the cost of using a single model increases relative to the cost of using two. For example, the likelihoods given for the scanlines A-A' and B-B' in Figure 5.8(b) and (c) indicate that the le sign and the letter O belong together, but the stop-sign and cup do not.

We now have all of the tools necessary to compare two regions using profile analysis. As noted previously, we look at one scanline for each border pixel pair, using the data size and local tangent to determine whether we analyze a vertical or horizontal line. For each line we find the best-fit, best-order polynomials, calculate the individual and combined MDLs, and calculate the measure of merit. The algorithm stores the measure of merit and the order, the sum-squared error, and the number of data points for each of the three polynomials.

When the algorithm finishes analyzing all of the scanlines it calculates the mean order, mean sum-squared error, and mean number of data points for the individual region data and the combined data. It then uses these average values to calculate a mean MDL for each region and for their combination using (11). Substituting these values into (13) returns an average measure of merit for the two regions.

Because it also calculates the measures of merit for each individual scanline, the algorithm can calculate not only a mean measure of merit, but also the median measure of merit. Using (14) it then calculates mean and median likelihoods of coherence. Both of these statistics are general values describing the coherence of the two regions. In general, however, they are not the same value. After testing 102 region pairs and manually inspecting the median and average results, we came up with the following rule: if the mean likelihood is less than 0.5, then return the lesser of the mean and median likelihood values, otherwise return the greater of these two values.

The result of this rule is that the profile analysis always returns the most extreme value of the mean and median likelihoods, but uses the mean likelihood to make a decision when the two statistics straddle 0.5.

Table 5.3: Results of the profile analysis compatibility test on the two-spheres and stop-sign and cup image. The P-hole region does not contain enough points on any scanline to fit a polynomial to it.

Region A	Region B	Compatible?
two-spheres 1	two-spheres 2A	No
two-spheres 1	two-spheres 2D	No
two-spheres 2A	two-spheres 2B	Possibly
two-spheres 2B	two-spheres 2C	Possibly
two-spheres 2C	two-spheres 2D	Possibly
two-spheres 2A	two-spheres 2D	Possibly
stop-sign	cup	No
stop-sign	pole	No
stop-sign	letter S	Possibly
stop-sign	letter T	Possibly
stop-sign	letter O	Possibly
letter O	O-hole	Possibly
stop-sign	letter P	Possibly
letter P	P-hole	insufficient information

Because there is some dependency upon which border the algorithm follows, we run the algorithm once along the border of each region. Each run produces a likelihood of coherence, which we average to get a final likelihood for the coherence of the hypothesis region pair.

Figure 5.8(b) and (c) demonstrate some of the strengths of the profile analysis compatibility test. When analyzing B-B', the algorithm compares two planar surfaces. Because the MDL takes the cost of the model order into consideration, the models chosen for each individual region are both linear despite the noise and texture of the painted stop-sign. This strongly limits the possible model orders for the combined profile which speeds the search for the best-order best-fit model as the algorithm only has to check one case. Since the surfaces are coherent, however, using a linear model for the combined profile still results in a shorter description length than two separate models.

For the A-A' case, however, limiting the order of the combined profile model actually helps the algorithm to return a low likelihood. Because the combined profile is much more complex than either of the two individual profiles, the limitation on model order increases the error in that case, reinforcing the fact that the two surfaces are not coherent.

The A-A' case also shows how the reflectance ratio plays a part in the profile analysis. Note in the top graph of Figure 5.8(c) that the intensities of the individual regions at the border are actually similar for this particular scanline. However, because the reflectance ratio is not constant between the two regions, the average reflectance ratio used to modify the combined intensity profile actually increases the intensity disparity for the A-A' scanline. In fact, whenever two adjacent regions do not have a constant reflectance ratio along their mutual borders, this will potentially create more disparities in the intensity profiles. In this way the reflectance ratio actually helps reinforce the intensity profile analysis.

Table 5.3 shows the final results of the profile analysis compatibility test for the two-spheres and stop-sign and cup image. For these two images the results are correct, with the exception of the hole in the letter P for which there is too little data to make a comparison. Of note are the region pairs (2,3) and (4,5) in the two-spheres image which have relatively low positive values despite fairly good matches like Figure 5.7(b). The problem in this case is that the individual model fits are so good that even very small errors like those in the bottom graph of Figure 5.7(b) cause reasonably large differences between the combined and individual MDL values. It turns out that a small amount of noise or texture in the image, such as we see in Figure 5.8(b), can actually make the profile analysis perform better because the individual models do not fit the data so precisely.

5.4. Merging the results

Given these three tests of compatibility for curved and planar white and colored dielectrics under white uniform illumination, we now have to combine them into a single merge likelihood in order to create the region graph. All three of these tests return likelihoods in the range [0,1]. However, they behave differently for different images, have different failure modes, and differ in their reliability, making the task of combining them in a useful manner less straightforward than it may appear.

The first method we tried was multiplying the three values together. This is an appealing approach because these tests are weak methods that, in theory, are most trustworthy when they return low merge likelihoods. Multiplication amplifies the effect of small values, encouraging a not-merge decision when any one of the tests returns a low merge likelihood.

Unfortunately, multiplication goes too far in this direction. We found three basic problems. First, as we will see, all three tests are not necessarily trustworthy when they return a low merge likelihood, henceforth called a negative response. Using multiplication, a single false negative response forces a not-merge decision even if the other two tests return high merge likelihoods, or positive responses. This is not the behavior we want from our system.

The second problem with multiplication is that several low positive responses can generate a not-merge decision. For example, if each of the three compatibility tests return a likelihood of 0.75, then multiplying them together produces a value less than 0.5, which is a not-merge decision. Again, this is not the behavior we want from our system.

The final problem with multiplication is that it makes it difficult to compare different combinations of tests. For example, if for a given region pair the gradient direction test decides it does not have enough information to make a decision, then the likelihood of the region pair depends only on the reflectance ratio and profile analysis comparisons. Therefore, two responses of 0.75 result in a positive response overall. However, if the gradient direction test does have enough informa-

Table 5.4: Overall performance of the three compatibility tests on 179 examples.

Test	Positive Correct	Negative Correct	False Pos.	False Neg.	% Pos. Correct	% Neg. Correct	Total % Correct
RR	134 / 134	16 / 37	21	0	100%	43%	88%
GD	49 / 59	20 / 26	6	10	83%	77%	81%
PA	120 / 138	31 / 38	7	18	87%	82%	86%
Wt. Avg.	138 / 140	33 / 39	2	6	99%	85%	96%

tion and also returns a 0.75 value, then the overall response is negative. This penalizes region pairs that contain enough information for all of the tests to return a value. For this and the aforementioned reasons, we opted not to use simple multiplication to combine the test results.

Instead of multiplication, we chose to use weighted averaging. To determine the relative weights, early in the system development we tested the reflectance ratio, gradient direction, and profile analysis compatibility tests on 179 region pairs. 140 of these region pairs were part of the same object, 39 were not. The images were: ball-cylinder, mug, mug-plane, cylinder, pepsi, plane, plane-cylinder, stop-sign and cup, stop-sign and egg, and two-cylinder.

The overall results of these tests are shown in Table 5.4. The profile analysis proved to be the most balanced test in terms of when it failed. The reflectance ratio test, on the other hand, tended to return false positives when it failed. Note, the variances of these false positive results are not, in general, any different than the variances for correct positive results so modifying the threshold variance was not a solution.

To find the best weighted average we exhaustively searched the space of weights in 0.05 increments, assigning at least a 0.1 weight to each compatibility test. The error surface turned out to have a fairly well-defined maximum at the point (0.25, 0.1, 0.65) where the weights are for the reflectance ratio, gradient direction, and profile analysis compatibility tests respectively. A two-dimensional view of this error surface appears in Figure 5.9. In cases where one or more of the tests returns an invalid result because of insufficient information we use the following two rules. First, if only one test returns a result then use that result. Second, if only two tests return a result then split the weight of the unused test evenly between the two. This results in the following algorithm.

1. If none of the tests return a valid result for lack of information, return 0.5.
2. else if all of the tests return a valid result, then the overall result is given by (15).

$$L(coherence) = 0.25RR + 0.1GD + 0.65PA \tag{15}$$

3. else if the profile analysis and reflectance ratio return valid results then use (16)

$$L(coherence) = 0.3RR + 0.7PA \tag{16}$$

4. else if the profile analysis and gradient direction return valid results then use (17)

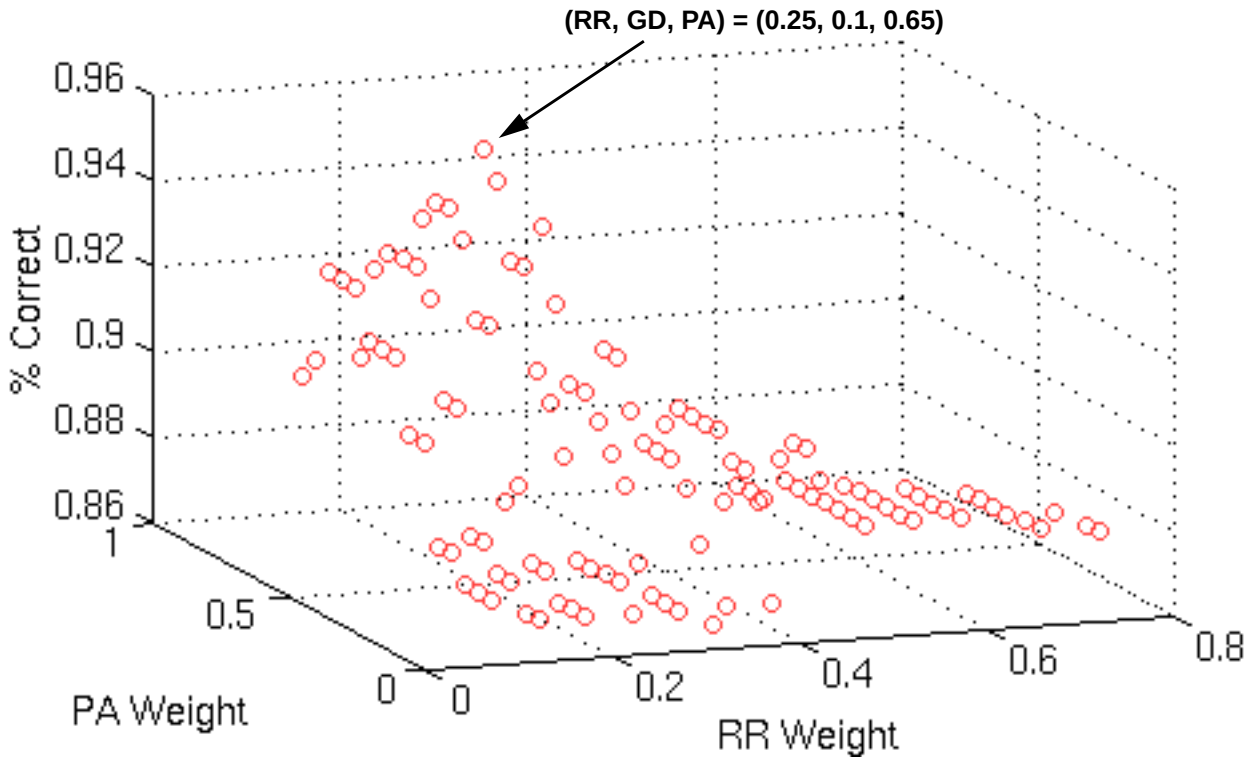


Figure 5.9: Graph of the %correct surface for different weightings of the compatibility tests based on 180 different region pairs. The maximum is located at (RR, PA, GD) = (0.25, 0.65, 0.1). This graph shows the 3D slice with the gradient direction weight at 0.1.

$$L(coherence) = 0.225GD + 0.775PA \quad (17)$$

5. else if the reflectance ratio and gradient direction return valid results then use (18)

$$L(coherence) = 0.425GD + 0.575RR \quad (18)$$

6. else return the result of the single valid test.

Table 5.5 shows the results of each test, their product and their weighted average for the two-spheres and stop-sign and cup images. Because of the weighting scheme and the different failure modes the weighted averaging returns the best results for these two images. While multiplication returns stronger negative results, as with the (stop-sign, cup) and (stop-sign, pole) region pairs, it produces dangerously low results for the region pairs (two-spheres 2A, two-spheres 2B) and (two-spheres 2C, two-spheres 2D). The weighted averaging results are more robust to low positive results and yet still handle the false positive reflectance ratio results such as (two-spheres 1, two-spheres 2A) and (two-spheres 1, two-spheres 2D).

In addition to these specific statistics, we can visualize the compatibility test data in few different ways. Figure 5.10 and Figure 5.11 show three different 2-D graphs of the results, highlighting the correlation between pairs of the compatibility tests. We can also view this data in 3-D as in Figure 5.12, which shows two views of a 3-D plot.

Visual inspection of the 2-D graphs gives us an idea of the relative quality of the three compatibility tests. In Figure 5.10(a) we see that the reflectance ratio and profile analysis are strongly corre-

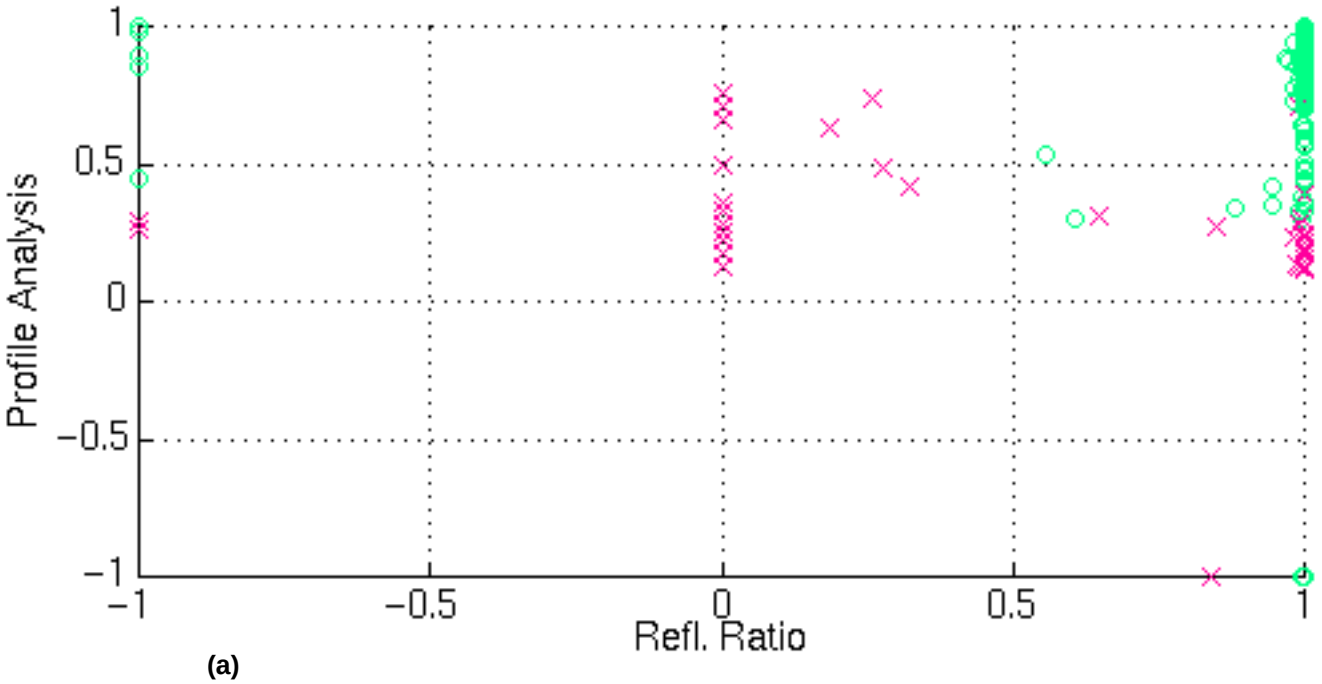
Table 5.5: Results of all three tests, their product, and weighted average for the two-spheres and stop-sign and cup images.

Region A	Region B	RR	GD	PA	Product	W. Avg.	Compatible ?
two-spheres 1	two-spheres 2	0.67	0.0	0.06	0.0	0.21	No
two-spheres 1	two-spheres 5	0.97	0.0	0.11	0.0	0.31	No
two-spheres 2	two-spheres 3	1.00	1.00	0.58	0.58	0.73	Possibly
two-spheres 3	two-spheres 4	0.99	0.99	1.00	0.98	1.00	Possibly
two-spheres 4	two-spheres 5	1.00	1.00	0.68	0.68	0.79	Possibly
two-spheres 5	two-spheres 2	1.00	1.00	0.93	0.93	0.95	Possibly
stop-sign	cup	0.0	na	0.26	0.0	0.18	No
stop-sign	pole	0.0	0.0	0.13	0.0	0.08	No
stop-sign	letter S	1.00	na	1.00	1.00	1.00	Possibly
stop-sign	letter T	1.00	na	1.00	1.00	1.00	Possibly
stop-sign	letter O	1.00	na	1.00	1.00	1.00	Possibly
letter O	O-hole	1.00	na	1.00	1.00	1.00	Possibly
stop-sign	letter P	1.00	na	0.99	0.99	0.99	Possibly
letter P	P-hole	1.00	na	na	1.00	1.00	Possibly

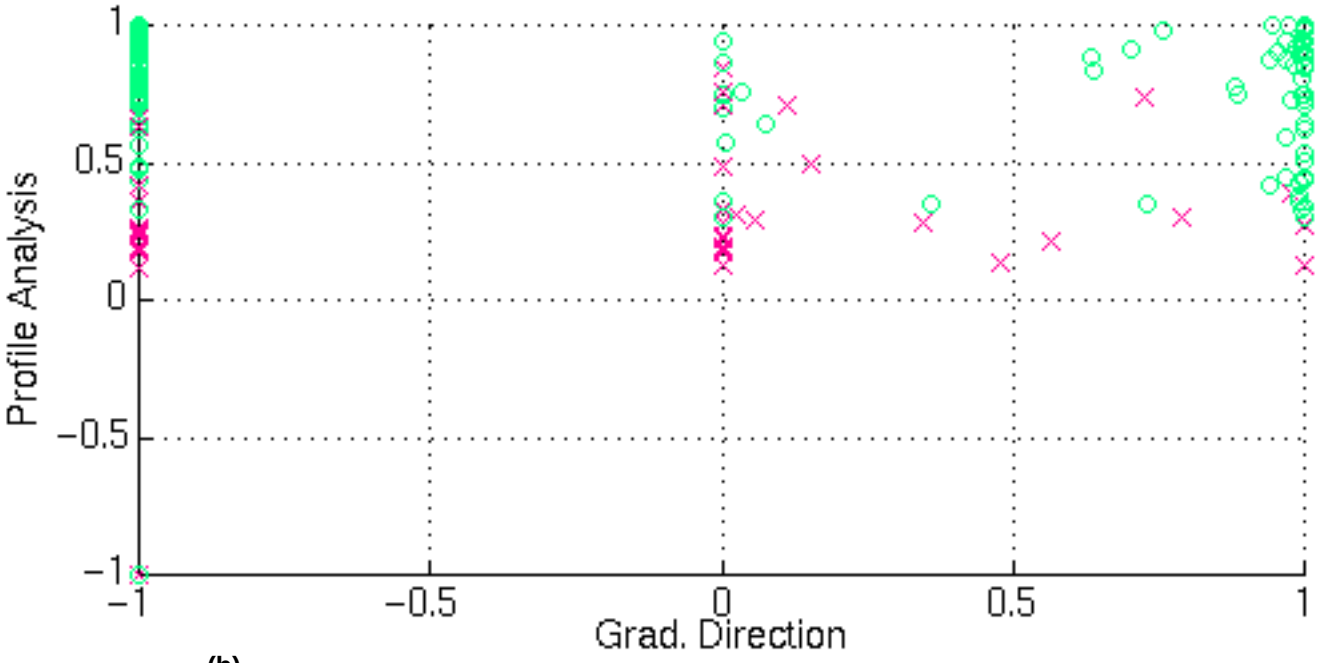
lated; all but one of the data points in the upper right section of the graph are regions pairs that are part of the same object. Therefore, when both of these tests return high values then the likelihood is high that the two regions should be merged. Likewise, when both return low values the regions should not be merged. There is, however, a gray area in the middle where the profile analysis does not return definitive values. We can also see from this graph the susceptibility of the reflectance ratio analysis to false positives, as shown by the relatively large number of red x's with a profile analysis result of 1.0. Note, however, that in all but one of these cases the profile analysis returns a relatively low number, showing the different failure modes of these two compatibility tests.

