LEARNING TO SEGMENT IMAGES INTO MATERIAL AND OBJECT CLASSES

Kenton Guadron McHenry, Ph.D.
Department of Computer Science
University of Illinois at Urbana-Champagin, 2008
Jean Ponce, Adviser

This dissertation addresses the task of learning to segment images into meaningful material and object categories. With regards to materials we consider the difficult task of segmenting objects made of transparent materials such as glass. To do this we consider information in the form binary features. Unlike more traditional unary features which consider information contained within a single location, binary features which consider information between pairs of locations are used to capture the notion of transparency (i.e. being able to see through something). We begin by using this in an edge based approach to locate the edges of glass objects. Segmenting transparent regions, which is desirable in order to locate objects, is ambiguous with this binary information alone. We deal with this by treating this information as a measure of discrepancy, relating how different two regions are from one another. We then combine this with a complimentary affinity measure which relates how well two regions belong together. These two measures are then combined within a single energy function which can be optimized to segment regions of transparent material. With regards to opaque objects an initial segmentation can be constructed using local features within regions produced from an over segmentation of the image. Our interest here is in improving these local segmentations by incorporating global information. Using global features (i.e. features that consider all regions simultaneously) and synthetically generated contrastive data an energy based model is constructed to estimate the quality of a given segmentation. Based on these segmentation quality estimates we attempt to improve a given segmentation.

LEARNING TO SEGMENT IMAGES INTO MATERIAL AND OBJECT CLASSES

BY

KENTON GUADRON MCHENRY

B.S., California State University of San Bernardino, 2001

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2008

Urbana, Illinois

Doctoral Committee:

Professor Jean Ponce, Chair
Professor David Forsyth
Professor Narendra Ahuja
Assistant Professor Eyal Amir
Professor Cordelia Schmid, INRIA

Abstract

This dissertation addresses the task of learning to segment images into meaningful material and object categories. With regards to materials we consider the difficult task of segmenting objects made of transparent materials such as glass. To do this we consider information in the form binary features. Unlike more traditional unary features which consider information contained within a single location, binary features which consider information between pairs of locations are used to capture the notion of transparency (i.e. being able to see through something). We begin by using this in an edge based approach to locate the edges of glass objects. Segmenting transparent regions, which is desirable in order to locate objects, is ambiguous with this binary information alone. We deal with this by treating this information as a measure of discrepancy, relating how different two regions are from one another. We then combine this with a complimentary affinity measure which relates how well two regions belong together. These two measures are then combined within a single energy function which can be optimized to segment regions of transparent material. With regards to opaque objects an initial segmentation can be constructed using local features within regions produced from an over segmentation of the image. Our interest here is in improving these local segmentations by incorporating global information. Using global features (i.e. features that consider all regions simultaneously) and synthetically generated contrastive data an energy based model is constructed to estimate the quality of a given segmentation. Based on these segmentation quality estimates we attempt to improve a given segmentation.

To Kenton Dale McHenry and Gloria Virginia McHenry

Acknowledgments

The research presented in this dissertation was partially supported by the National Science Foundation under grants IIS-0308087, IIS-0535152 and Toyota. Toyota also provided the data used for our scene segmentation method, which we are grateful for. I would also like to acknowledge the UIUC College of Engineering for supporting me with the SURGE fellowship and the UIUC Graduate College for supporting me with a Graduate College fellowship.

I would like to thank my adviser, Professor Jean Ponce, for his guidance during work on this thesis. I would also like to thank Professor David Forsyth for the various discussions we have had over the years. Lastly, I would like to thank Svetlana Lazebnik, Yasutaka Furukawa, Akash Kushal, James Davidson, and Sal Candido.

Table of Contents

Chapter 1 Introduction **1**
1.1 Segmentation Tasks 5
1.1.1 Segmenting Glass Objects 5
1.1.2 Segmenting Scenes 6
1.2 Contributions and Outline 7

Chapter 2 Background **9**
2.1 Transparent Materials 9
2.2 Opaque Materials 12
2.3 Non-Local Properties 14

Chapter 3 Glass Edges **17**
3.1 Edge Detectors and Glass 17
3.2 Edge Cues 19
3.2.1 Color Similarity 20
3.2.2 Blurring 21
3.2.3 Overlay Consistency 21
3.2.4 Texture Distortion 22
3.2.5 Highlights 23
3.3 A Local Classifier 25
3.4 Multiple Classifiers 28
3.4.1 Logical OR 29
3.4.2 Weighted Sum 29
3.4.3 Exponential Model 29
3.5 Global Integration 30
3.6 Experiments 30
3.7 Discussion 33

Chapter 4 Glass Regions **38**
4.1 Characterizing Glass Regions 39
4.1.1 Discrepancy 40
4.1.2 Affinity 41
4.2 Combining the Measures 45
4.3 Experiments 48
4.4 Discussion 49

Chapter 5 Scene Segmentation and Estimating Segmentation Quality . . . 53
5.1 Characterizing Objects within a Scene . . . 53
5.2 Local Model . . . 55
5.2.1 Features . . . 55
5.2.2 Probabilistic Model . . . 58
5.2.3 Segmentation . . . 59
5.3 Global Model . . . 62
5.3.1 Features . . . 62
5.3.2 Energy-Based Model . . . 63
5.3.3 Segmentation Improvement . . . 65
5.4 Experiments . . . 67
5.5 Discussion . . . 71
5.6 Figures . . . 73

Chapter 6 Conclusion . . . 80
6.1 Summary . . . 80
6.2 Future Work . . . 81

Appendix A Active Contours . . . 83
A.1 Snakes and Geodesic Active Contours . . . 83
A.2 Supervised Geodesic Active Regions . . . 87

Appendix B Energy Functions for Energy Based Models . . . 89

Appendix C Error Amplification . . . 91
C.1 Discrete Outcomes . . . 91
C.2 Continuous Outcomes . . . 93

References . . . 95

Author's Biography . . . 104

Chapter 1

Introduction

Segmentation is one of the most difficult tasks in computer vision. Semantically labeling portions of the image is what is needed for computers to eventually act upon the real world and the objects within it. However the task of assigning these labels is often ill defined and ambiguous. Lighting conditions, orientation, perspective, and color variations all distort what objects and scenes look like from one instance to another. Choosing the correct features to discriminate among the types of classes desired within the computational resource constraints at hand is essential. This dissertation considers the problem of supervised segmentation, specifically within scene segmentation and the largely overlooked area of material recognition/segmentation.

Unsupervised segmentation methods do not use prior information and instead attempt to group images into parts that appear to belong together [14, 17, 19, 58, 102, 118] or partition the image into pieces that appear to be different [48, 64, 93, 95, 104, 119, 120]. The result is a set of image regions without any semantic meaning attached. Supervised segmentation methods [8, 9, 21, 78, 86, 101, 114] involve learning the appearance of a predetermined set of classes (Figure 1.1). The result of such methods is an image with regions labeled as to belonging to a particular class, in essence using recognition of a particular class to segment the image. In other words, unlike unsupervised segmentation methods, the image is segmented into regions and labels that have some semantic value.

Essential to supervised segmentation methods are the features used to describe the