The other two 2-D graphs also show correlation between the compatibility tests. What is clear from these graphs is that the gradient direction test is slightly less reliable than the other two. Its strengths, as seen in Figure 5.11 are that it has a more balanced failure distribution than the reflectance ratio and can still return useful results when the borders are too dark for the reflectance ratio to work properly. The latter observation is supported by the three data points on the upper left side of Figure 5.11.

The 3-D graph, shown from two viewpoints in Figure 5.12 gives us the intuition that the merge cases are reasonably well-separated from the not-merge cases, something that is not as clear from the 2-D graphs. We can visualize a volume centered on the (0,0,0) point in the data space within which most of the red x's would fall.



(a)



(b)

Figure 5.10: 2D graphs showing the correlation between the three compatibility tests. (a) profile analysis and reflectance ratio, (b) profile analysis and gradient direction. The green circles show regions pairs that are part of the same object, the red x's show regions pairs that are not.

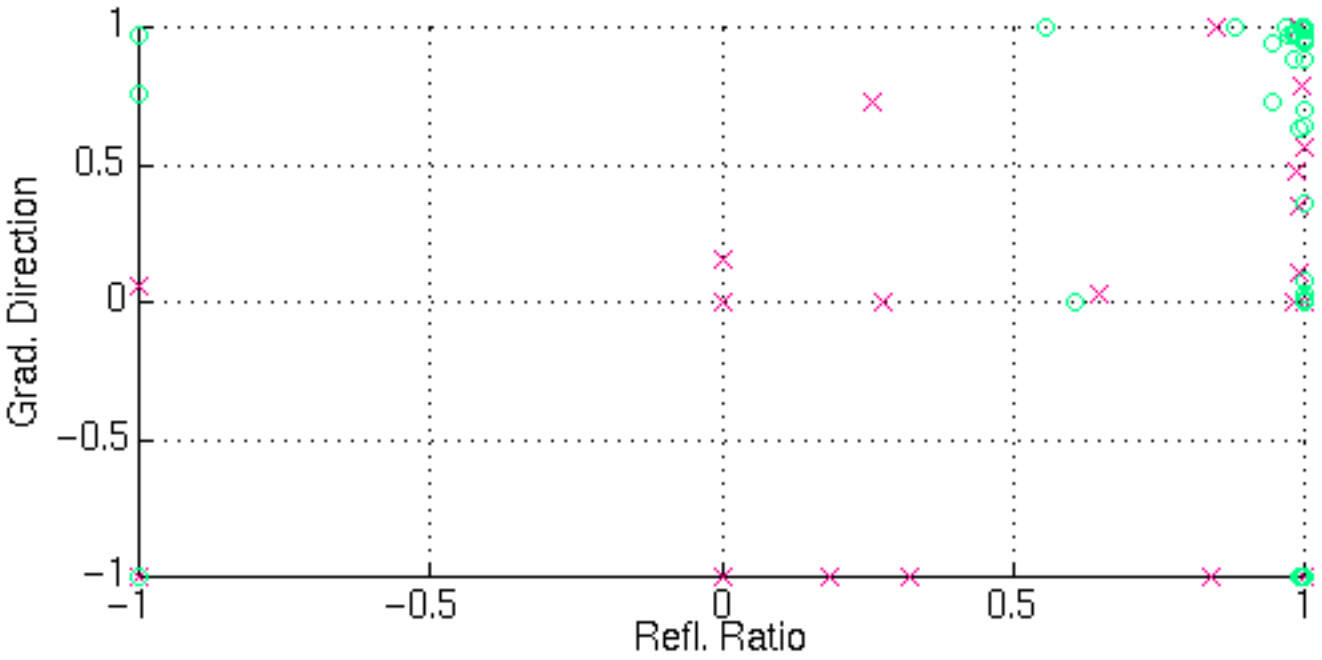


Figure 5.11: 2D graph showing the correlation between the gradient direction and the reflectance ratio. The green circles show regions pairs that are part of the same object, the red x's show regions pairs that are not.

Perhaps the most important observation about these graphs is that planar surfaces are not optimal to separate the merge cases from the not-merge cases. In order to obtain better performance from this combination of operators, we would need to use a more general threshold surface. Future work in this direction should investigate learning techniques such as artificial neural networks which could learn a non-linear mapping within the space [need to cite someone broad]. Artificial neural networks could also integrate other statistics into their decision-making process such as the length of the borders, absolute errors, and other information not currently used. The key to using learning methods is gathering a large number of balanced data points with which to train the network.

What this section has shown is that we can reliably and robustly test for region compatibility using physical characteristics such as the reflectance ratio and gradient direction as well as more intuitive methods like the profile analysis. Furthermore, by combining the results of these tests, which have different failure modes and different strengths, we can obtain better performance than by only using a single method.

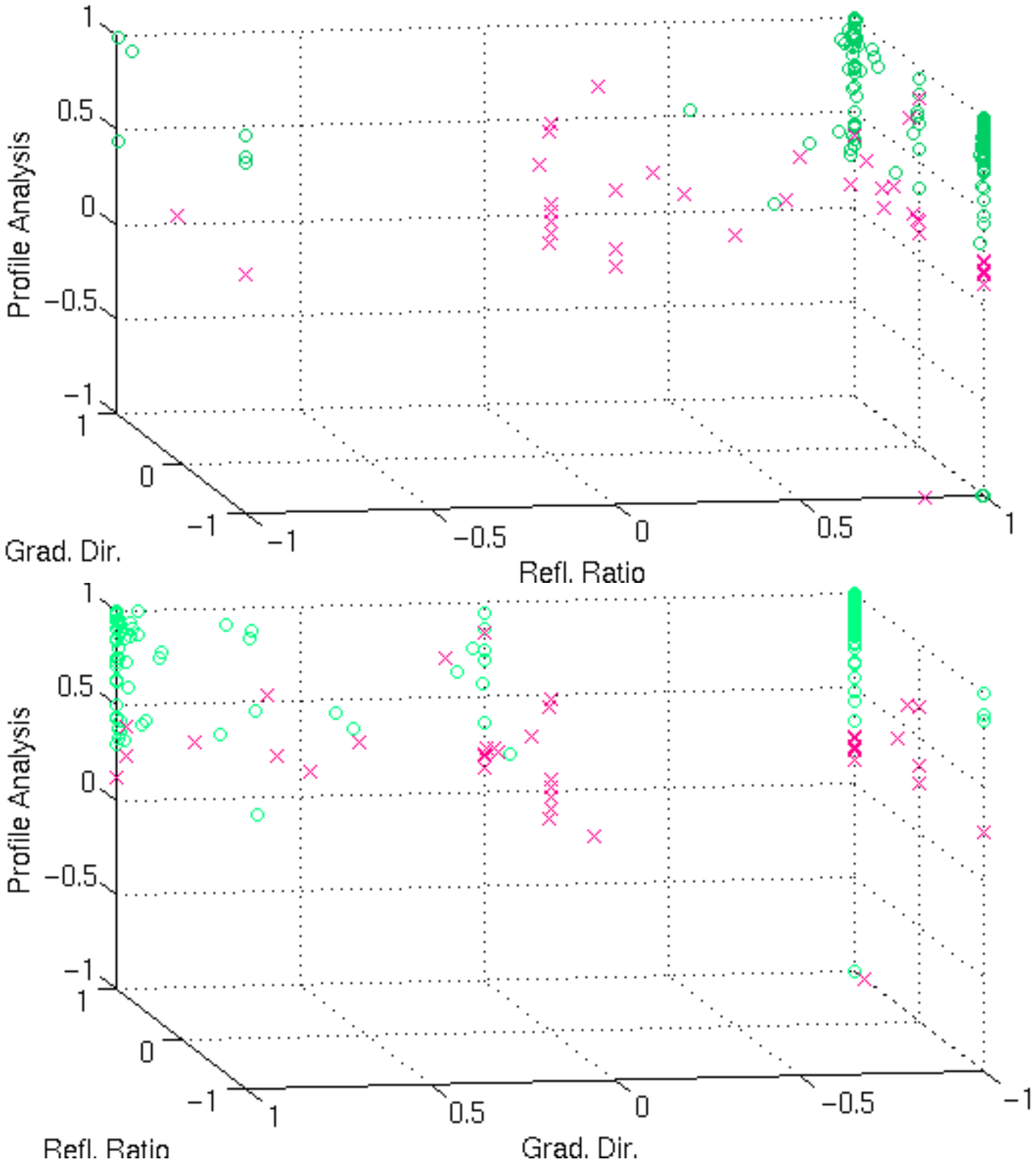
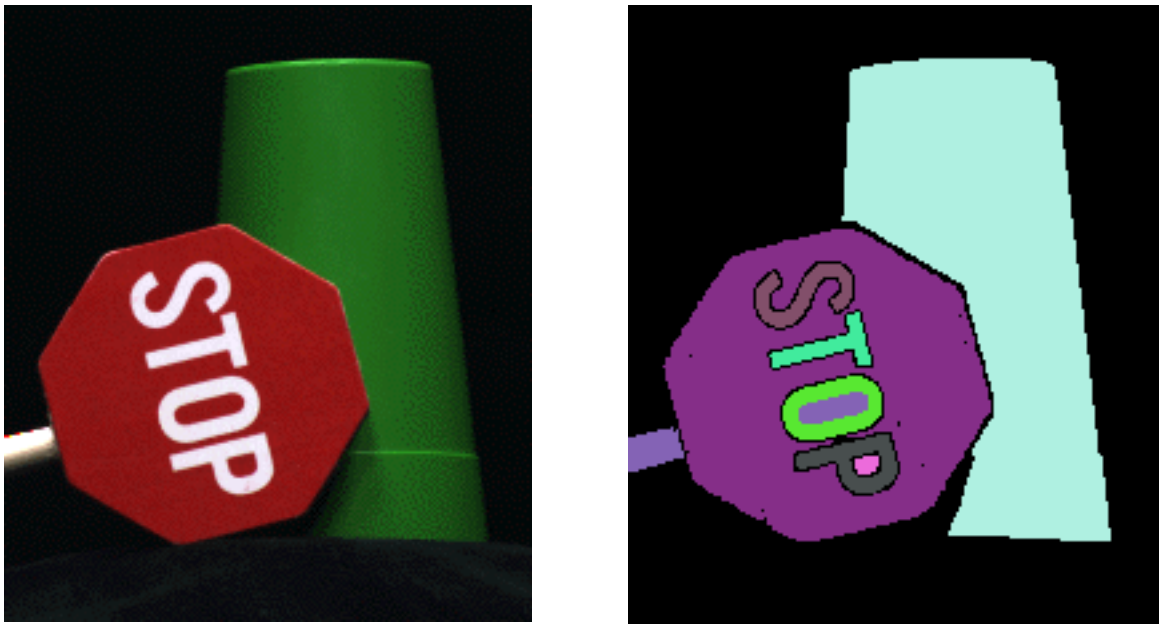


Figure 5.12: 3-D plot of the compatibility test results. The green circles show region pairs that are part of the same object, the red x's show region pairs that are part of different objects.

Chapter 6: A picture is worth a thousand bugs



(a) (b)

Figure 6.1: (a) Stop-sign and cup image taken in the Calibrated Imaging Laboratory, and (b) initial segmentation of the image.

In every research endeavor there are milestones and markers along the way telling us how well we're doing. There are also hurdles, hills, rivers, and mountains to cross to reach the final goal. The stop-sign and cup image shown above turned out to be not only a number of the milestones, but also several of the hills, rivers, and mountains in the way.

At first glance it appears to be a simple image of two piece-wise uniform objects, one of which divides naturally into two parts: the stop-sign and pole. The pixels fall within the dynamic range of the camera, the colors are clear, and the noise due to the imaging system is approximately one pixel value out of 256. In terms of the compatibility tests, the borders are reasonably long and contain plenty of information. Finally, the image is not very complex, consisting of only nine regions. Because of these qualities it appeared to be the perfect image for our system and became the first milestone and general test image for all of the system components. We learned very quickly, however, that simplicity is deceptive and perfect real images don't exist.

It turns out that the stop-sign and cup image contains many hidden challenges. These challenges motivated a number of the system design choices and sometimes forced us to backtrack and try new methods. Because of this, it turned out that if the system worked on the stop-sign and cup

image then it would work on most of the other test images as well. This chapter outlines the challenges and the changes we made to deal with this deceptive, but in the end very helpful image.

6.1. Challenging the initial segmentation routine

6.1.1 Choosing a method

The first challenges raised by the stop-sign & cup image occurred during the development of the initial segmentation algorithm. As noted previously, the first initial segmentation method we tried was the linear clustering algorithm of Klinker, Shafer, & Kanade [24]. This method works by first subdividing an image into small blocks and calculating the dimensionality of each block in R, G, B space. It then progressively merges blocks with the same dimensionality so long as the aggregate block has the same dimensionality as the two individual blocks. Finally, it iteratively generates hypotheses based on the largest block groups with linear dimensionality in color space and grows these regions to obtain a more precise segmentation than the initial rough subdivision.

This method works well on uniformly colored curved objects that exhibit significant shading. Images with these characteristics generate multiple linear regions that roughly correspond to objects with some holes where interreflection or highlights occur. However, the stop-sign and cup image presents a different problem. The stop-sign and its letters, in particular, stymied the linear clustering algorithm.

What happens is that the initial image blocks on and around the stop-sign letters include portions of both the white letters and the red sign. Because these two portions of the sign are each uniformly colored and have uniform intensity, blocks containing both of the them are linear in color space, while blocks containing only the red sign or the white letters are only points in color space. This means that when the algorithm selects linear regions as seed regions that both the white and red portions of the stop-sign fall within the linear cluster. This generates a poor initial segmentation and implies that multi-colored piece-wise uniform planar or nearly planar surfaces will cause this particular linear clustering algorithm to fail.

It was because of the linear clustering algorithm's performance on the stop-sign and cup image that we turned to normalized color to obtain good initial segmentations. Of course, the image still had some things to teach us.

6.1.2 Global normalized color threshold

Our initial version of the normalized color segmentation contained only three thresholds: the dark threshold, the size threshold, and the local normalized color threshold. The system used the dark threshold to mask out the background and the local normalized color threshold to determine when two neighboring pixels were "close enough" to be considered part of the same region. The size threshold comes into play after the initial region growing when the system discards regions that are too small.

It turns out that, as uniform as the stop-sign appears, it contains a lot of small-scale texture due to the fact that it is thickly painted wood. The brush strokes are clearly visible in the image if you look closely at it. Because of this small-scale texture, the normalized color on the stop-sign is not constant from pixel to pixel. Given that paint is an inhomogeneous dielectric, this effect is probably due to small highlights on the brush-strokes which locally modify the normalized color.

Therefore, the local normalized color threshold must be reasonably large for the algorithm to classify the red stop-sign pixels as a single region. Note that this problem does not occur with the white letters because the highlights in that case do not affect the normalized color of the surface as the illumination and body colors are similar.

After setting the local normalized color threshold very high, however, the algorithm began merging together different regions of other test images. Basically, if the boundary between two regions changed smoothly enough, the neighboring pixels would all fall within the relaxed local normalized color threshold and the algorithm would miss the region boundary.

This problem motivated the creation of the global normalized color threshold. As described in Chapter 3, this threshold keeps the normalized colors of all of the pixels in a region within a circle centered on the initial seed region. This forces the algorithm to stop growing a region at its boundaries even though the change from one pixel to the next is less than the local normalized color threshold. The global normalized color threshold allows the initial segmentation to be fairly flexible within a region such as the red stop-sign, but forces it to cut-off when the normalized color strays too far from the initial seed region.

Once the region growing algorithm matured, we found yet another stumbling block. The holes in the letters O and P are quite small. The P-hole, in fact, is less than 50 pixels. This forced us to build flexibility into the size threshold and, as we will see later, forced the compatibility tests to deal with very small regions.

6.1.3 Finding border pixels

The next system component to receive the stop-sign and cup test was the border finding algorithm. Unlike most of the other test images, the stop-sign region contains holes where the letters are. As the border finding algorithm only finds the exterior border, this raised the issue of what to do when one region was inside another. It turns out that we ignored the problem in the end. The algorithm that finds border pixel pairs and specifies which regions are adjacent only needs one border to follow. By following around the letters, the system is able to find a set of border pixel pairs for those two regions and specify that the letters and the stop-sign are adjacent. The same is true for the P-hole and the O-hole and their respective encompassing letters.

As noted in Chapter 3, however, the border pixel pairs found by traversing one region are often slightly different than those found by traversing the adjacent region's border. Because of this, when two sets of border pixel pairs exist the compatibility tests use both and average the results. When only one set exists, as with the stop-sign and the letters, the compatibility tests can use only one.

We found that, overall, using one or two sets of border pixel pairs did not make a significant difference in the final segmentation results. The stop-sign and cup image, though, made us think about the issue.

6.1.4 Determining white regions

With the initial segmentation and border polygons in hand we now turned to assigning the initial hypothesis lists. Unlike the two-spheres image, our synthetic test image, and a number of the other real images, the stop-sign and cup image contains white regions. This forced us to define "white" in order to determine which regions would receive the white plastic/white illumination

initial hypotheses. As described in Chapter 3, we define white regions as those regions with an average normalized color that falls in a circle of a given radius centered on white. The stop-sign and cup image helped us to define the size of that circle.

6.2. Analyzing real image data and real objects

The compatibility tests were the next portion of the system to encounter the stop-sign and cup hurdle. In the end, it affected the implementation of all three tests.

6.2.1 Reflectance ratio

The reflectance ratio was the first test we implemented. Initially, we set the reflectance ratio thresholds based upon the two-spheres image. However, the difference between regions of the same object and regions of different objects in that image are quite small. The thresholds appropriate for that image turned out to be too tight for the stop-sign and cup image which, as noted above, contains small-scale texture on the stop-sign. With the relaxed thresholds, the reflectance ratio turned out to be a good performer, although it is prone to false positives along short borders and borders containing little curvature.

6.2.2 Profile analysis

Developing and testing the profile analysis compatibility test was the next step. This development involved a number of decisions, including how to compare different models and whether to compare surfaces or profiles. As described in Chapter 5, we use Rissanen's Minimum Description Length to compare different models, and we chose to compare intensity profiles along border-crossing scanlines. This was the basis for the profile analysis compatibility test, and the basic algorithm worked well on the two-spheres synthetic image. However, the stop-sign and cup test image forced additional modifications.

The first problem we encountered was that the profile analysis did not work well when comparing adjacent planar patches of the stop sign. While it found good models for the individual regions, the combined regions did not line up correctly after the reflectance ratio normalization, causing large errors. An example of the original and normalized profiles from the stop-sign and cup image are shown in Figure 6.2(a), (b), and (c). As we can see from this image, normalizing the white region by the reflectance ratio does not correctly line up the profiles of the two regions, which should form a single plane. This is because of the border pixels identified in Figure 6.2(a) which have a lower intensity than the pixels in the center of the region. Because of this, the polynomial fit to the combined profile has a high order and is not a good model for the step-edge. For this case, the profile analysis returns a value of 0.62. While this is still likely to be a merge, it is a low value for such a simple case.

The problem we found is that the reflectance ratio calculated by the algorithm, while constant along the border, did not accurately reflect the actual albedo difference between the two regions. The problem was that the initial region finding algorithm pushed the edges of the white letter regions to the point where the border pixels contained portions of both the red and white regions. Thus, while they still fell within the normalized color thresholds, their intensity was significantly lower than the white pixels in the center of the region as shown in Figure 6.2(a). It is important to note that at this point the initial region finding algorithm did not include the shrinking stage.