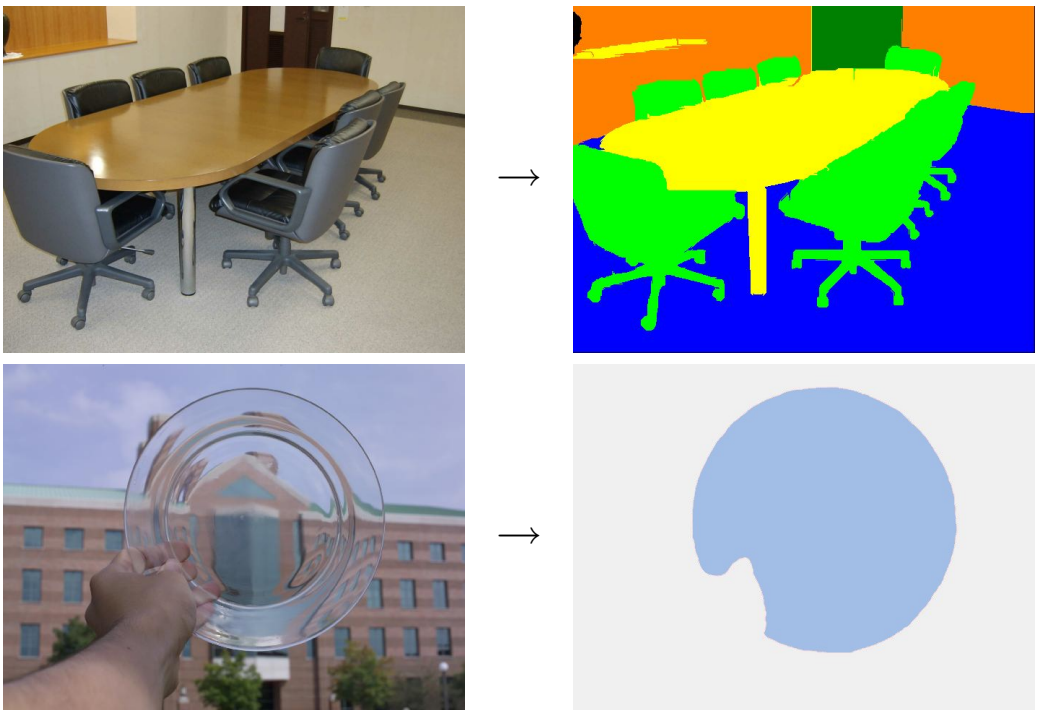

Figure 1.1: **Supervised Segmentation:** As opposed to unsupervised segmentation methods we segment images by classifying portions of the image as to being apart of a predetermined set of classes which our system has been trained to recognize via example images. Local visual properties such as color and texture can be used to discriminate between various classes. This thesis however is concerned about what can be done with less local properties. For example in the case of transparency, binary features (features that look at two regions at once), are used to locate glass objects.

classes. These features can involve recognizing objects such as faces and characters [114, 115], recognizing parts [78], and the shape of the object [8, 9]. However, usually the features of choice involve the material properties of objects such as color, reflectance [81] and texture [117]. Texture is often sufficient to distinguish different opaque materials thus most work in material classification has taken the approach of using texture classification to distinguish and identify materials [13, 34, 55, 59, 60]. Local texture however is but one of the available visual cues for material classification [30]. As argued by Adelson et al [1], there are many other cues for materials, many of which that can not sufficiently be described locally. In Adelson et al. various visual terms for describing materials are given. For example a mineralogist may use the words: luster (the optical quality of the surface), resinous (like

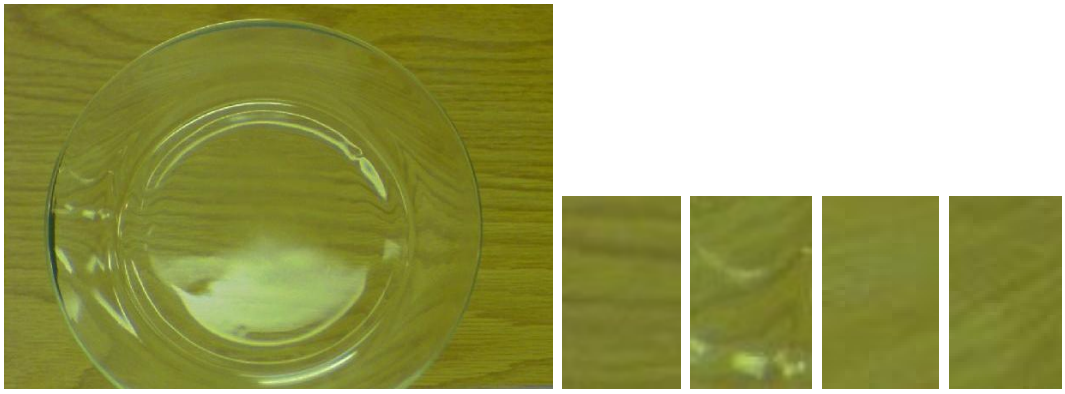

Figure 1.2: Transparency entails being able to see through something. Recognizing this involves having some idea what the background looks like (i.e. the image without the transparent object). Consider the image on the **left** of a glass plate in front of a uniform background. From **left** to **right** we have a small window from the background and three small windows taken from the plate. Looking at each individually it is difficult to say that you are looking at something transparent. However, taken together with the window that comes from the background we might suspect a transparent effect is present.

plastic), transparent, greasy, metallic, and dull. In addition minerals have habits, or typical forms. Example habits include: granular, fibrous, porous, scaly, nodular, columnar and platy. While some of the terms can be described as textures, such as dull, granular and scaly, others have properties that can not be described locally. For example, consider transparent materials. Transparent materials have no consistent local color or texture properties since their appearance always changes with the scene on the opposite side. A similar argument can be made for highly specular materials [27, 97, 98, 99, 100, 108].

We propose a less local approach to material classification for the purpose of supervised image segmentation. Rather than looking at one region at a time we consider relationships between multiple regions. In the case of transparent materials we look at pairs of regions and ask the question, "does region A look like a glass covered version of region B?" By doing this we are able to capture what it means to be transparent, something that we argue cannot be done by looking at one location by itself. Consider the image of a glass plate in front of a uniform background shown in Figure 1.2. To the right of the image four small windows are shown, the first taken from the background and the other three from the glass

Figure 1.3: Local information such as color and texture are not always enough to discriminate between differing classes. Consider the image on the **left**. The scene contains two classes, "desk" and "wall" which locally appear the same. The two windows on the **right** are taken from the two classes respectively. Based on color and texture the appear very similar.

plate. Observed individually it is difficult to say that we are looking at something that is transparent. However if each window from the glass plate is considered simultaneously with the window from the background one might notice that one is a blurred version of the other, similar but with specular highlights, or similar but with a distorted version of the texture. By recognizing the background and distortions to this background, effect that are associated with highly reflective and refractive glass objects, we attempt to recognize and segment transparent glass objects. In the case of opaque materials we look at all regions at the same time and ask the question "does this labeling look correct?" Consider the image of an indoor scene shown in Figure 1.3. Next to it are too small windows taken from the "desk" and the "wall" respectively. Though from different classes the color and texture of the two regions are very similar. However if we consider information beyond the local region, such as their location in the images, the classes nearby this region, the shape of the segments if this region was assigned this particular class we could potentially resolve much of the ambiguity. For example, most indoor scenes with desks have chairs nearby. This observation could be used in the above image to correctly classify the desk as "desk" as opposed to "wall". While locally two different materials can occasionally look similar there are often global properties

that can still distinguish them. It is these global constraints that we attempt to learn and use to improve the quality of our segmentations. We now describe the basic layout of the approach used.

1.1 Segmentation Tasks

Similar to the work of [39, 49] we utilize small local regions of the image that are assumed to have nearly uniform color and texture. Groups of these regions are then represented by a set of features. The number of regions considered at once and choice of features differs for transparent and opaque materials.

1.1.1 Segmenting Glass Objects

From the work of [5, 33, 79] we get an impression of the usefulness that knowledge of the background scene plays in dealing with transparent objects. The visual appearance of a transparent object always changes according to the scene behind it. Because of this there can be no completely local information that would be discriminative for transparent materials such as glass. However if we had some idea of what the background should be we could then create features that could capture the types of distortions we might expect from this background scene being seen through glass.