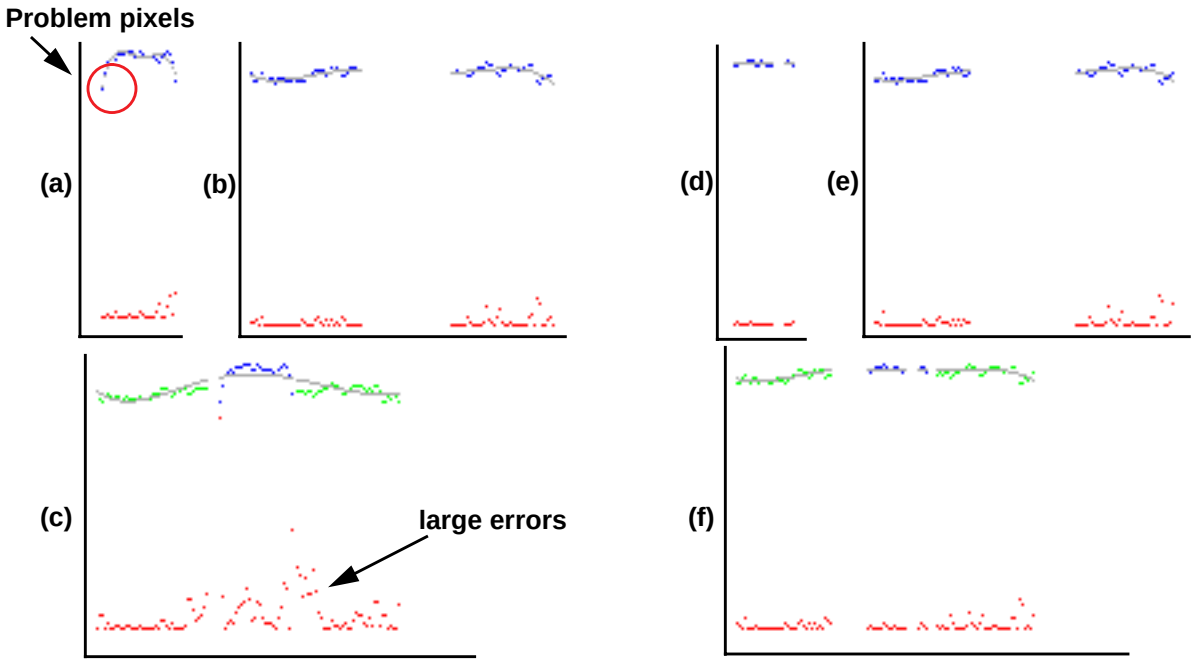


Figure 6.2: (a) Letter profile, (b) sign profile, (c) normalized profile and best-order best-fit polynomial. (d) Letter profile after shrinking, (e) sign profile after shrinking, (f) normalized profile after shrinking. Note that the better reflectance ratio estimate lines up the region profiles more accurately. The highlighted pixels in (a) are the cause of the problem.

The solution to the problem, of course, was to include the shrinking step. We found that shrinking the regions in a single pass through the image removed a sufficient number of the noisy border pixels to allow an accurate calculation of the reflectance ratio. Figure 6.2(f) shows the same profile from the stop-sign and cup image after implementing the shrinking step. Note that the two regions now line up in a single horizontal line, unlike the profile in Figure 6.2(c). The profile analysis returns a value of 1.00 for the stop-sign and letter regions if we use the shrinking step. This is a significant improvement over the 0.62 result obtained without it.

The next issue that came up was that the high-order polynomials were too good at fitting the combined profile crossing the stop-sign and cup. While the planar stop-sign was well-modeled by a line, and the cup by a cubic, their combination profile for some scanline was modeled well-enough by a 5th order polynomial to cause a false positive result. This prompted the idea that for two regions that are part of the same object their combined normalized profile should have similar complexity as the two individual regions. This in turn led to the decision to limit the maximum order of the polynomial model for the combined profile to be equal to or less than the maximum order polynomial used for the two individual regions. This modification improved the performance of the profile analysis compatibility test on the stop-sign and cup image as well as the other test images.

6.2.3 Gradient direction

Finally, we began working on the gradient direction comparison. A simple version of the test worked quite well on the two-spheres image, which is an ideal image for the gradient direction comparison as it contains only curved objects and no texture or imaging noise. The stop-sign and

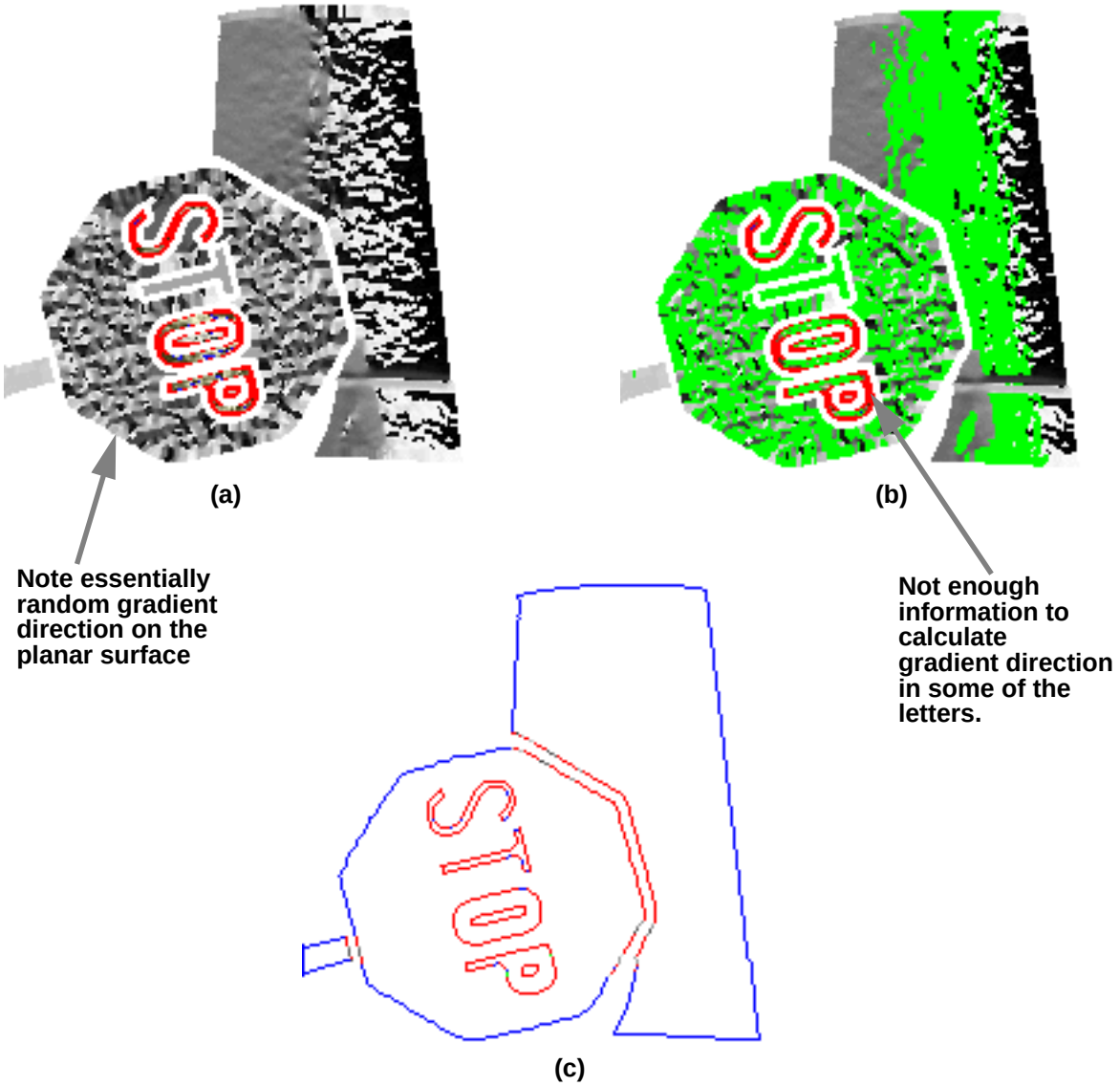


Figure 6.3: (a) Unthresholded gradient direction, (b) thresholded gradient direction, (c) comparison of border gradient directions. Green points in (b) indicate thresholded gradient directions. Red border points in (c) indicate at least one of the two border pixels falls below the threshold.

cup image, however, does not have a single border where both regions contain significant curvature. The gradient direction over the stop-sign and letter regions is ill-defined as the regions are approximately uniform.

As a result of the small intensities and essentially random image gradient directions, the basic test returned negative results for all of the region pairs in the stop-sign and cup image. This prompted the use of the gradient intensity threshold. Figure 6.3(a) shows the unthresholded gradient direction image, and Figure 6.3(b) shows the image with the thresholded points highlighted. Note that this essentially removes the stop-sign from consideration, causing the gradient direction test to indicate that it does not have enough information to make a decision. This is clear if we look at the

number of invalid border gradient directions, as indicated by the red points in Figure 6.3(c). By knowing when the gradient direction is invalid, we can have the weighted averaging scheme discard the gradient direction compatibility test for these cases, resulting in more accurate overall merge likelihoods.

6.3. Extracting segmentations

Just when we thought the system could correctly handle the stop-sign and cup image, it decided to cause yet more problems in the graph generation and segmentation extraction stages.

6.3.1 Graph generation

To set up the hypothesis graph, we want to compare all adjacent hypotheses for similarity. However, in the implementation we assumed that white regions would not be adjacent to one another. After all, if two regions of an image are both white and adjacent, then the initial segmentation algorithm should not separate them and they should form a single region. This turned out to be a false assumption, as pointed out by the stop-sign and cup image.

Prior to the implementation of the shrinking step in the initial segmentation stage, a thin strip of red separated the letters O and P in the image. This produced the correct results because the border pixel pair search around the two letters would always find the red region first. However, the shrinking step removed the thin strip of red.

By removing the strip of red between the two letters, the border pixel pair search along one letter's border would now sometimes find another letter. Because of this, the algorithm classified the two white regions as adjacent.

The end result was that the algorithm punted when asked whether the two regions were a likely merge and it assigned a value of 0.0 to the merge edge in the hypothesis graph. The best final segmentations from this graph specified the red sign, the letters S, T, and O, and the O-hole as a single surface, the pole and cup as single surface, and the P and P-hole as a single surface.

The solution to this problem is straightforward: analyze white hypothesis pairs like any other pairs. The result is the correct segmentation of the image into the stop-sign, pole, and cup.

6.3.2 Handling local minima

The stop-sign and cup image contained one more challenge before admitting defeat. In the initial version of the segmentation system, we did not prefer any hypothesis for any reason. Therefore, the planar-planar hypothesis pairs and curved-curved hypothesis pairs possessed exactly the same merge weights. Because of this, it was a random function as to which of these two alternatives would be at the top of the edge list.

Consider as an example, a subset of the stop-sign and cup hypothesis graph shown in Figure 6.4. The initial sorted edge list contains all six merge edges and the twelve not-merge edges. The best edge connects the sign and the letter T. The planar hypotheses for this region pair happen to be at the top, even though the curved hypothesis pair has the same merge likelihood. After merging the two hypotheses, indicated by the green ellipsoid, the algorithm removes the two curved hypotheses for these regions as shown by the green crosses.

At the beginning of the second iteration, the edge joining curved hypotheses for the O and O-hole

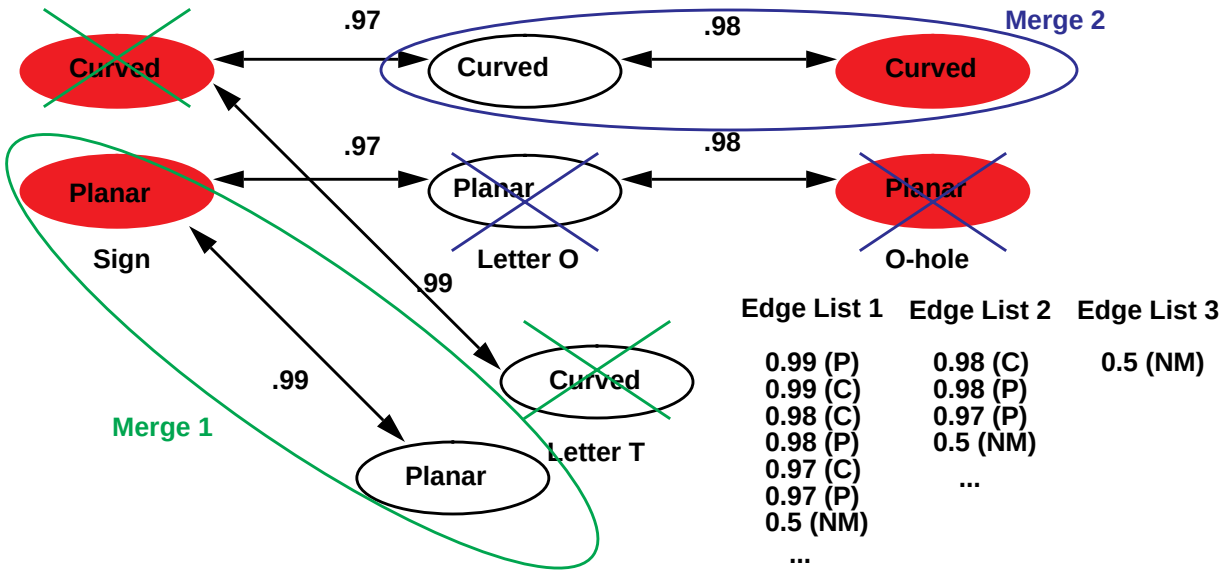


Figure 6.4: Subset of the stop-sign and cup graph. Not merge edges (not shown) connect adjacent planar and curved hypotheses. Also shown are the edge lists at the beginning of each of the three iterations. The third edge list contains only a single not-merge edge.

are at the top of the edge list as shown in Edge List 2. Therefore, the algorithm joins the two curved hypotheses and removes the planar ones as shown by the blue ellipsoid and crosses.

Unfortunately, this leaves only a single not-merge edge in the edge list as the aggregate curved and planar hypotheses are incompatible. Therefore, this graph results in two regions in this particular run, even though the best solution joins all four regions together as either a planar or a curved surface.

It turns out that this problem is not uncommon in the test image set. It happens because of the multilayer nature of the graph and the fact that curved and planar hypotheses don't merge. Also, recall that we run the extraction algorithm on N different graphs where N is the number of hypotheses. In the nth graph, the algorithm forces the nth hypothesis to be in the segmentation by making it the only explanation for its region. If that singular hypothesis is not the same type as the hypotheses selected for the other regions of its object, then a sub-optimal graph results.

The initial solution we found was to prefer the hypothesis--curved or planar--that was similar to the singular hypothesis in that graph. In other words, if the singular hypothesis for a given graph specifies a planar surface, then, if we let E be the set of most likely edges in the sorted edge with similar likelihoods, pick an edge $e \in E$ connecting two other planar hypotheses if one exists.

In the example shown in Figure 6.4, this would mean that if the singular hypothesis were planar, that the algorithm would select the edges connecting the planar hypotheses from both edge lists 1 and 2. Edge list three would then contain two edges: a merge edge with a value 0.97, and a not-merge edge with a value of 0.5. The end result would be a single aggregate region, which is a better solution than we obtained before.

This heuristic keeps the algorithm from falling into local minima so long as we do not prefer one hypothesis over another for a given region. As we will see in Chapter 7, when we start preferring hypotheses we have to move to a more general solution.

Chapter 7: Ranking hypotheses

The output of the system described so far is a set of segmentations which specify both a region grouping for the image as well as the possible high-level explanations for each group. The basic system assigns each of these explanations an equivalent likelihood. Given the discussion in Chapter 2, this may be the theoretically correct procedure, as any of the fundamental hypotheses can generate the same appearance patch.

However, Chapter 2 also notes that certain hypotheses require "weirder" or more improbable components to describe a specific image region. For example, for a planar surface to generate the shading in the two-spheres image either the transfer function must change intensity over the surface, the illumination must be extremely complex, or some combination of the two must occur. However, for a curved surface to generate the two-spheres image requires only a piece-wise uniform object and a single point light source. If we again use the MDL principle to guide our ranking, then we should prefer curved hypotheses over planar hypotheses for the regions of the two-spheres image.

In fact, ranking hypotheses based on their complexity given a specific image region is part of the discussion in Section 2.3. It is true that we cannot rank hypotheses in general except by reasoning about their likelihood in the world. We did this in Chapter 2 when we separated the fundamental hypotheses into the more likely and less likely categories. However, once we have attached a set of hypotheses to a specific image region then we can talk about the complexity of the hypothesis elements required to explain that particular region.

What this chapter describes is a method of implementing this kind of complexity analysis. Specifically, we focus on ranking the curved and planar hypotheses specifying dielectrics under uniform illumination. What this requires is a method of comparing the likelihood that a particular image region was generated by a planar surface to the likelihood that it was generated by a curved surface. To develop this method we would like to make as few assumptions as possible about the image and the scene.

There are three major issues to solve in order to implement a method of ranking hypotheses. The first is how to determine the likelihood that a particular hypothesis generated a particular image region. In this work we focus on the likelihood that a particular region was generated by a planar surface. If this likelihood is high, we prefer a planar hypothesis; if it is very low, we prefer a curved hypothesis. Section 7.1 describes the assumptions and implementation of the test of planarity.

The second issue is how to put this information into the hypothesis graph. Somehow we need to modify the edge values to reflect these likelihoods while not significantly modifying the final region groupings found by the segmentation extraction algorithm. Section 7.2 describes our solution to this problem.

Finally, we have to decide how to update the edges of the hypothesis graph after merging two regions. For example, if the algorithm merges a hypotheses from a region better described by a curved hypothesis with one from a region better described by a planar one, how should it rank the aggregate hypothesis? Section 7.3 deals with this issue.

Section 7.4 then discusses the results of this modification to the system. As we will see, ranking hypotheses turns out to be a powerful tool. The resulting set of segmentations contains more information about the objects in a scene than the basic system described so far.

7.1. A test of planarity

We focus upon the likelihood that a planar surface created a particular image region because it is the most restrictive case in terms of the surface complexity. We base this test of planarity on the assumption that the global illumination for the scene is distant with respect to the extent of the objects. This assumption implies that the angle between the surface normal of a plane and the light source does not change significantly over the surface's extent. If we model the body reflection of dielectrics as Lambertian, which assumes that an object's intensity depends only upon the angle between the light source and the surface normal, then the apparent intensity of a planar surface patch will be constant within a given threshold over it's extent.

We also make the assumption that shadows on a planar surface will be sufficiently dark as to fall under the dark threshold in the initial segmentation method.

The combination of these two assumptions with the system assumption of piece-wise uniform objects implies that a planar surface patch will possess approximately uniform intensity over its extent. On the other hand, a sufficiently curved surface will contain variations in intensity. Clearly, we cannot differentiate between small regions of slowly curving surfaces and regions of planar surfaces, but this observation does allow us to differentiate between surfaces with significant curvature and surfaces approximating a plane.

To make this differentiation in a given image we look at the uniformity of the intensity of each image region. We model a planar surface's appearance with a normal distribution. The assumption is that most of the pixel intensities will fall within a range around the mean intensity. A threshold variance based on the test images defines this range.

However, we cannot use the same variance for each region because of the different relative intensities. For example, the pixels on a dark curved surface may fall within the same absolute range as the pixels on a bright but slightly textured planar surface. What we must do is compare the variances relative to the average brightness of the region.

To accomplish this for a given region the algorithm first calculates the sample mean and variance of the absolute pixel values [29]. It then divides the sample variance by the square of the average pixel value for the region, resulting in a relative variance. The complete formula for the relative variance is given by (1), where $\bar{I}$ is the average pixel intensity and N is the number of pixels in the region.

$$\sigma_{rel}^2 = \frac{1}{\bar{I}^2} \sum_{i=1}^{N} \frac{(I_i - \bar{I})^2}{N-1} \tag{1}$$

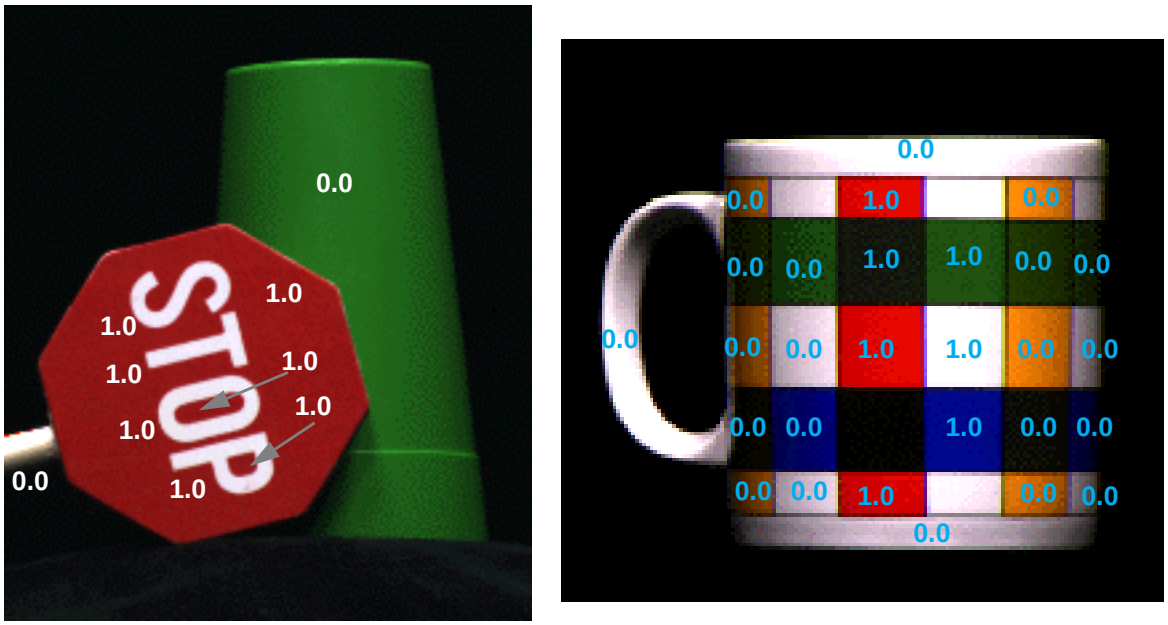


Figure 7.1: Planarity test results on (a) the stop-sign and cup image and (b) the mug image. A high value indicates the region has approximately uniform intensity.