We argue that transparency involves both the perception of the background scene and the distortion to that scene caused by a transparent object covering a portion of it. Thus rather than looking for transparency within a single region we attempt to detect both the background and occluding transparent object simultaneously by looking at pairs of regions. Rather than looking at a single location and asking "is this glass?", we look at two locations and ask "is one location a glass covered version of the other?". In chapter 3 we describe the set of binary features used to identify the properties of a glass covered scene and how these are used to label the image. Initially we attempt to identify small sections of glass edges by

first using an edge detector then comparing small windows along the two sides of small edge snippets. A support vector classifier is trained to identify glass edges given several training images. In order to limit the number of false positives we restrict the output to only contain positively labeled glass edges that have the least ambiguity. The result is a segmentation consisting of a sparse number of positively labeled glass edges. To increase the number of positive edges we perform a hysteresis step between pairs of regions. If two edge pieces are classified as glass and have an edge path connecting them, then we classify the entire edge path as glass. Further, we assume only one glass object is present in the image. Under this assumption we enclose all positive glass edges by a single contour in order to identify a glass region.

In chapter 4 we extend our edge based method to a region based method. Starting with an initial over-segmentation of the image we compare pairs of regions. We use the support vector classifier trained previously as a discrepancy measure between regions indicating how much they appear to belong in different classes (i.e. glass and non-glass). We show how this information alone is ambiguous and introduce a complementary affinity measure based on edge continuity to indicate how much two regions belong together. Finally a geodesic active contour method [16] is used to minimize an energy function based on these complimentary measures. Unlike the edge based approach multiple glass objects are able to be segmented within an image.

1.1.2 Segmenting Scenes

In the case of opaque materials unary features are possible (i.e. we can ask "is this location wood?"). Similar to the work of [39] we use many features to describe the local material, shape and geometric properties. When our local model is constructed features will be selected based on their discriminative power for the given class. In chapter 5 we describe the features we use, why they were chosen, and how they are used to locally label the image.

We again consider information that can not be captured locally but could potentially

improve segmentation results. This time however we consider properties that are global in that they consider all regions at once as opposed to binary properties as before. Global information such as context [54, 89, 106], and shape [8, 22, 23, 94] have been used for this task. Here, rather than arbitrarily assigning some global constraint on the segmented image we attempt to learn one. In chapter 5 we introduce a number of global features which attempt to capture global information such as shape, context, compactness, and group location tendencies. An energy based model [57] is then trained via contrastive divergence learning [37] in order to prefer (assign lower energies to) segmentations that are more correct (i.e. with a greater percentage of regions labeled as they are in the ground truth). Training data for the contrastive divergence learning is generated synthetically from the ground truth labellings of a training set. The result is a set of positive examples with associated contrastive examples that are respectively worse in that a larger percentage of regions are mis-labeled. The resulting energy functions is finally used in a greedy manor to refine and impose learned global constraints on our initial segmentation based on purely local information.

1.2 Contributions and Outline

The main contributions of this dissertation can be summarized as follows:

- Our collection of binary cues which we use to look for transparent materials such as glass. Unlike many opaque materials it is difficult to identify transparency locally at one region. Instead two regions are considered at once in attempt to find covered versions of the background.

- A method that uses the above binary cues to label the edges of a transparent object within a single image.

- An extension to the above method that allows us to label regions as to being part of a transparent object or not. Unlike the edge based method which used the Snakes

of Kass et al. [48] as a last step to identify one region in the image outlined by the identified glass edges, the region based method is able to segment multiple separate glass objects within an image.

- Our collection of global cues which we use to apply global constraints to an image segmentation and identify better segmentations. Unlike most segmentation methods that incorporate non-local information we do not choose one particular cue such as context or shape but instead provide many cues designed to capture various global properties from a scene and allow our system to choose those among them that are useful.

- To our knowledge, our segmentation improvement method is unique in using an energy based model trained on synthetically generated contrastive data to evaluate the quality of segmentations. By damaging the ground truth segmentations for our training set we are able to generate many examples of segmentations that are relatively worse in quality. An EBM trained from this data is then used to compare two segmentations and identify one as an improvement over the other.

The rest of this dissertation is organized as follows. In chapter 2 we begin by describing the related work on transparency, material recognition and scene segmentation. In chapter 3 we argue that transparency can not be detected local to a single region, describe our set of binary cues for locating transparency, and use these to train a classifier capable of detecting the edges of glass. In chapter 4 we extend this to regions and segment the image into glass and non-glass objects. In chapter 5 we describe our set of local and global features along with the energy based model we will use to incorporate this non-local information and improve a segmentation constructed from purely local information. Finally, Chapter 6 summarizes the contributions of this work and considers possible future research directions. The work described in this dissertation has been previously published in [69, 70, 71].