After calculating the relative variance for a given region, we compare it to the threshold variance using a chi-squared test. We specify the threshold variance with respect to the maximum pixel value. To obtain the relative threshold variance we divide the specified threshold variance by the square of the maximum pixel value. The result of the chi-squared test indicates the likelihood that the sample is from the population of surfaces with uniform appearance.

We computed the relative variance of 76 curved regions and 27 planar regions in the ten test images. Except for the images containing the mug, the difference between the relative thresholds of the curved and planar regions is an order of magnitude or more. Based on these results we selected a threshold variance of 180, which translates to a standard deviation of 13.4 pixel values out of 255 and a relative threshold variance of 0.0029. This threshold gives the best possible results on the set of ten test images.

We take the additional step of only making a ranking decision when the results are very strong or very weak. If the chi-squared test returns a value between 0.25 and 0.75, then we claim the results are ambiguous and are not a good basis for ranking the hypotheses. Given the large average difference between the relative thresholds of the curved and planar regions, however, very few of the regions fall within this middle range.

Figure 7.1 shows the results of the test of planarity on the stop-sign and cup image and the mug image. Note that the regions at the center of the mug have high likelihoods of being planar because they have almost uniform intensity.

Overall, this is a simple test. It makes weak assumptions about the image, and yet it allows us to rank the hypotheses currently in the hypothesis list in most situations.

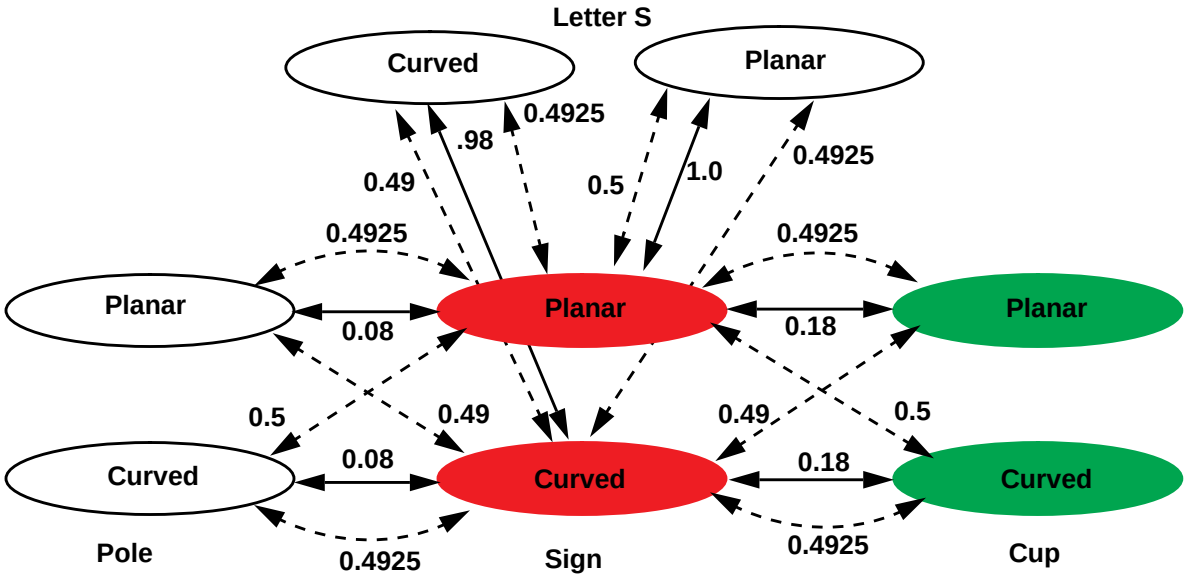


Figure 7.2: Partial hypothesis graph of the stop-sign and cup image with the edges modified according to the likelihood of the hypotheses given the image regions. The solid lines indicate merge edges, while the dashed edges indicate not-merge edges.

7.2. Modifying the hypothesis graph

Once we have the results of the planarity test we want to incorporate this information into the hypothesis graph. To accomplish this we modify the edge values in the graph. In particular we degrade the edge values connecting hypotheses that do not agree with the results of the planarity test.

One alternative to this method of incorporation is to assign values to the hypothesis nodes themselves and use these values combined with the edge values to find the best final segmentations. However, in order to use the same segmentation extraction algorithm, which requires us to sort the edge values to find the best edge, we would still have to modify the edge values according to the node weights prior to sorting. Therefore, we decided to simply modify the edge values directly.

After running the compatibility tests on each hypothesis pair we also run the planarity test on each hypothesis' region. If both hypotheses match the results of the planarity test on their respective regions, then we do not modify the edge value connecting the two. If one of the hypotheses does not match its planarity result, the algorithm reduces the edge value connecting them by multiplying it by 0.985. If both the hypotheses disagree with the planarity results, the algorithm reduces the edge value by multiplying by 0.98. This gives a hierarchy of edge values, penalizing most the edges connecting the hypotheses that disagree most with the planarity test.

While this may not seem like a significant change in the edge values, recall that for each pass the highest-probability first algorithm sorts the edges by value and chooses the best edge in the graph. Therefore, these slight differentiations in edge values cause the algorithm to prefer some hypotheses over others without significantly changing the overall likelihood of the various image interpretations.

As noted above, we define disagreement so that only strong disagreement leads to an edge modification. If a hypothesis is planar then we penalize its corresponding edge if the planarity result is less than 0.25. Likewise, if a hypothesis is curved then we penalize its edge if the planarity result is greater than 0.75. This leaves alone edges between ambiguous regions.

Figure 7.2 shows a partial hypothesis graph for the stop-sign and cup image with the updated weights. Note that the edges connecting the more likely hypotheses now have larger values than edges connecting the less likely hypotheses. In terms of the segmentation extraction algorithm, this means the edges connecting the more likely hypotheses will be higher in the sorted edge list.

7.3. Modifying the segmentation extraction algorithm

In addition to the initial edge weight modification, we also need to dynamically update the edges as the algorithm merges hypotheses into larger clusters. We not only need to determine the planarity of new aggregate hypotheses, but also modify the not-merge edges on the fly to reflect the preferences of the planarity test.

The first question is how to define the planarity of an aggregate hypothesis. We decided to average the planar likelihoods of the individual hypotheses. The algorithm then uses this average result to modify the new edge values calculated for the aggregate hypothesis. These modifications take place exactly as described above; the algorithm penalizes region pairs where one or both hypotheses disagree strongly with their planarity likelihood by multiplying the appropriate edges by 0.985 and 0.98, respectively.

It turns out that this modification of the hypothesis graph causes the segmentation extraction algorithm to fall into local minima in a number of cases. The mug image provides a good example of the problem. As noted previously, the regions near the center of the mug are almost uniform in intensity. As a result, the algorithm degrades the curved hypotheses for those regions. Likewise, the regions near the edge of the mug show significant curvature and the algorithm degrades the planar hypotheses for those regions.

What ends up happening is that the segmentation extraction algorithm splits the body of the mug into two regions, one planar and one curved. This situation is shown on the left of Figure 7.3. This, unfortunately is a local minimum. There are at least two solutions that are better: the body of the mug as a single planar object, and the body of the mug as a single curved object. In both cases the handle remains a separate curved region. For the mug, the segmentation specifying it as a curved handle and a single curved body turns out to the be global optimum because more of the regions display curvature than display uniform intensities. However, because the segmentation extraction algorithm is only step-wise optimal it cannot find this solution without help.

Somehow, we have to perturb the state of the algorithm and start it again. We don't want to lose the information contained in the region groupings found so far, but we want the algorithm to consider other explanations for those region groupings.

Our general solution is to have the computer re-attach the possible explanations for the different region groupings and then run the algorithm again. For example, as noted above in the first pass the segmentation extraction algorithm divides the mug into two regions, one planar and one curved. Now let's reattach an aggregate curved hypotheses and an aggregate planar hypotheses to the planar region grouping, and let's do the same for the curved region grouping. This situation is

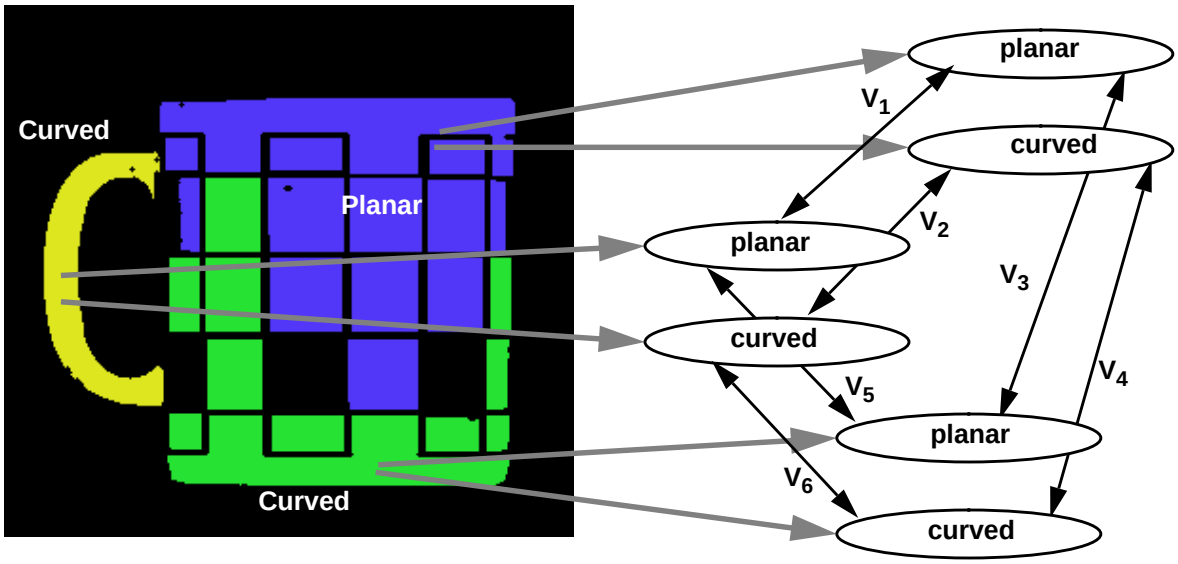


Figure 7.3: Results after the first segmentation extraction pass. We perturb the situation by re-attaching hypotheses to the aggregate regions and making a second pass.

shown at the right of Figure 7.3. The algorithm calculates the weights joining the aggregate regions by averaging the appropriate hypothesis edges and then modifying those edges according to the average result of the planarity test on all of the hypotheses in each aggregate region. By running the algorithm again, it is able to consider joining the two curved explanations for the two mug body regions. It can also explore the two planar explanations.

Essentially, what this procedure does is bump the algorithm out of a local minimum and give it the opportunity to keep searching for a better solutions. For the case of the two initial hypotheses currently implemented, there will always be two possible interpretations for each aggregate region. We can generalize the algorithm to situations with more hypotheses per region by searching the space of alternative interpretations for each aggregate region after the first pass. Then we re-attach the most likely alternative interpretations and begin the modified highest-probability first algorithm again. With more layers in the graph, this process might have to be repeated more than once. Exploring this topic, however, is an area of future research.

Note, however, that we still maintain the singular hypothesis in each graph, which means that one of the aggregate regions will have only a single explanation. With respect to the mug image and the graph in Figure 7.3, this means that one of the three regions will still have only a single explanation. If the singular hypothesis is part of one of the mug body aggregate hypotheses, then this explanation will be more likely to join with the aggregate hypothesis that proposes a similar shape. Therefore, in the mug image the singular hypothesis determines whether the optimal segmentation of the mug body is curved or planar.

7.4. Analysis of results

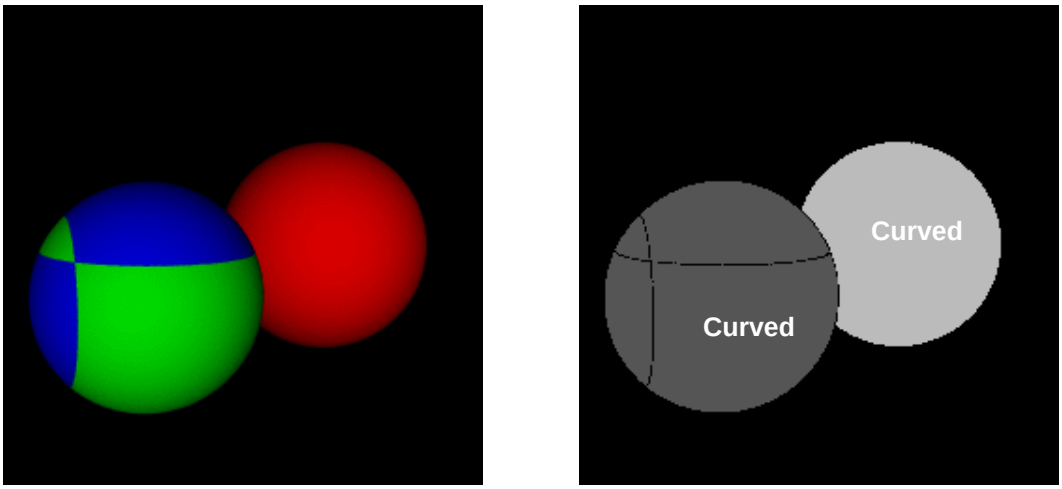
The following figures show the results of the modified system on the set of ten test images. In nine of the ten cases, the most likely interpretation of the image matches the actual objects. In the cup-

plane image in Figure 7.7, however, the most likely interpretation of the small region of the plane behind the mug handle is curved. The second most likely interpretation of the entire image specifies that this portion of the plane is planar.

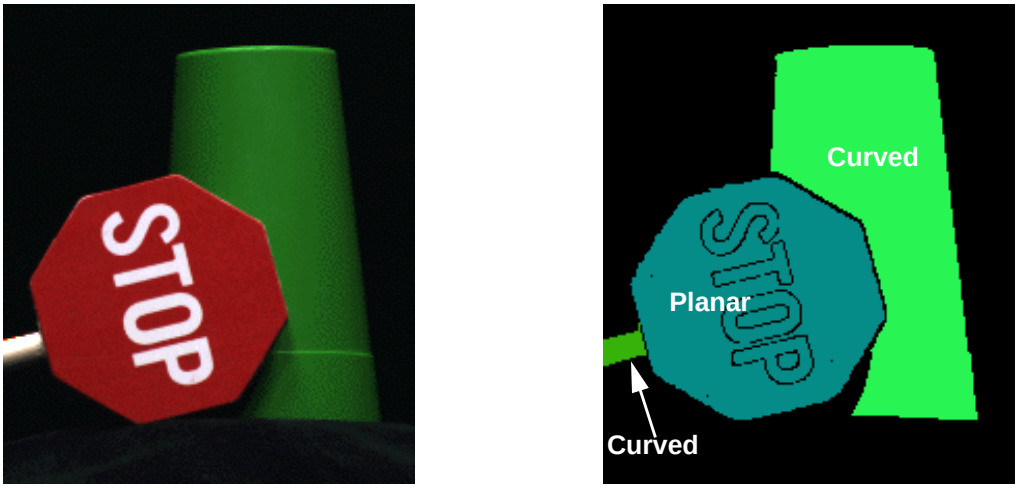
The reason the system rates the curved hypotheses higher is because of the shadows generated on the plane by the mug. These shadows cause some variation in the intensity of the planar regions, but do not all fall under the dark threshold. Thus, this image breaks one of the assumptions underlying the planarity test.

This case is not a failure, however, but actually shows the strength of our approach to segmentation. While the system did not select the set of hypotheses that most closely matched the objects as the most likely segmentation, the system returns more than one segmentation. Because we force each hypothesis for each region to be in at least one of the segmentations returned by the system, the system considers and ranks a number of different image interpretations. In this case, the system ranked the set of hypotheses most closely matching the objects as the second most likely interpretation. Note that the region groupings returned in both cases are the same.

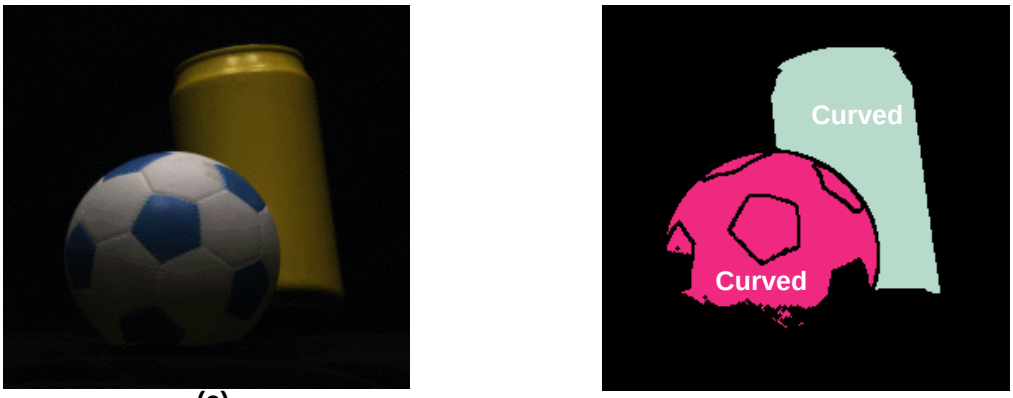
The importance of this feature is that any algorithm using the result of our approach to segmentation can pick and choose which interpretation to work with. If the algorithm has more knowledge about the scene or the objects in the scene, then it can pick the interpretation that most closely matches this knowledge.



(a)



(b)



(c)

Figure 7.4: Segmentation results for the two-spheres, stop-sign and cup, and ball-cylinder images. The text indicates the most likely shape interpretation for each image.