Chapter 2

Background

In this chapter we review the relevant prior work involving transparent materials, opaque materials and segmentation methods that incorporate non-local properties to improve segmentation results.

2.1 Transparent Materials

Much of the work that has dealt with transparency has been concerned with layer separation [6, 25, 42, 43, 46, 105, 111] where motion is used separate a moving reflected background from a scene seen through a transparent sheet of glass. These methods have little to do with transparency since they focus on the separation of differing motions rather than the presence of a transparent material. An exception to this is the work of [61, 62] which uses statistics of natural images to separate a reflected layer using only one image. Specifically they have observed that the two overlays should have a minimal number of edges and corners. A cost function is constructed around this and used to find a pair of optimal layers. Beyond this most work on transparency has dealt with the 3D reconstruction of objects. In Hata et al [33] an array of lines is projected onto a drop of transparent paste. Given the index of refraction of that paste a genetic algorithm is used to evolve possible 3D surfaces to explain the distortion of the uniform lines in the image. In Murase et al [79] the goal is to reconstruct the surface of a wavy pool of water given the depth of the pool, the index of

refraction of water, and a background image underneath the water. Points are tracked on the surface as the water moves. The mean position of these points is taken to be the position of the point if the water were still (i.e. the points position without refractive distortion). Given this point and the offset point at each frame a surface normal can be calculated. Though the authors track points on the moving surface and use their mean position as the un-distorted position it is important to note that if the background were known, points could be matched to those of a single image of a wavy pool and the surface of the single image reconstructed. Ben-Ezra et al [5] again uses motion to reconstruct the shape of a transparent object, this time given the assumption that the background is far away and that light rays entering through the back of the object are mostly parallel. Points are tracked on the surface of the object as the camera moves around it. Objects, restricted to superquadrics, are then simulated and refined according to an energy function that desires reconstructions that comply with the tracked points using gradient descent. The dielectric properties of glass have also been used to reconstruct the surface of glass objects [75, 76, 77]. Miyazaki et al [75] use the specular properties of glass to reconstruct the surface using several images of the object taken with differently oriented polarizing filters. Un-polarized light becomes partially polarized when reflected from an objects surface. The surface normal at each pixel can be obtained given the intensity values observed as the polarizing filter varies from 0^0 to 180^0 (with some disambiguation since there are in general two possible normals).

Outside of 3D reconstruction there is the work of [3] which derives an optical flow equation to recover the refractive properties of a stationary object in front of a moving textured background. Once the distortions are recovered they can be applied to a new scene giving the effect that the transparent object is present. In Osadchy et al [82] they are concerned with recognizing transparent objects. Given a library of known object shapes and a light source direction an image of a transparent object can be registered to its corresponding shape within the library by observing its specularities. Basically the normals from a possible shape at the specular locations in a test image should occupy a small cluster on the Gaussian