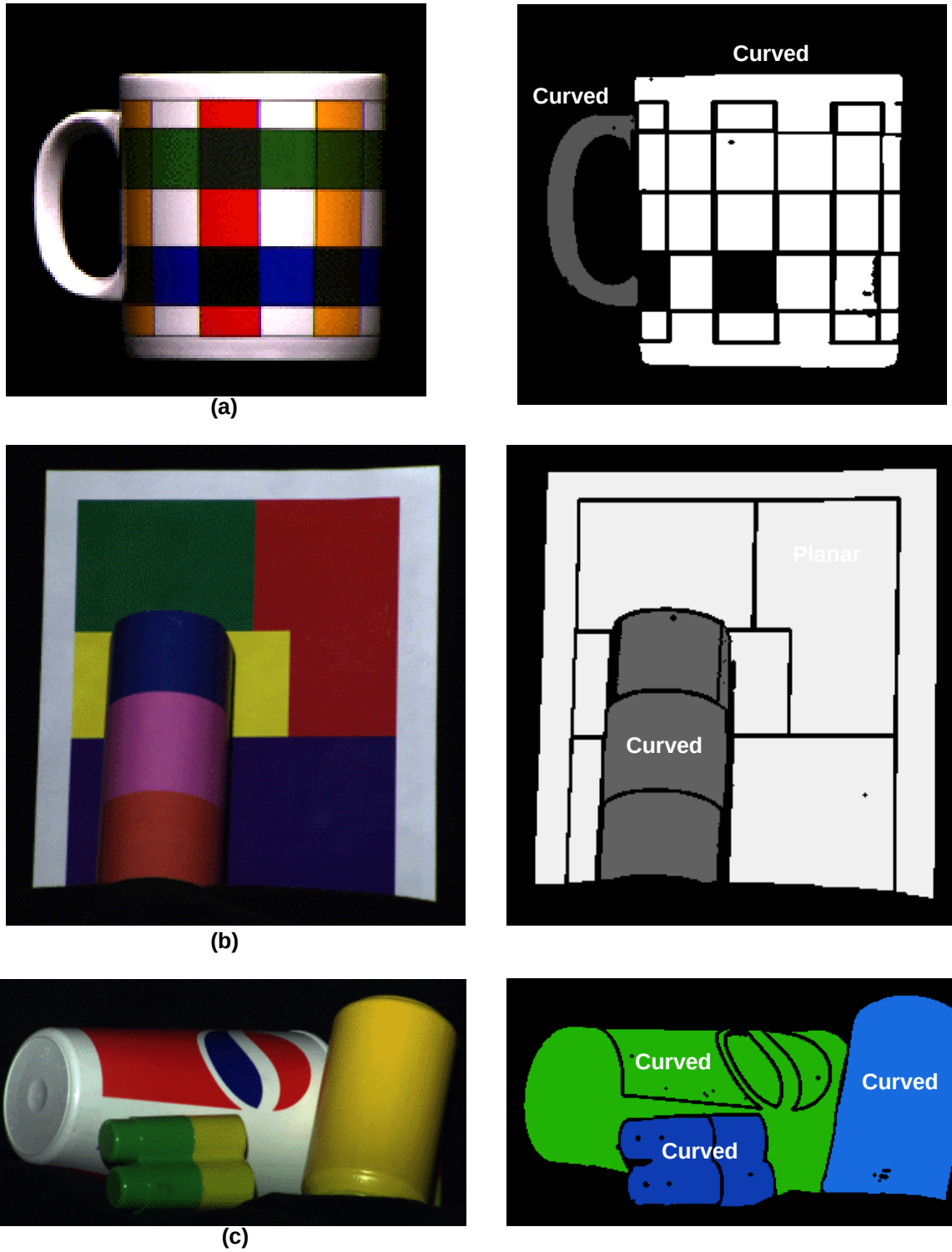


Figure 7.5: Segmentation results for the (a) mug, (b) cup-plane, and (b) pepsi images. The text indicates the most likely shape interpretation for each image.

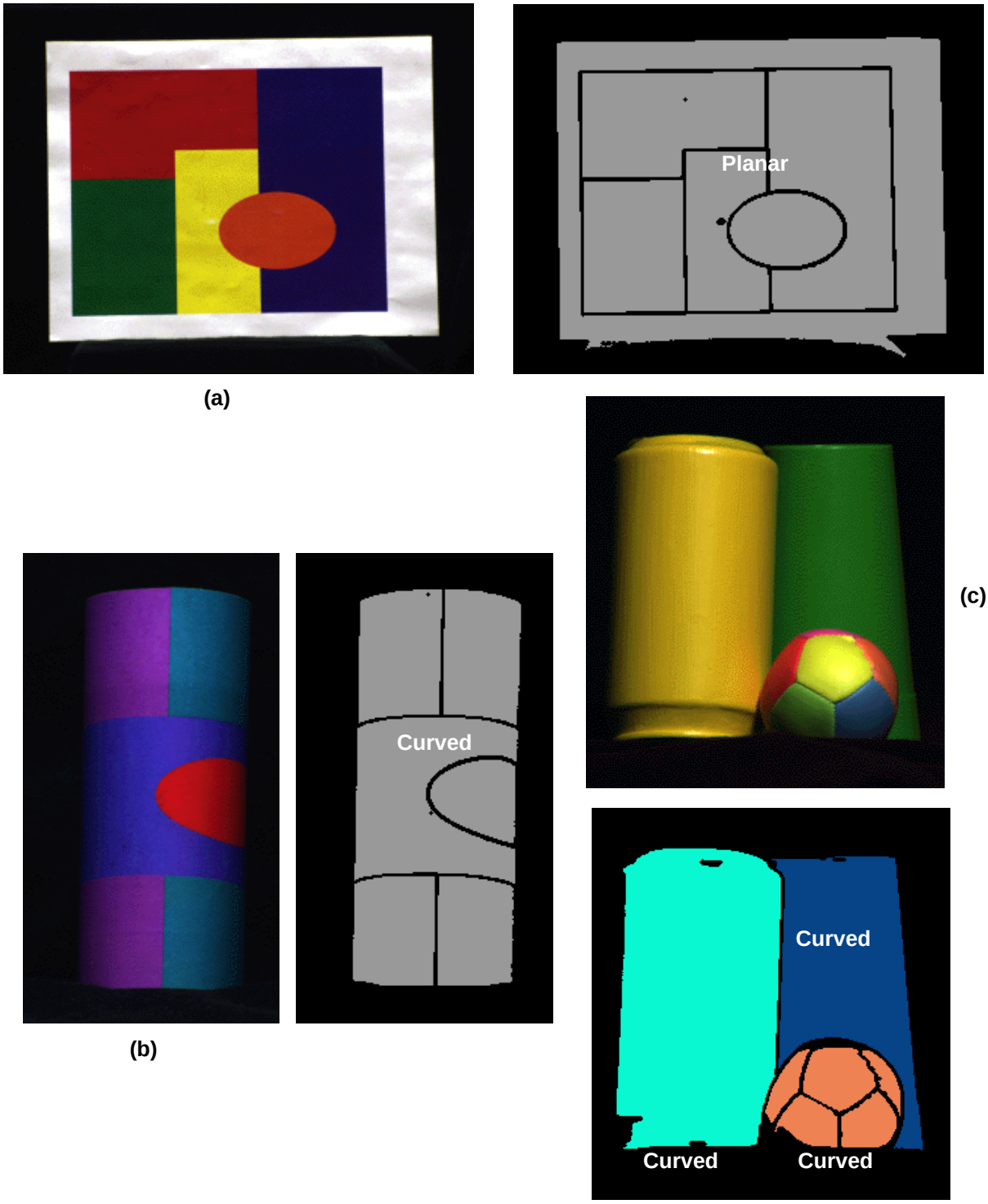


Figure 7.6: Segmentation results for the (a) plane, (b) cylinder, and (c) two-cylinder images. The text indicates the most likely shape interpretation for each image.

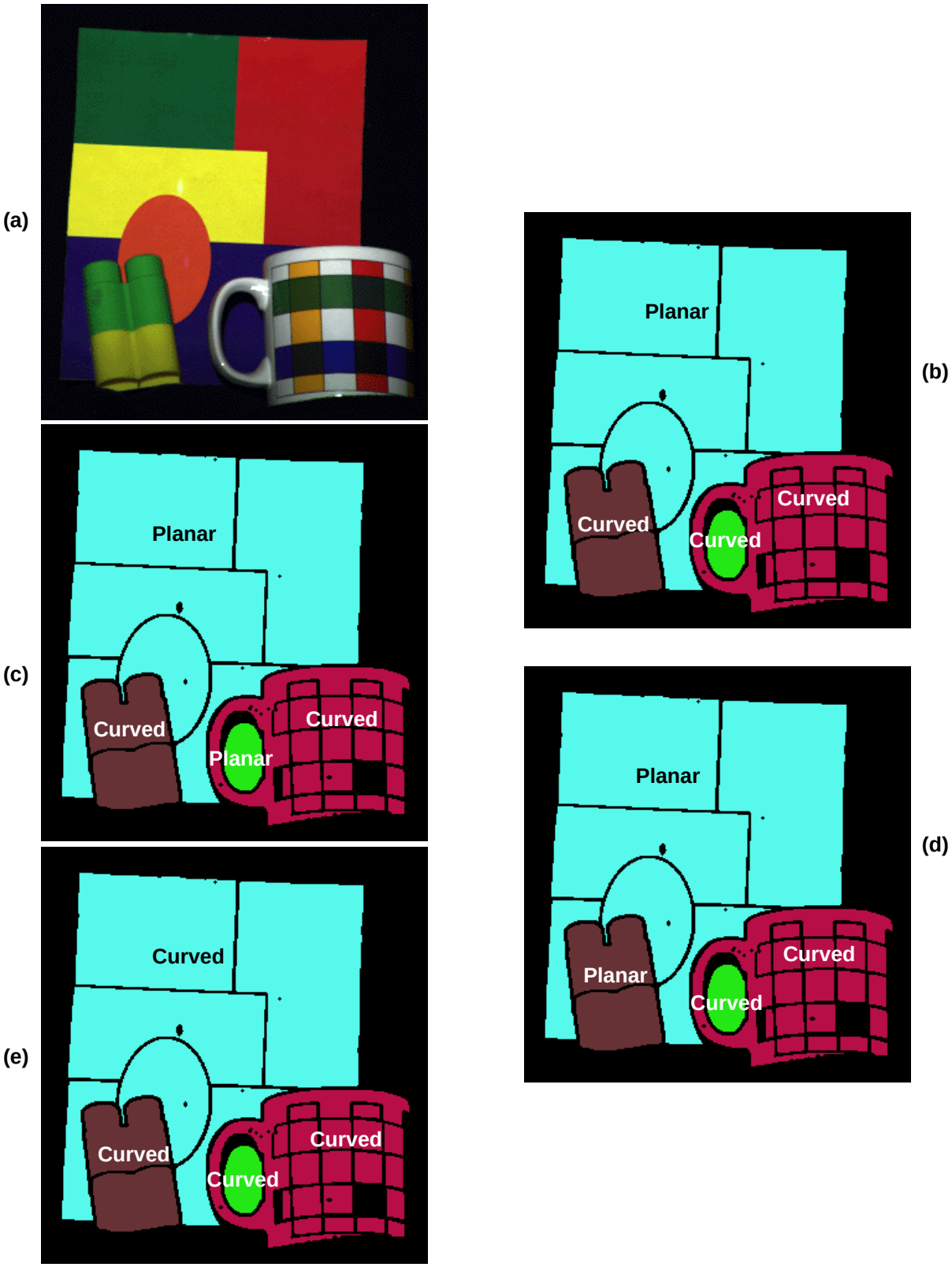


Figure 7.7: The top five results for the plane-cylinder image. (b) The most likely shape interpretation of the image, and (c) the second most likely shape interpretation, (d) third most likely interpretation, (e) fourth most likely interpretation.

Chapter 8: Expanding the initial hypothesis list

So far, our implementation has been limited to two hypotheses per region and only three of the boxes in the hypothesis merger tables from Chapter 2. One of the strengths of our approach to segmentation, however, is that the same general system can handle numerous hypotheses.

This chapter explores the issues associated with expanding the initial hypothesis list. This includes developing new tests for hypothesis compatibility and integrating the results of these tests into the hypothesis graph. Sections 8.1 and 8.2 discuss the former, and section 8.3 presents the latter issue. We explore these issues by expanding the initial hypothesis list to include (Colored Dielectric Displaying Surface Reflection, White Uniform, Curved) when appropriate. Section 8.4 shows the results of the expanded system on a set of test images with specular highlights.

By adding this hypothesis to the initial hypothesis list, the system is able to handle images containing multi-colored objects displaying specularity, or highlights. The ability to handle specularity moves the system beyond previous physics-based segmentation algorithms because it can not only correctly merge the highlight regions with their body reflection counterparts, but also correctly merge multi-colored objects into coherent surfaces.

8.1. Characterization of specular regions

The key to developing a compatibility test for regions displaying surface reflection and their body reflection counterparts is to understand the relationship of these two types of reflection. As discussed in Chapter 2, a number of models exist for both types of reflection [3][15][40][57][60]. Rubin & Richards were perhaps the first to study of the relationship of physical properties and appearance in color imagery [52]. However, it was Shafer who first proposed the physically-based dichromatic reflection model [53]. Klinker, Shafer, and Kanade then refined this model and used it to identify highlight regions in an image and the surfaces to which they belonged [24]. The dichromatic reflection model and the resulting relationship of body and surface reflection also inspired other physics-based vision algorithms and applications [2][7][16][43]. In particular, Novak undertook a detailed parameterization and characterization of the relationship between body reflection and surface reflection and was able to determine light source color, body color, and relative surface roughness from an object's appearance.

The dichromatic reflection model and the work of Klinker and Novak in characterizing and refining that model provides a body of knowledge upon which we can build a compatibility test [25][43]. There are a number of characteristics of this relationship that we can measure. We approach this task in the same manner as the other compatibility tests, looking for characteristics that must be true for two regions to have the surface/body reflection relationship and be part of the same object.

To this end, we now give a brief description of the appearance of a highlight region on an inhomo-

geneous dielectric in terms of its characteristic shape in color space. Based on this characteristic shape we then describe a set of attributes we can measure and test. We conclude this section by making several observations about a highlights appearance in the image and how this adds additional constraints to the problem.

As described by Klinker, Shafer, and Kanade, the colors of the surface reflection within a highlight and the body reflection around it form a skewed-T in color space [24]. This space can also be thought of as a 3-D histogram, where a bucket is filled if that color exists in the image.

Figure 8.1(a) shows an image of a plastic egg with a highlight in the blue region, and Figure 8.1(b) shows its initial segmentation. Figure 8.1(c) shows the 3-D histogram of the blue region, the highlight region, and the pixels in between them.

From Figure 8.1(c) we can identify three components of the skewed-T. The first component is the body reflection vector. This vector represents the normalized color of the body reflection and is one of the two major vectors forming the skewed-T. Under white light this is the equivalent of the normalized color of the object.

The second component is the surface reflection that falls within the dynamic range of the camera. The surface reflection forms the second major vector of the skewed-T in color space. Because the surface reflection does not involve the pigment particles in an inhomogeneous dielectric, the light source color defines the vector's direction. Note that some of the pixels in the surface reflection vector fall within the blue region because of the finite size of the local and global normalized color thresholds used in the initial segmentation.

The third component of the skewed-T is the color clipped portion of the surface reflection vector. This portion follows the edge of the color cube defined by the dynamic range of the camera. Because surface reflection will often exceed the dynamic range of one of the camera colors, any algorithm that deals with highlights must take this into account in some manner.

The important characteristics of the skewed-T are as follows. First, the body and surface reflection of an single color surface form a plane in color space. The body color and light source color define this plane. Second, most of the pixel colors lie very close to either the body or light source vectors. Third, these two vectors meet in a well-defined manner, and, finally, the camera usually clips the colors of some of the surface reflection pixels.

It is important to explain the "well-defined manner" in which the body and surface reflection vectors meet. Put simply, these two vectors must intersect in the upper 50% of the body reflection vector, where the pixel exhibiting the maximum intensity body reflection defines the upper end of this vector and the black point (0,0,0) defines the lower end. Klinker initially proposed this rule as a heuristic, and the rule was later shown to be theoretically correct by Novak [25][43].

The 50% rule immediately suggests several characteristics to test for compatibility. We can look at the minimum and maximum intensity values in the hypothesized highlight region and compare them to the body reflection intensities. We can also measure whether the body and surface reflection vectors intersect in the upper 50% of the body reflection vector. In section 8.2 we develop these tests further.

In addition to the characteristics of the body and surface reflection in color space, we can also look at the attributes of a highlight in the intensity image. If we assume a distant viewer, the Lam-

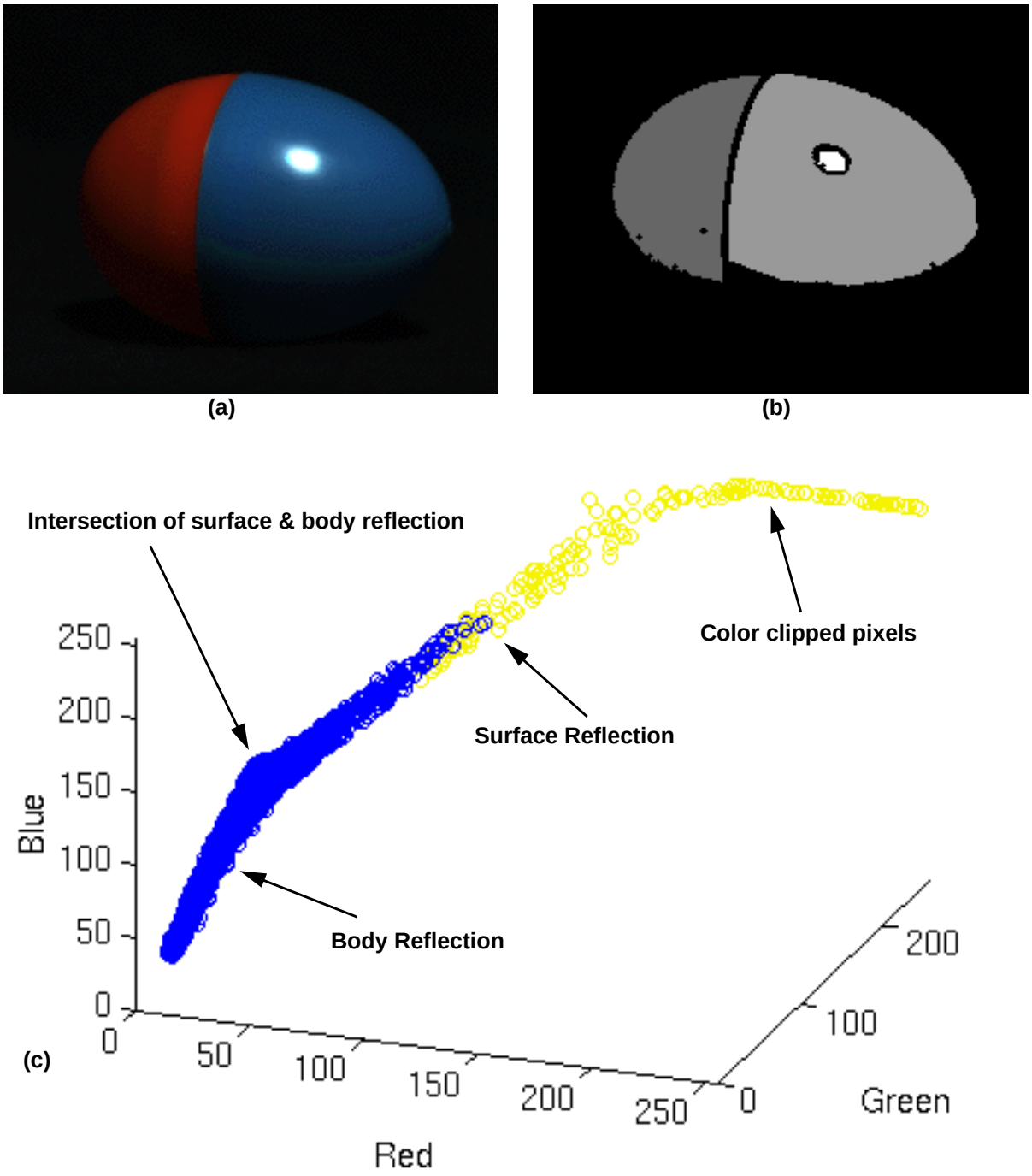


Figure 8.1: (a) Picture of a plastic egg with a highlight. (b) Initial segmentation of the egg image. (c) Histogram of the blue region, the highlight region, and the pixels between them. The blue region is shown in blue, the highlight region and surrounding pixels in yellow.

bertian model for body reflection, and the Torrance & Sparrow model for surface reflection, then we can qualitatively describe the appearance of a highlight on a curved or planar inhomogeneous dielectric surface.

First consider a curved dielectric surface under white uniform illumination with a highlight such

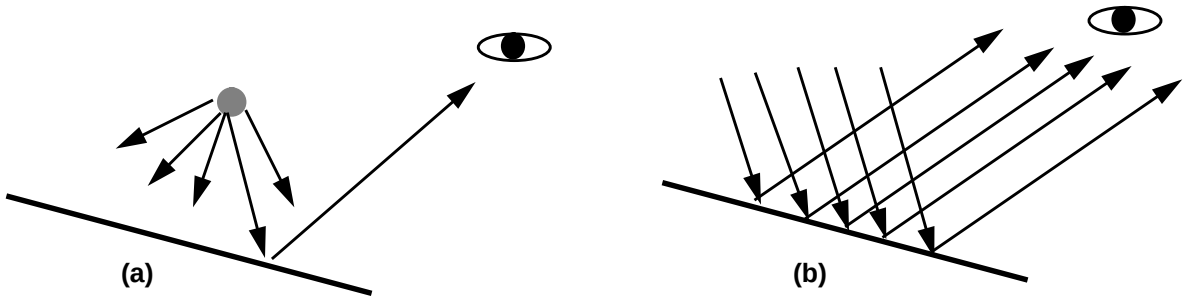


Figure 8.2: (a) Proximal light source, (b) distant light source. For a distant viewer the proximal light source produces a highlight of limited extent and the rest of the plane varies in intensity. The distant light source generates either an extensive highlight or an area of uniform intensity.

as the egg in Figure 8.1(a). Considering only the body reflection, we can describe it as having varying intensity according to the angle(s) between the surface and the light source(s). As the hypothesis proposes uniform illumination this light is directional, unlike a diffuse illumination environment. Whether or not the light is close to or far from the surface, therefore, the body reflection will vary in intensity. Furthermore, because of the varying surface curvature, directional lighting, and limited extent of the surface reflection phenomenon, the highlight will not cover the entire surface. The egg is one example of this situation.

Now consider a planar dielectric surface under white uniform illumination with a highlight. There are two sub-cases within this hypothesis as shown in Figure 8.2(a) and (b). First, if the light source is close to the plane, then there will be variation in the plane's body reflection intensity due to the changing angle between the surface and the light source. Furthermore, the highlight will be bounded to a finite area of the plane because the angle between the planar surface and the nearby light source changes rapidly across the surface. This case is similar to the situation with a curved surface described above.

In the second sub-case shown in Figure 8.2(b) the light source is far away from the plane. This implies that there will not be any variation in the plane's body reflection intensity because the angle between the planar surface and the light source does not change. This also implies that if there is a highlight on the plane, it will have a large extent and most likely cover the entire planar patch in the image, overriding the body reflection vector.

The result of this thought experiment is that in the two cases where the highlight has a small finite extent the adjacent body reflection region varies in intensity. When the highlight has a large extent, however, it is unlikely there will be an adjacent body reflection region if the viewer is reasonably distant from the scene. Therefore, the test for uniform intensity described in Chapter 7 becomes a useful tool. If a proposed body reflection region does not vary in intensity, then it is unlikely that the corresponding proposed surface reflection region is actually surface reflection, no matter what the other tests indicate.

8.2. A test for highlight regions

The characteristics described in the previous section are the basis for the highlight compatibility test. The test takes as input a proposed highlight region and a proposed body reflection region and returns a likelihood that they fit the skewed-T model. The characteristics previously described, however, require some processing to find. In particular, we have to deal with the color clipping

present in the highlight region as shown in Figure 8.1(c). This section describes the details of the processing required to find the compatibility measure.

Before beginning any processing, however, the compatibility test makes an initial check of the uniformity of the proposed body reflection and highlight regions. This check is based upon the observations made at the end of the previous section. If the sum of their likelihood of uniformity is greater than or equal to 1.0, then the compatibility test returns a low likelihood of a merge. This likelihood of uniformity, or planarity, is the same test used and described in Chapter 7.

After testing for uniformity, the first step in the compatibility test is to grow the highlight region to include any bright pixels that lie between it and the proposed body reflection region. This step is necessary because the initial segmentation algorithm is based upon normalized color. The pixels in between a highlight and body reflection region change normalized color quickly and the initial segmentation algorithm will not, in general, group them with any region. Figure 8.1(b) is a good example of this.

These pixels, however, include a significant amount of information in the histogram. They form a significant portion of the surface reflection vector. In particular, the portion nearest to the base region. Therefore, we want to include them in the highlight region.

To grow the highlight region we first do a grassfire transform on an area of the image around the highlight region. This area is 10 pixels larger in all directions than the bounding box of the highlight region. The grassfire transform labels pixels within this area that are not already part of any region with a number indicating their 4-connected distance from the highlight region.

Using the results of the grassfire transform, the algorithm then adds to the highlight region all of the pixels in the area around the it that have an intensity greater than 30 out of 255 and which have a grassfire value less than 10. Growing the region this way limits the extent of the growth to pixels close to the highlight region. In the egg image the growth step adds all of the pixels between the highlight and blue regions to the highlight region. The histogram of these pixels is shown in yellow in Figure 8.1(c).

The next step in the process is to find the body reflection and surface reflection vectors in color space. To find the body reflection vector the algorithm first generates the covariance matrix of the body reflection region pixels and then uses singular value decomposition [SVD] on the covariance matrix to find the primary vector of the pixels in color space. Note that we do not take any special measures to cut out the pixels in the body reflection region that lie partly on the surface reflection vector. As the vast majority of the pixels in the body reflection region fall on the body reflection vector, the pixels on the surface reflection vector do not strongly affect the primary vector returned by SVD.

Finding the surface reflection vector, however, requires more care as the color clipped pixels form a significant portion of the highlight region. To get an estimate of the surface reflection vector the algorithm first finds the covariance matrix of the highlight region pixels that fall completely within the dynamic range of the camera. The primary vector from the SVD of this matrix then provides an estimate of the surface reflection vector.

The next step in the process is to “un-clip” the pixels that go beyond the dynamic range of the camera. This problem was originally identified and solved by Klinker in her work on physics-based segmentation [25]. The method is based on the following observation: given the estimate of

the surface reflection vector we can project the clipped pixels onto this vector so long as at least one of the red, green, or blue color values falls within the dynamic range of the camera. Given the surface reflection vector and a pixel with at least one color value within the dynamic range, we can calculate the appropriate distance along the vector S for that pixel using (1),

$$d = \frac{c_i - \bar{c}_i}{s_i} \tag{1}$$

where c_i is the pixel value of the color i that falls within the dynamic range, $\bar{c}_i$ is the average value of color i in the highlight region, and s_i is the ith element of the principal surface reflection vector.

After calculating d, which is the distance along the principal surface reflection vector from the average value at which the pixel should be, we can calculate the corrected pixel values. Given d, the corrected value for color c_i is given by (2).

$$c_i = \bar{c}_i + ds_i \tag{2}$$

If two colors fall within the dynamic range then the algorithm uses (1) to calculate d values for both of the colors and uses the average of the two d values in (2). The algorithm discards all pixels with the maximum value for all three colors in the highlight region.

Figure 8.3 shows the histogram of the corrected pixels. Note that the range of the pixel values is now beyond the dynamic camera range.

Prior to finding the intersection point of the body and surface reflection vectors, the algorithm maps all of the pixels onto the plane defined by those two vectors. It then recalculates the primary vectors for the surface reflection and body reflection using the two-dimensional color corrected points. The algorithm finds the primary vectors using SVD on the 2x2 covariance matrices.

Given these two lines on a plane, it is straightforward to calculate their intersection point. A more difficult task is how to ascertain the length of the body reflection vector so that we can calculate where on the body reflection vector the intersection takes place. In order to use the 50% rule, we need to know if the intersection takes place in the upper half of the body reflection vector.

What complicates this task are the pixels in the body reflection region that display some surface reflection. These pixels tend to be brighter than pixels displaying only body reflection. Therefore, if we consider these pixels when calculating the length of the body reflection vector then the vector will be too long. This may cause the calculated intersection point to fall in the lower half of the body reflection vector even in actual highlight/body reflection cases.

Klinker also had to deal with this problem [25]. Klinker's algorithm modeled the body reflection region as a cylinder in color space. The highlight vector points away from one side of this cylinder. To avoid using pixels containing surface reflection Klinker used the brightest pixel on the side of the cylinder opposite the highlight vector to estimate the length of the body reflection vector. We use a similar solution.

The algorithm has already estimated the primary vectors for the body and surface reflection. These two vectors form a plane upon which the algorithm projects all of the pixel values. To cre-

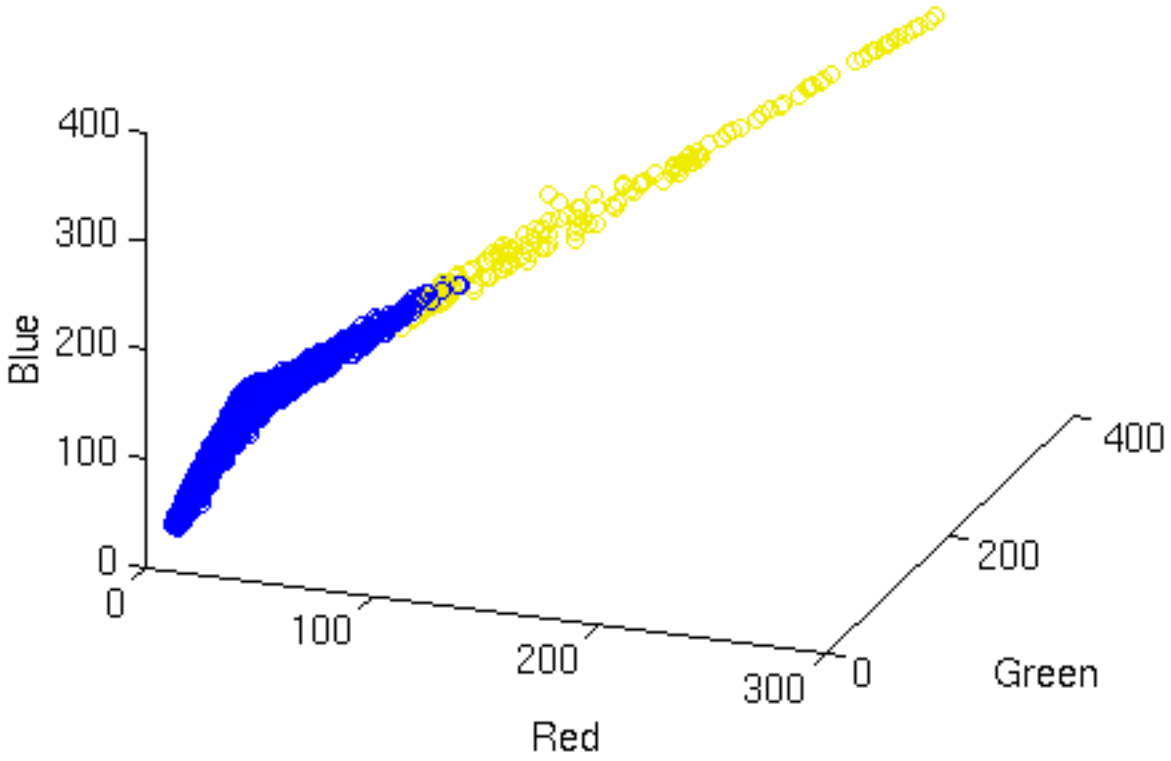


Figure 8.3: Histogram of pixel values for the blue and highlight regions after correcting for color clipping. The blue region pixels are shown in blue, the highlight pixels in yellow.

ate this plane the algorithm defines a coordinate system with the body reflection vector as one axis and a vector perpendicular to it and pointing in the highlight vector direction as the second axis. This means that all pixels in the body reflection region on the opposite side of the body reflection vector from the highlight vector will have negative values for their second coordinate. Therefore, the algorithm uses the maximum intensity point with a negative or zero-valued second coordinate to estimate the length of the body reflection vector. This gives a length value uncorrupted by the pixels which contain some surface reflection.

Finally, given the length of the body reflection vector, we can calculate where on that vector the intersection takes place. If the intersection takes place in the upper half of the vector, and no further out than the estimated length of the vector, then this test does not argue against the compatibility of the proposed body reflection and highlight regions.

In addition to the intersection test, we use three other characteristics of the body reflection and highlight reflection to argue against compatibility. First, given the 50% rule, all of the proposed highlight region pixels must be brighter than a pixel half the intensity of the maximum body reflection pixel as measured above. If it is not, then the compatibility test returns a low likelihood for the proposed body reflection and highlight regions.

Second, given that all of the pixels in the highlight region must be brighter than half the brightness of the brightest body reflection pixel, the average intensity of the highlight region should be greater than the average intensity of the body reflection region. If it is not, then the compatibility

test returns a low likelihood.

The final observation is that the brightest body reflection pixel must be less than or equal to the brightest surface reflection value. Again, if this is not true the algorithm returns a low likelihood.

If the proposed body reflection and highlight regions pass all of these requirements, then the algorithm needs to return a likelihood of a merger for the hypothesis pair. Given that it is unlikely for a region pair that is not a base region and a highlight region to have these characteristics, we would like the likelihood to be high.

Given d_x, the distance along the body reflection vector to the intersection point, the algorithm uses

$$L(merge) = 1 - (d_x - 0.75)^2 \quad (3)$$

to obtain the likelihood of a merge. This means that in the worst case the likelihood will be 0.9375 if the algorithm passes all of the tests described above. Otherwise, the algorithm returns a likelihood of $1x10^{-7}$.

This completes the compatibility test which compares a hypothesis proposing (Colored Dielectric Displaying Surface Reflection, White Illumination, Curved | Planar) with a hypothesis proposing (Colored Dielectric, White Illumination, Curved | Planar). The test returns a likelihood that the two hypotheses are part of the same surface and should be merged.

8.3. Integrating a specular hypothesis into the system

Given the compatibility test, we now have to integrate the new hypotheses into the system. We have to decide when to attach the specular hypothesis to a region, how to deal with special cases, and how to assign values to the not-merge edges in the graph. If we use a ranking system as described in Chapter 7, we have to somehow rank the specular hypotheses in comparison to the others.

We first consider the issue of when to attach the specular hypotheses to a region. Recall that all of the initial hypotheses in the basic system as well as the specular hypotheses being considered propose white uniform illumination. Recall also that surface reflection on inhomogeneous dielectrics is the color of the illumination. Therefore, highlight regions should be white. This implies that we only need to attach specular hypotheses to regions the initial segmentation algorithm classifies as white.

In practice this is true so long as the central pixels of the highlight regions are close to the upper bounds of the camera's dynamic range. In the egg image, for example, the algorithm classifies the highlight region as white. It turns out that highlight regions that fall within the camera's dynamic range change normalized color sufficiently quickly that the initial segmentation algorithm does not group those pixels with any region. This is a shortcoming of the initial segmentation algorithm, but it allows us to be confident in assigning specular hypotheses to only white regions.

In this exploration of expanding the initial hypothesis list we also assume that specular hypotheses will not be adjacent to one another. Any merge edge connecting adjacent specular hypotheses by default receives a low likelihood value.

To handle the additional hypotheses, we modified the ranking system described in Chapter 7 to be

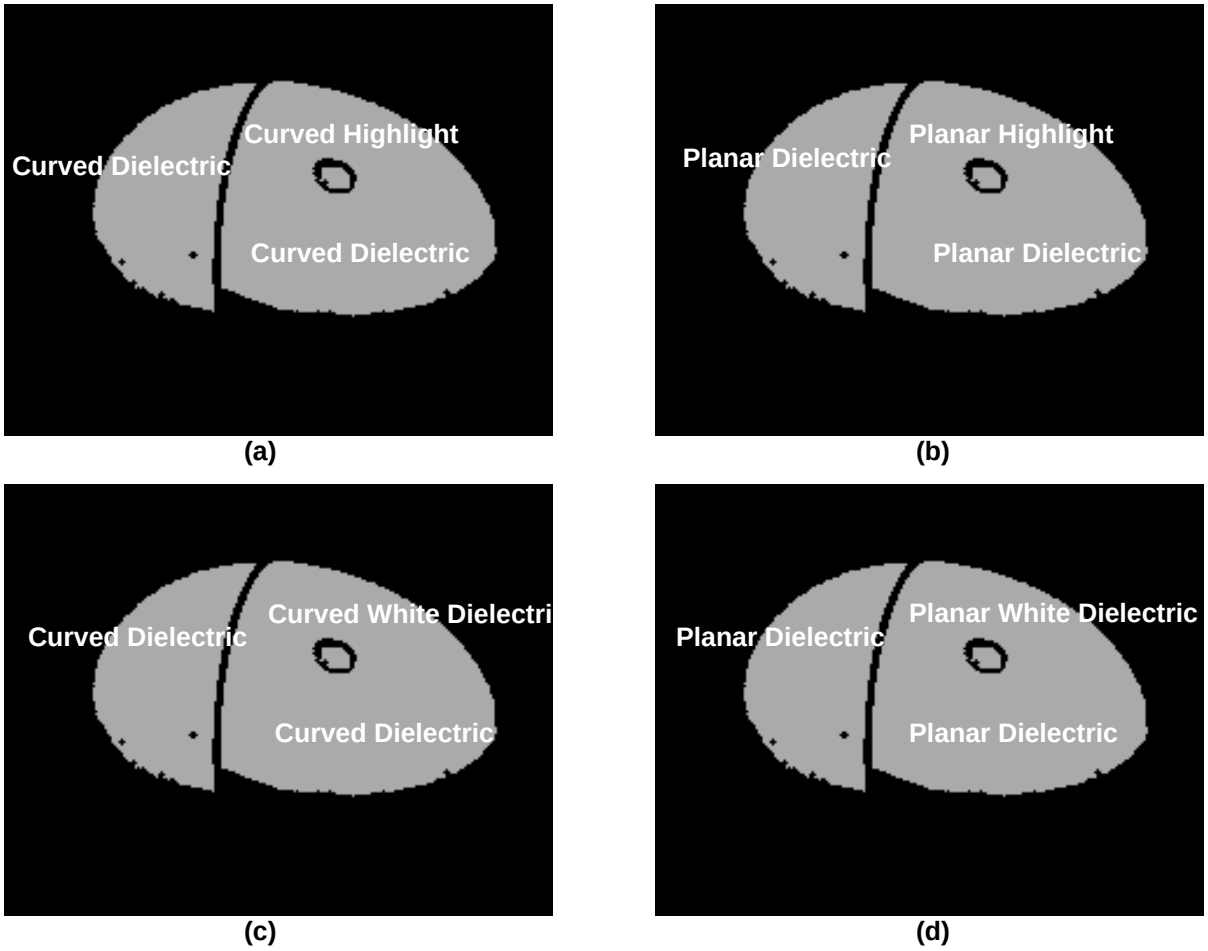


Figure 8.4: Final set of segmentations of the egg image in order of likelihood: (a) curved two-colored object with a specularity, (b) planar two-colored object with a specularity, (c) curved three-colored object, (d) planar three-colored object.

a four-tiered system where not-merge edges are penalized according to how the hypotheses fit their regions.

Each not-merge edge begins with a penalty of 0. Then the algorithm uses the planarity test described in Chapter 7 to determine whether the region matches the proposed hypothesis shape. For each mismatch between a hypothesis and its region the edge receives one penalty point. In other words, if one hypothesis' shape matches its region and one does not, the edge receives a single penalty point. If neither hypothesis' shape matches its region, the edge receives two penalty points.

In addition to these two tests, the algorithm arbitrarily gives a penalty point to any not-merge edge connected to a specular hypothesis. This is because when two different objects overlap, the body reflection hypotheses are more likely to be adjacent because of the limited extent of a highlight. In particular, this handles the case where a bright white object is next to a darker colored object. A prime example of this is the handle on the stop-sign in the stop-sign and cup image, which looks like a highlight in color space.

After this string of tests, each not-merge edge has a certain number of penalty points. A not-merge edge with no penalty points receives a value of 0.5; a not-merge edge with one penalty point gets a value of 0.4925; a not-merge edge with two penalty points gets a value of 0.49; and all other not-merge edges get a value of 0.4875. This arranges the not-merge edges into four tiers according to how well the hypotheses match their respective regions.

Other than these changes, the system described in Chapter 3 and extended in Chapter 6 remains the same.

8.4. Analysis of results

Figure 8.4 shows the final set of segmentations on the egg image. In order of likelihood, the segmentations of the egg image are: the egg as a single curved two-colored object with a specularity, the egg as a single planar two-colored object with a specularity, the egg as a single curved three-colored object, and the egg as a single planar three-colored object.

Figure 8.5 and Figure 8.6 show the final set of segmentations of the stop-sign and cup image. Note that in this case the white regions all receive a specular hypothesis in their initial list. Because the white regions and the surrounding red sign do not fit the criteria of the specularity test, the most likely segmentation returned by the system is the same as before: the stop-sign, cup, and handle are three separate surfaces, and the stop-sign is planar and the cup and handle are curved. Only in the less likely segmentations do the specular hypotheses appear, and then they appear as separate surfaces since the compatibility test returns a low likelihood for merging them with the red sign.

The biggest problem with extending the system to handle specularity is that the initial segmentation algorithm does not always find the highlight regions. A secondary problem occurs when the highlight region has completely saturated the camera, leaving no pixels to provide an estimate of the surface reflection vector. In order for the specular extension to be robust, these are problems that need to be addressed in future work.

What the experiments in this chapter show is that by making simple extensions and integrating new hypotheses we can expand the basic system to deal with more complex images. These images are more complex than those dealt with by previous physics-based algorithms because they include multi-colored objects as well as dielectrics displaying both surface and body reflection.

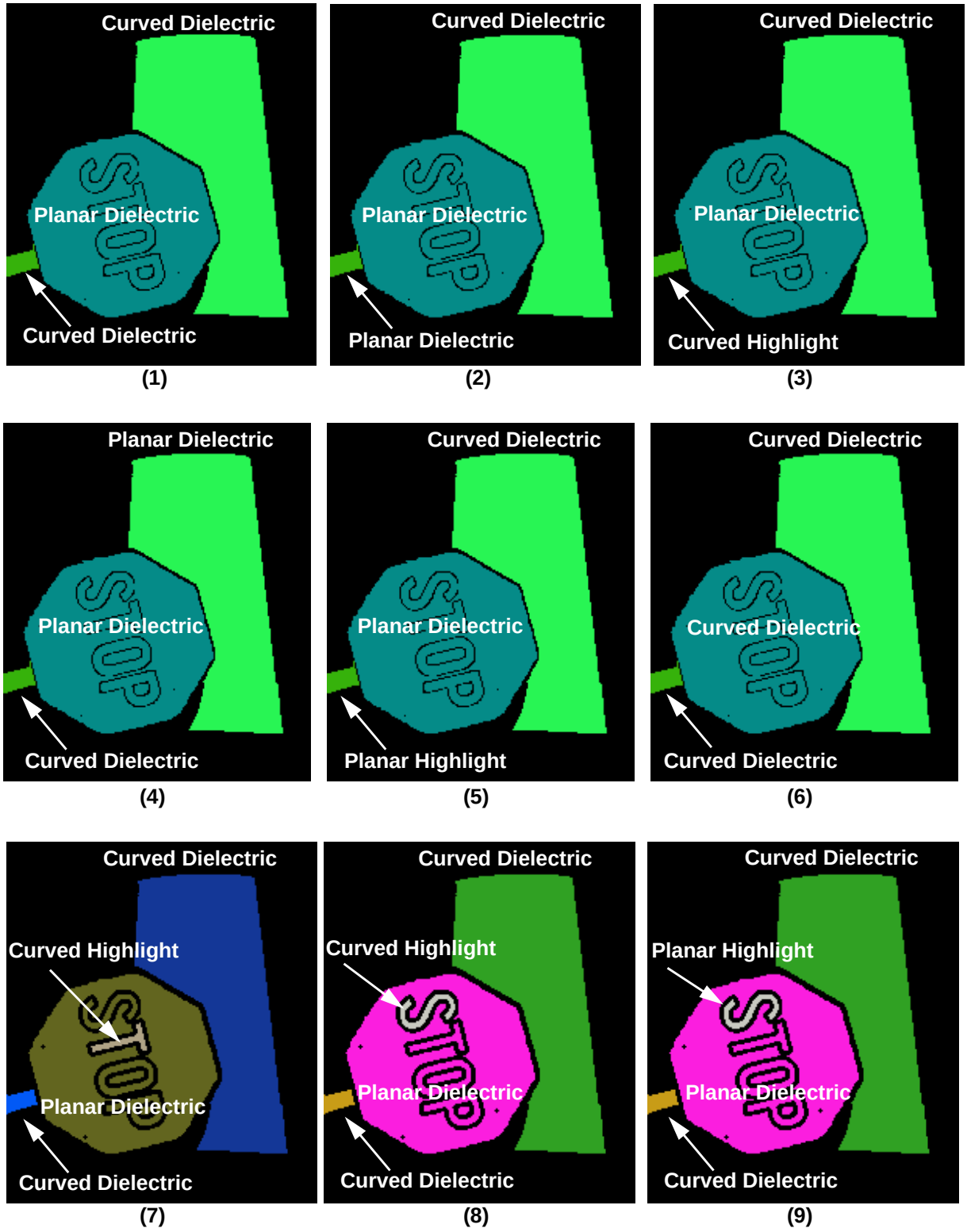


Figure 8.5: Top nine segmentations of the stop-sign and cup image. All of the hypotheses specify White Uniform Illumination. The text indicates the material and shape of each single or aggregate hypothesis in each segmentation.

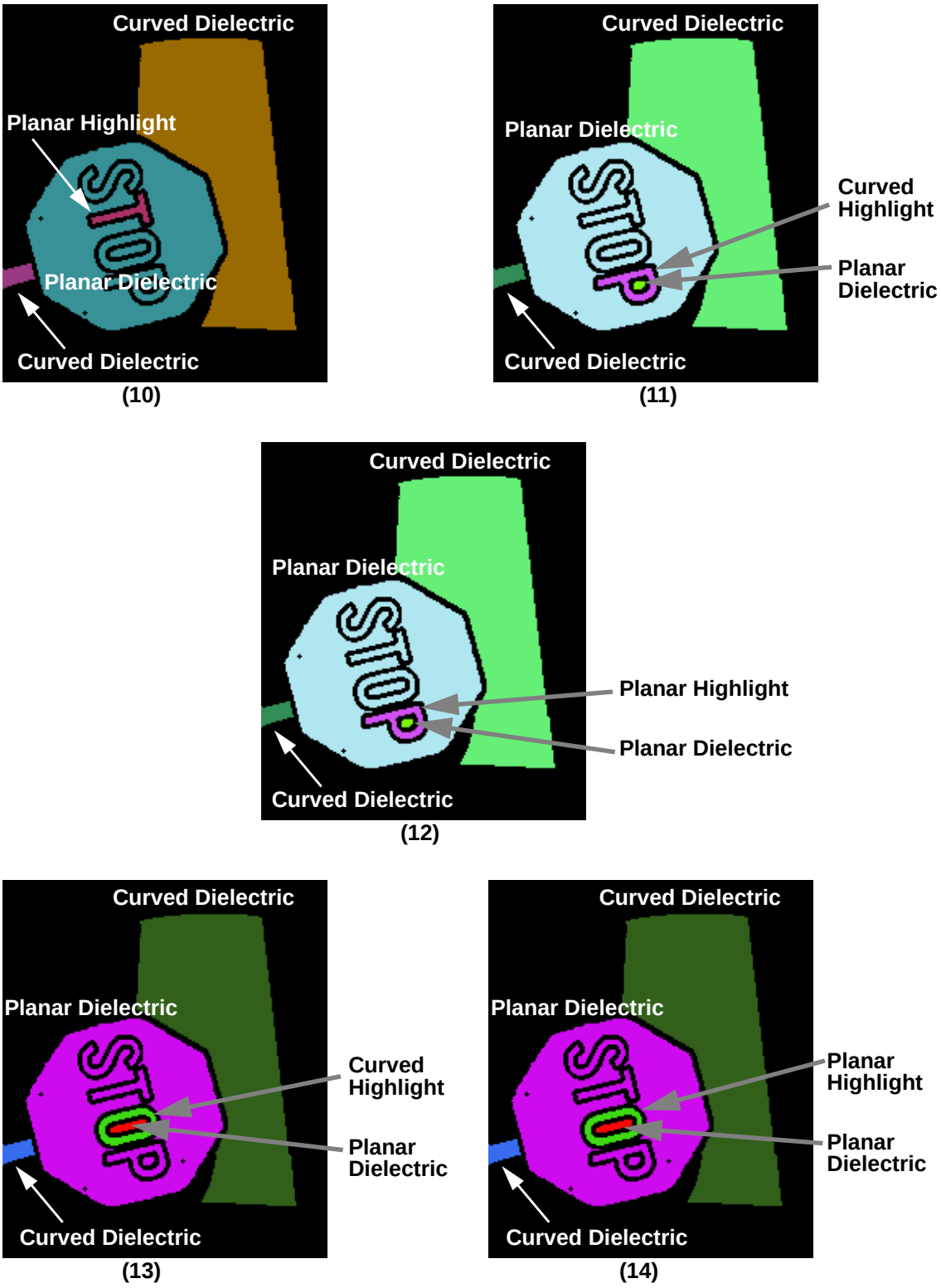


Figure 8.6: Final five segmentations of the stop-sign and cup image.

Chapter 9: Contributions and future work

9.1. Contributions & conclusions

This thesis contains two major contributions to the field of computer vision. First, it presents a theoretical framework for physics-based image segmentation. Second, it presents a segmentation system based upon this framework that moves beyond previous physics-based segmentation techniques both in the type of objects it can handle and the segmentation descriptions it provides. The implementation also demonstrates the generality of the theoretical framework.

The theoretical framework in itself is important for a number of reasons. First, it presents a set of complete parametric models for the individual scene elements: shape, illumination, and the transfer function. The parametric models allow us to reason about the space of possible explanations for a given image. Enabling this type of reasoning is essential in computer vision because of the many-to-one nature of the mapping from scenes to images.

Second, it develops a set of taxonomies of the scene elements based upon the parametric models. These taxonomies again help us to think about the space of explanations for an image and begin to enumerate them using broad classes. The process of enumeration led us to the set of fundamental hypotheses, which are a specific enumeration of the explanations for a given image region. The taxonomies also have two potential future uses. First, they form a basis for classifying algorithms and scenes according to their complexity. Second, they should be a good theoretical basis for a standard data base of color imagery.

Finally, the theoretical framework gives us a methodology for comparing the hypotheses of adjacent regions. The most important aspect of this methodology is that it reduces the complexity of the problem by limiting the number of hypothesis pairs that need to be compared for similarity. Without this reduction of complexity, segmentation using multiple hypotheses quickly becomes intractable.

The segmentation system based on the framework also makes a number of contributions. First and foremost it moves beyond current color image segmentation approaches by intelligently handling scenes containing multi-colored objects. In order to accomplish this, the system has to explore higher-level features of image regions to search for compatibility.

To facilitate this exploration we developed three general tools for measuring the compatibility of inhomogeneous dielectric objects under uniform illumination, which includes point and area light sources. These compatibility tests, based on the reflectance ratio, gradient direction, and intensity profiles, proved to be accurate indicators of whether two image regions could possibly belong to the same surface. Independent of the system, these compatibility tests should have application in other vision tasks where we want to find coherent surfaces.

The second major contribution of the implementation is that it shows the strengths and expand-

ability of the theoretical segmentation framework. In particular it is an example of a general segmentation system and its component parts. Chapter 8 showed how we can expand this general system to include new hypotheses and new compatibility tests. Future expansion is straightforward given new compatibility tests for different hypothesis pairs.

The third major contribution is a new type of segmentation output. The framework and system return a set of rank-ordered segmentations that indicate the most likely region groupings and region interpretations. The set of rank-ordered segmentations provides more information than a single, potentially incorrect segmentation, and the region interpretations provide more high-level information about the scene than previous segmentation methods.

Finally with the theoretical framework and the system we have added yet another layer to vision. Segmentation, almost by definition is a low-level process. But this thesis has shown that we can divide this low level into two layers. The first layer is the task of finding pixels in an image that can reasonably be assumed to belong to a single surface. The second layer takes these initial regions and searches a different, higher-level feature space for compatibility between the regions.

By splitting segmentation into these two tasks, each one becomes easier. To accomplish the initial task we can focus upon image processing techniques including color-based region growing, texture segmentation, and grayscale segmentation techniques. For the second task we are able to work with regions, which contain more information than pixels. Therefore, we can think about this information at a higher-level in terms of the shape, illumination, and material properties of the objects in the scene.

This two-level merging process has the potential to allow us to handle more complexity, more intelligently than using only image processing techniques or using the higher-level features from the beginning. Previous physics-based techniques reflect this dichotomy somewhat. For example, Healey uses region splitting followed by a merging step using physical models [16]. Klinker divides an image into blocks, uses physical reasoning to group the blocks, and then uses the aggregate blocks to generate hypotheses for a second region growing step [25]. Neither, however, explicitly divides the process into two stages. This work shows that it is a potentially useful division of labor.

It is worth noting here all of the constants and thresholds used in the entire segmentation process. Table 9.1 describes them and their default values. It also shows whether they are allowed to change, or are held constant. Overall, Table 9.1 shows that 17 of the 21 thresholds and parameters are constant for all of the test images. Only four of the parameters associated with the initial segmentation algorithm are allowed to change. Thus, while the system has a large number of parameters, we can set most of them once, based on a set of test images. We only need to tune a small number of them for particular images.

In conclusion, the framework and system described herein help to further define and explore the fields of physics-based vision and physics-based segmentation. They make contributions on both the theoretical and practical level and move the state-of-the-art forward.

9.2. Where do we go from here?

Given the system's performance on the set of ten test images, we can claim the system works well on a restricted class of images. However, the test set does not really indicate the boundaries of the

Table 9.1: System Constants, Parameters, and Thresholds

Variable/Threshold	Default Value	Held Constant?
Dark Threshold (Initial Segmentation [IS])	13 pixel values (of 255)	Adaptable
Local Normalized Color Threshold (IS)	0.04 Euclidean Distance	Adaptable
Global Normalized Color Threshold (IS)	0.15 Euclidean Distance	Adaptable
Region Size Threshold (IS)	100 pixels	Adaptable
Seed Region Size	3x3 pixels	Constant
Adjacent Region Search Distance	8 pixels	Constant
Minimum Border Length	5 border pairs	Constant
White Normalized Color Threshold Radius	0.04 Euclidean Distance	Constant
Reflectance Ratio Threshold Variance	0.08	Constant
Gradient Direction Threshold Variance	10 buckets	Constant
Gradient Intensity Threshold	$2\ \frac{\text{pixel values}}{\text{unit pixel distance}}$	Constant
Gradient Direction Minimum Percentage	20%	Constant
Gradient Direction Parallel Threshold	148	Constant
Number of Gradient Direction Buckets	10	Constant
Maximum Polynomial Order (Profile Analysis)	5	Constant
Analysis Test Weights (GD, RR, PA)	(0.1, 0.25, 0.65)	Constant
Default Discontinuity Edge Value	0.5	Constant
Uniform Intensity Relative Threshold Variance	0.0029	Constant
Highlight Region Maximum Growth Distance	10 pixels	Constant
Highlight Region Minimum Pixel Value	30 pixel values	Constant
Optimal Highlight Intersection Distance	0.75	Constant

system's performance. Finding these boundaries is necessary in order to characterize the applicability of this work to current machine vision tasks. That said, however, it is important to note that this work is not a final product, but proof of concept for a new segmentation framework.

Perhaps more importantly, we need to explore the boundaries of the system's capabilities in order to form a plan for future research. To that end, we took an additional set of pictures, most of which bend or break the underlying assumptions in some manner, in order both to show the limitations and strengths of the system and to discover where future research can make the greatest impact.

9.2.1 Characterizing the limitations

Figure 9.1 shows the new test images, and Figure 9.2 shows their initial segmentations. The first thing we note is that the initial segmentation routine has problems with several of the images. The first problem is that, because the initial segmentation algorithm relies only on color, objects of similar color get merged together in the initial segmentation. This occurs in both the blocks image and the shirt image. In the blocks image, the donut gets merged with the red blocks, and in the shirt image, the two shirts lying on top of one another get grouped together. The issue this highlights is that any general-purpose initial segmentation algorithm needs to take into account intensity discontinuities as well as color discontinuities. These two characteristics need to be used together so that edge-based considerations do not over-segment the image and color-based considerations do not under-segment it.

The second major problem with the normalized color segmentation algorithm is that it cannot handle small-scale texture. As we see in the dave, dino, and shirt pictures, using bounded normalized color to grow the regions is insufficient for general images; the textured regions contain many holes, and the borders between them are ragged and far apart. To be truly general, we need to include texture as well as color and edge information.

The lesson to be learned from these images is that, even though this thesis did not focus on the initial segmentation algorithm, it is a significant limiting factor to better performance. Just as in other fields such as speech recognition, higher-level processes are only as good as their inputs. Thus, future research needs to look more closely at improving the initial segmentation algorithm.

These test images also give us an indication of how we need to expand the initial hypothesis list. For the most part, these images are dielectrics under white uniform illumination. However, two of the pictures also contain dielectrics under general colored illumination where there is interreflection between two objects. In the big cups image, for example, there are long thin regions between the cups that display strong interreflection. Likewise, in the stuff image the red donut and the pepsi cup affect one another's appearance. Expanding the initial hypotheses to include (Colored | White Dielectric, Colored General Illumination, Curved | Planar) is, therefore, necessary to correctly segment and interpret these images.

Moving on to the results of the analysis and merging stage, Figure 9.3 shows the final region groupings for these test images. From these results we identify three major problems motivating future research. First, we see that a couple of image regions of different objects that have smooth, reasonably long borders are grouped together. In particular, the system groups the two right-most cups in the big cups image, and it groups the pepsi cup and red donut in the stuff image.

In both of these cases there is a small region of interreflection between two large regions. The region of interreflection, unfortunately, is compatible with both of its neighboring regions according to the three tests. The major reason for this is that the small interreflection region does not contain enough shape information for the profile analysis test to return a reliable result. Because the algorithm does not *dynamically recalculate the edge weights* after merging regions, all three regions merge together in the final segmentation.

On the other hand, if the algorithm did recalculate the edge weights after, for example, merging the pepsi cup with the small interreflection region, then the aggregate region may not be compatible with the remaining region. In the case of the pepsi cup, if we recalculated the three tests after

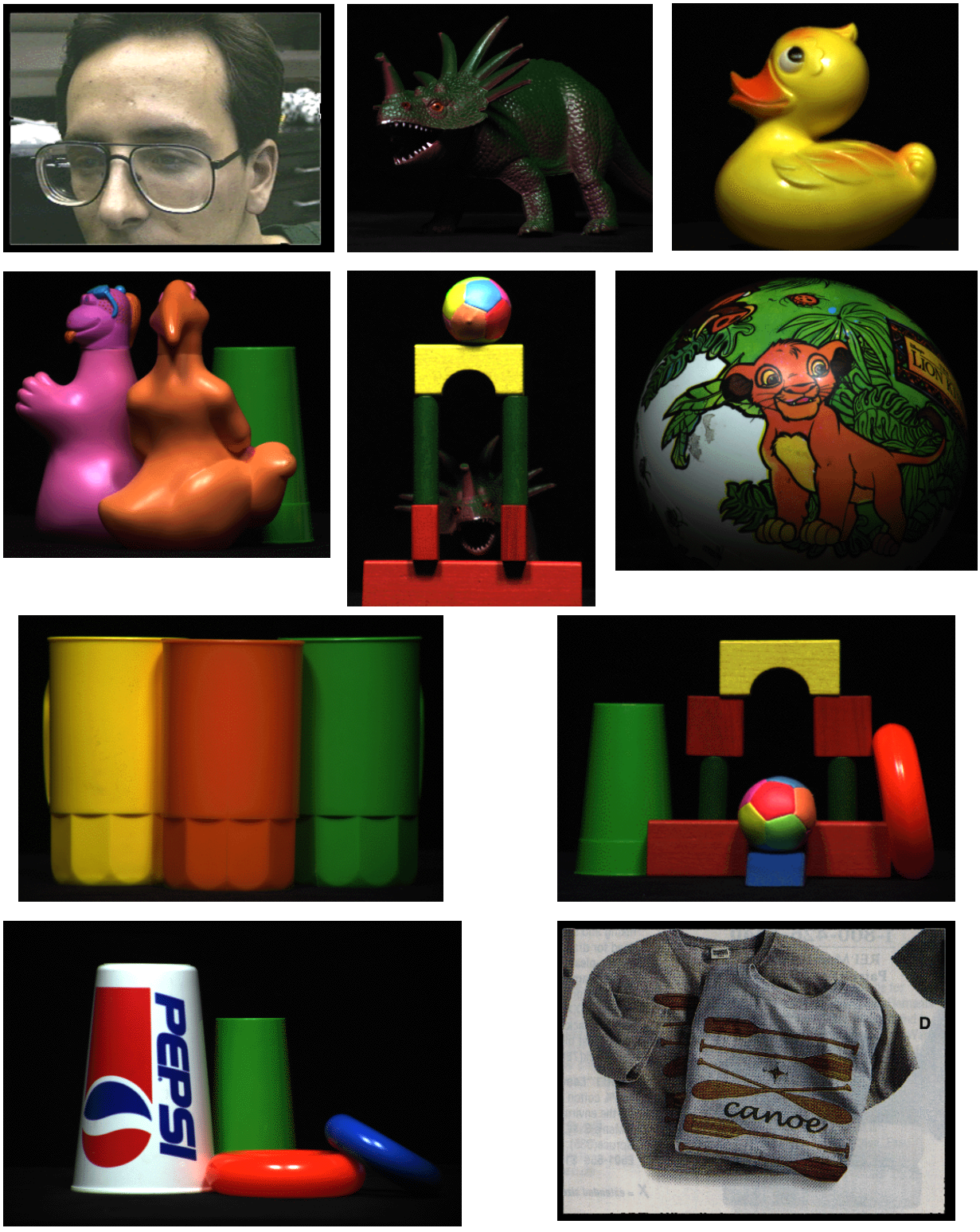


Figure 9.1: Extra test images. Top row: dave, dino, duck. Second row: kooky, tower, lion. Third row: big cups, blocks. Bottom row: stuff, shirt. All of these images, except possibly kooky, contain parts that break the system assumptions.

Figure 9.2: Initial segmentations of: dave, dino, duck, kooky, tower, lion, big cups, blocks, stuff, and shirt. Note the problems the initial segmentation algorithm has with textured surfaces, in particular.

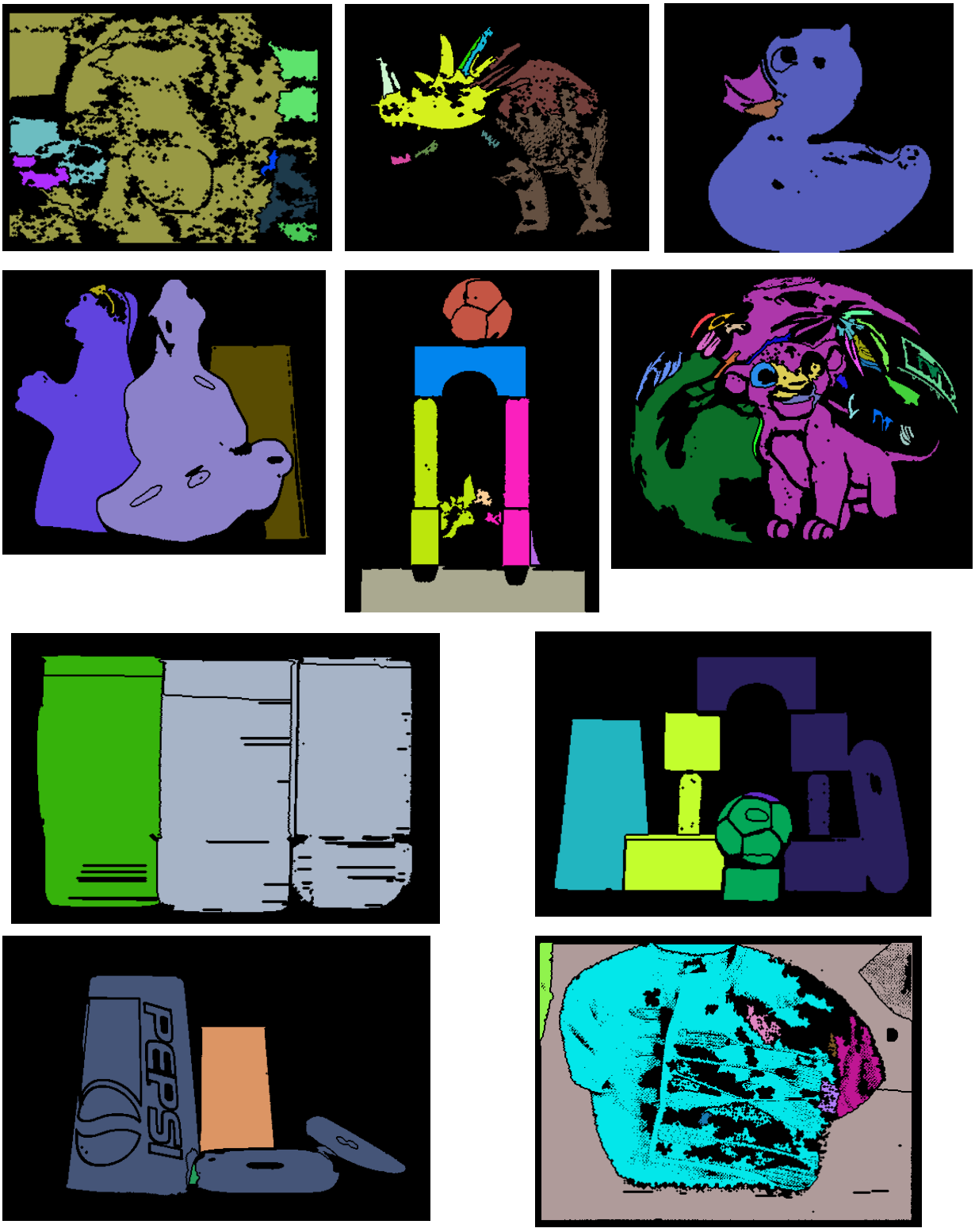


Figure 9.3: Final region groupings of: dave, dino, duck, kooky, tower, lion, big cups, blocks, stuff, and shirt.

merging the interreflection region with the pepsi cup, then the aggregate pepsi-interreflection region may not be compatible with the donut as the larger region's shapes are incompatible.

The second major problem occurs in images where regions of different objects with short, irregular borders get grouped together. For example, dave, tower, stuff, and blocks, all contain objects linked by short borders. The intensities along these borders do not contain significant variation in such a small area, implying that the reflectance ratio test will always return a positive compatibility result. Therefore, if the profile analysis returns a weak result, neither strongly positive nor negative, then the regions will have an edge value indicating compatibility.

A prime example of this scenario is the dave image. In the dave image, the upper left region gets merged with dave's head, even though they should be incompatible. Note, however, that the background is not close enough to dave's head for the algorithm to consider them adjacent. Only a small portion of a region of dave's hair is adjacent to the background region. In this case, the border between the hair and the background do not contain significant intensity variation, causing the reflectance ratio to return a strongly positive result, and the profile analysis returns an ambiguous result. While dynamically recalculating the weights might help this case because of the increased shape information for the profile analysis to work with, there is a basic problem with short borders; short borders often do not contain sufficient information with which to make a decision. Currently the algorithm requires borders to have at least 5 pixels in common. A useful direction of future research would be to design an effective *border length filter,* or somehow integrate border length information into the edge weights. This feature, in particular, would help keep the system from under-segmenting the images.

The third major problem in these images is *specular candidate identification*. This problem is twofold. First, the initial segmentation algorithm often misses specular regions, such as on the red donut in the stuff image. Normalized color is not a reliable tool for finding highlights unless they come close to saturating the camera. Second, not all highlights are white. A highlight on a rough surface such as the ball in the blocks image, for example, does not saturate the camera and retains a significant amount of body reflection color. In the interest of efficiency, we did not assign a specular hypothesis to all colored regions. However, it may be possible to broadly identify candidate highlight regions when assigning the initial hypotheses, whether or not they are white or colored. Future research in finding and identifying the specular highlight regions would help the algorithm deal with images such as kooky, blocks, lion, and stuff.

It is important to note that the algorithm did not do poorly on all of the images. Despite the fact that all of the images except kooky break some assumption of the system or contain a hypothesis not yet implemented, the system does a reasonable job. For example, in the dave image the algorithm groups together the different regions of dave's face and for the most part separates it from the background. In the dino image, the system separates the face and body, and groups together most of the body regions. In the duck image, the system groups together the body parts and separates the beak, which has both varying intensity and significantly different shape than the body. The kooky image is an excellent result. Likewise, the tower image is an excellent result except for merging portions of the dinosaur's head with the blocks because of the very short borders. In the lion image the system groups most of the regions together despite the fact they are not piecewise uniform. In the cup image, the small region of interreflection between the right two cups is the only problem. In the blocks image, the short border between the ball and ramp, and the joining of the red blocks and red donut are the major problems. Likewise, in the stuff image short borders

again confound the compatibility tests. Finally, in the shirt image the system merges the designs on the shirt with the body of the shirt and yet keeps them separate from the background. Overall, therefore, we feel the system actually performs reasonably well on these images except for the problems identified previously.

9.2.2 Future work

As identified above, there are two major directions for future research based on this segmentation framework and system. The first is improving the initial segmentation algorithm. Currently, the initial segmentation algorithm limits the system to images containing piecewise uniform objects, and does not take into account edges which may divide objects with similar colors. Furthermore, if we want to handle texture or grayscale images, we have to develop alternative initial segmentation algorithms. Low level algorithms exist for these types of images; we need to find methods that fit the needs of the system.

The second direction involves making improvements to the system and expanding its capabilities to allow it to handle both more complex objects and more complex scenes. The tasks ahead include implementing:

- dynamic recalculation of the edge weights,
- a border length filter, and
- better specular candidate identification.

Incorporating these capabilities into the system will give it greater robustness and coverage than its current instantiation.

It is also important to begin applying this framework and system to mainstream vision tasks. Given that the system attempts to find coherent surfaces in an image, there is a natural fit between this segmentation system and object recognition tasks. In particular, there is the potential for application to video data-base library tasks. The utility of a general segmentation algorithm is twofold. First, as an interactive tool for segmenting and identifying objects when building image data-bases and labeling images, and second, as an automatic tool for looking at unlabeled pictures and identifying object characteristics.

To begin to apply the system to such tasks, however, the system not only needs to handle more complex imagery, but also be more robust to noise, small regions, and other aspects of everyday scenes. The extent of the improvements will depend upon the task and the required output of the system. What this thesis has provided, however, is the framework for a general-purpose segmentation system and the direction we must move in order to build it.

Bibliography

[1] E. Adelson and J. Bergen, “The Plenoptic Function and the Elements of Early Vision,” in *Computational Models of Visual Processing* (M. S. Landy, and J. A. Movshon Ed.), MIT Press, Cambridge, 1991.

[2] R. Bajcsy, S. W. Lee, and A. Leonardis, “Color image segmentation with detection of highlights and local illumination induced by inter-reflection,” in *Proc. International Conference on Pattern Recognition*, Atlantic City, NJ, 1990, pp. 785-790.

[3] P. Beckmann, and A. Spizzochino, *The Scattering of Electromagnetic Waves from Rough Surfaces*, Artech House, Norwood, 1987.

[4] M. Bichsel and A. P. Pentland, “A Simple Algorithm for Shape from Shading,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1992, pp. 459-465.

[5] P. Breton, L. A. Iverson, M. S. Langer, S. W. Zucker, ”Shading flows and scenel bundles: A new approach to shape from shading,” in *Computer Vision - European Conference on Computer Vision*, May 1992, pp. 135-150.

[6] C. R. Brice and C. L. Fenema, “Scene analysis using regions,” *Artificial Intelligence* 1, 1970, pp. 205-226.

[7] M. H. Brill, “Image Segmentation by Object Color: A Unifying Framework and Connection to Color Constancy,” *Journal of the Optical society of America* A 7(10), 1990, pp. 2041-2047.

[8] M. Born and E. Wolf, *Principles of Optics*, Pergamon Press, London, 1965.

[9] M. J. Brooks and B. K. P. Horn, “Shape and Source from Shading,” in *Proceedings, Int’l Joint Conf. on Artificial Intelligence*, August 1985, pp. 932-936.

[10] M. Cohen and D. Greenberg, “The hemi-cube: a radiosity solution for complex environments,” *Computer Graphics Proc. of SIGGRAPH-85*, 1985, pp. 31-40.

[11] T. Darrell, S. Sclaroff, and A. Pentland, “Segmentation by Minimal Description,” in *Proceedings of International Conference on Computer Vision*, IEEE, 1990, pp. 112-116.

[12] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, New York, John Wiley & Sons, 1973.

[13] J. D. Foley, A. van Dam, S. K. Feiner, J. F. Hughes, *Computer Graphics: Principles and Practice, 2nd edition*, Addison Wesley, Reading, MA, 1990.

[14] D. Forsyth, and A. Zisserman, “Reflections on Shading,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 13, no. 7, July 1991, pp. 671-9.

[15] X. D. He, K. E. Torrance, F. X. Sillion, and D. P. Greenberg, “A Comprehensive Physical Model for Light Reflection,” *Computer Graphics*, vol. 25, no. 4, July 1991, pp. 175-186.

[16] G. Healey, "Using color for geometry-insensitive segmentation," *Journal of the Optical Society of America* A 6(6), June 1989, pp. 920-937.

[17] B. K. P. Horn, *Robot Vision*, MIT Press, Cambridge, 1986.

[18] B. K. P. Horn, "Understanding Image Intensities," *Artificial Intelligence*, 8(11), 1977, pp.201-231.

[19] B. K. P. Horn, *Shape from Shading: A Method for Obtaining the Shape of a Smooth Opaque Object from One View*, Ph.D. thesis, MIT, 1970.

[20] R. S. Hunter, *The Measurement of Appearance*, John Wiley and Sons, New York, 1975.

[21] K. Ikeuchi and B. K. P. Horn, "Numerical shape from shading and occluding boundaries," *Artificial Intelligence*, 17(1-3), 1981, pp.141-184.

[22] D. B. Judd and G. Wyszecki, *Color in Business, Science, and Industry*, 3rd ed., John Wiley and Sons, New York, 1975.

[23] J. Kaufman and J. Christensen, *IES Lighting Ready Reference*, Illuminating Engineering Society of North America, 1985.

[24] G. J. Klinker, S. A. Shafer and T. Kanade, "A Physical approach to color image understanding," *International Journal of Computer Vision*, 4(1), 1990, pp. 7-38.

[25] G. J. Klinker, "A Physical Approach to Color Image Understanding," Ph.D. Thesis, Carnegie Mellon University, CMU-CS-88-161, May 1988.

[26] T. Y. Kong and A. Rosenfeld, *Topological Algorithms for Digital Image Processing*, North-Holland, to appear in 1996.

[27] J. Krumm, *Space Frequency Shape Inference and Segmentation of 3D Surfaces*, Ph.D. Thesis, CMU-RI-TR-93-32, Carnegie Mellon University, December 1993.

[28] M. S. Langer and S. W. Zucker, "A ray-based computational model of light sources and illumination," in *IEEE Workshop on Physics-Based Modelling in Computer Vision*, Cambridge, MA, June 1995, pp. 93-99.

[29] L. Lapin, *Probability and Statistics for Modern Engineering*, PWS Engineering, Boston, 1983.

[30] S. M. LaValle, S. A. Hutchinson, "A Bayesian Segmentation Methodology for Parametric Image Models," Technical Report UIUC-BI-AI-RCV-93-06, University of Illinois at Urbana-Champaign Robotics/Computer Vision Series.

[31] Y. G. Leclerc, "Constructing Simple Stable Descriptions for Image Partitioning," *International Journal of Computer Vision*, 3, 1989, pp. 73-102.

[32] C. H. Lee and A. Rosenfeld, "Improved methods of estimating shape from shading using the light source coordinate system," in B. K. P. Horn and M. J. Brooks, Eds., *Shape from Shading*, MIT Press, Cambridge, 1989.

[33] H.-C. Lee, "Method for Computing the Scene-Illuminant Chromaticity from Specular Highlights," *Journal of the Optical Society of America* A 3(10), 1986, pp. 1694-1699.

[34] H.-C. Lee, E. J. Breneman, and C. P. Schulte, "Modeling light reflection for color computer vision," *IEEE Trans. on Pattern Analysis and Machine Intelligence* PAMI-12(4), April 1990, pp. 402-409.

[35] A. Leonardis, *Image Analysis Using Parametric Models: Model-Recovery and Model-Selection Paradigm*, Ph.D. Thesis, LRV-93-3, University of Ljubljana, March 1993.

[36] A. Leonardis, A. Gupta, and R. Bajcsy, "Segmentation as the Search for the Best Description of the Image in Terms of Primitives," in *Proceedings of International Conference on Computer Vision*, IEEE, 1990, pp. 121-125.

[37] B. A. Maxwell and S. A. Shafer, "A Framework for Segmentation Using Physical Models of Image Formation," in *Proceedings of Conference on Computer Vision and Pattern Recognition*, IEEE, 1994, pp. 361-368.

[38] B. A. Maxwell and S. A. Shafer, "Physics-Based Segmentation: Moving Beyond Color," in Proceedings of Conference on Computer Vision and Pattern Recognition, IEEE, 1996.

[39] P. H. Moon and D. E. Spencer, *The Photic Field*, MIT Press, Cambridge, 1981.

[40] S. K. Nayar, K. Ikeuchi, and T. Kanade, *Surface Reflection: Physical and Geometrical Perspectives*, CMU-RI-TR-89-7, Robotics Institute, Carnegie Mellon University, 1989.

[41] S. K. Nayar and R. M. Bolle, "Reflectance Based Object Recognition," to appear in the *International Journal of Computer Vision*, 1995.

[42] F.E. Nicodemus, J. C. Richmond, J. J. Hsia, I. W. Ginsberg, and T. Limperis, *Geometrical Considerations and Nomenclature for Reflectance*, National Bureau of Standards NBS Monograph 160, Oct. 1977.

[43] C. Novak, "Estimating Scene Properties by Analyzing Color Histograms with Physics-Based Models," Ph.D. Thesis, School of Computer Science, Carnegie Mellon University, 1992.

[44] R. B. Ohlander, "Analysis of Natural Scenes," Ph.D. dissertation, Dept. of Computer Science, Carnegie Mellon University, April, 1975.

[45] Y. Ohta, T. Kanade, and T. Sakai, "Color Information for Region Segmentation," Computer Graphics and Image Processing, 13, 222-241, 1980, pp.222-241.

[46] D. Panjwani and G. Healey, "Results Using Random Field Models for the Segmentation of Color Images of Natural Scenes," in Proceedings of International Conference on Computer Vision, June 1995, pp. 714-719.

[47] A. P. Pentland, "Shape information from shading: a theory about human perception," in *Proceedings of Int'l Conference on Computer Vision*, 1988, pp.404-413.

[48] A. P. Pentland, "Local shading analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6, 1984, pp.170-187.

[49] A. P. Pentland, "Finding the Illuminant Direction," *Journal of the Optical Society of America*, Vol. 72, No. 4, pp. 448-455, April 1982.

[50] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, *Numerical Recipes in C: the*

Art of Scientific Computing, 2nd Edition, Cambridge University Press, Cambridge, 1992.

[51] J. Rissanen, *Stochastic Complexity in Statistical Inquiry*, World Scientific Publishing Co. Pte. Ltd., Singapore, 1989.

[52] J. M. Rubin and W. A. Richards, "Color Vision and Image Intensities: When are Changes Material?" *Biological Cybernetics* 45:215-226, 1982.

[53] S. A. Shafer, "Using Color to Separate Reflection Components," COLOR research and application, 10, 1985, pp. 210-218.

[54] R. Stone and S. Shafer, "The Determination of Surface Roughness from Reflected Step Edges," submitted to JOSA, 1993.

[55] J. M. Tenenbaum, M. A. Fischler, and H. G. Barrow, "Scene Modeling: A Structural Basis for Image Description," in *Image Modeling*, (Azriel Rosenfeld Ed.), Academic Press, New York, 1981.

[56] S. Tominaga and B. A. Wandell, "Standard surface-reflectance model and illuminant estimation," *Journal of the Optical Society of America* A 6(4), pp. 576-584, April 1989.

[57] K. Torrance and E. Sparrow, "Theory for Off-Specular Reflection from Roughened Surfaces," in *Journal of the Optical society of America*, 57, 1967, pp. 1105-1114.

[58] R. S. Wallace, "Finding Natural Clusters through Entropy Minimization," Ph.D. dissertation, School of Computer Science, Carnegie Mellon University, CMU-CS-89-183, June 1989.

[59] R. G. Willson, "Modeling and Calibration of Automated Zoom Lenses," Ph.D. dissertation, Dept. of Electrical & Computer Engineering, Carnegie Mellon University, January 1994.

[60] L. B. Wolff, *A Diffuse Reflectance Model for Dielectric Surfaces*, The Johns Hopkins University, Computer Science TR 92-04, April 1992.

[61] Y. Yakimovsky and J. Feldman, "A semantics-based decision theory region analyzer," in *Proceedings 3rd International Joint Conference on Artificial Intelligence*, 1973, pp. 580-588.

[62] R. Zhang, P. S. Tsai, J. E. Cryer, M. Shah, "Analysis of Shape from Shading Techniques," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, June 1994, pp. 377-384.

[63] Q. Zheng and R. Chellappa, "Estimation of Illuminant Direction, Albedo, and Shape form Shading," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 7, July 1991, pp. 680-702.