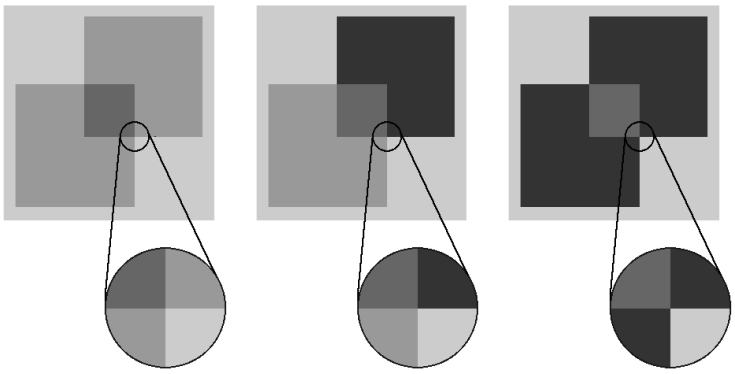

Figure 2.1: Adelson et al. [2] describes a set of constraints defined over junctions that allow for the perception of transparency. **Left:** A non-reversing junction in that the intensity consistently increases or decreases as we move across an edge. Either the vertical or horizontal edge can be caused by a transparent object. **Middle:** A single-reversing junction in that along one of the two edges the intensity either increases or decreases along one half and does the opposite in the other half. Transparency may only be perceived along the edge that does not reverse. **Right:** A double-reversing junction in that the intensity change reverses along both the vertical and horizontal edge. The perception of transparency is not present with this type of junction.

sphere.

Our goal here however is to segment materials. In the case of transparency this entails separating the image into transparent and non-transparent regions. Adelson and Anandan [2] introduce a linear model for the intensity of a transparent surface, namely

$$I = \alpha I_B + e \tag{2.1}$$

where I_B is the intensity of the background, α is a a blending factor, and e is the emission of the (semi) transparent surface itself. They use this model to derive a set of constraints on the brightness patterns of X-junctions at the boundary of transparent objects (Figure 2.1). From these a junction can be classified as one of three types: non-reversing if the intensity shift is in the same direction along both edges of the X, single reversing if it switches along one edge, and double reversing if it switches along both edges. Both non-reversing and single-reversing junctions allow for the possibility of a transparent edge being present, but double-reversing

ones do not. Singh [107] uses these constraints to separate transparent overlays from their background: Regions with similar color and texture are clustered together, and their edges are found. Each of the edge X-junctions are then labeled as either non-reversing, single-reversing, or double-reversing. The connected edges are then followed until returning to the junction, in effect outlining a region covered by a transparent overlay, or running into a double-reversing junction. If a double-reversing junction is reached, the algorithm backtracks to any single-reversing junctions that were seen along the way to try a different route. The reason for the clustering is to reduce the amount of data and increase the opportunity of successfully separating transparent regions. Notice that the constraints do not guarantee a transparent edge is present, but only the possibility. In fact it is very likely that an albedo pattern may satisfy these constraints. Thus taken locally the constraints only offer an indication of a possible transparent edge and must be then looked at globally to see if a region can be found that agrees with all the responses.

Unlike Singh et al, which identifies dark overlays (in order to have an emissive component) within synthetic and simple real world scenes, we are interested in recognizing and segmenting fairly clear transparent objects within complex scenes (such as glass plates and cups). Because glass is usually fairly clear the overlay constraints described above are not always usable thus more information is needed. In chapter 3 we describe the learning approach we use along with the visual cues we utilize to identify the presence of clear glass objects.

2.2 Opaque Materials

With regards to opaque materials visual cues such as color, texture [55, 84, 85, 90, 95], region similarity [91], and more recently geometry from visual properties [38, 39] can be used to segment images into meaningful categories. Randen et al. [90] reviews various methods that use filter bank responses to classify textures. These filtering approaches begin by convolving

an image with a set of filters, which respond to various spatial features within an image. The resulting responses can be concatenated to produce a feature vector for a particular material. By using a training set containing various example images of a specific material a classifier can be trained on these feature vectors to label this texture in novel images. In Varma et al [117] the goal is also to classify materials with invariance to illumination and viewpoint. To do this a number of textons, in this case clustered filter bank responses, are constructed from a training set from each class. Each training image is then represented by the histogram of their responses to these textons. A test image is classified by finding the training image with the smallest χ^2 distance between their histograms. The filter bank in this method is made robust to rotations by only considering the maximum response of each type of filter among rotations. The method is made robust to illumination changes by a pre-processing step which normalizes the intensity distribution within each image. In Lazebnik et al [56] affine invariant patches [73, 74] are used to classify textures. Repeatable interest points are first detected at various scales within an image then rectified to a canonical form that is invariant to affine transformations. These rectified regions are then represented by a descriptor [72], in this case spin images [45]. These descriptors are then clustered in order to reduce their number and leave only the most descriptive patches. Test images are classified by repeating this process and comparing the resulting representative patches to each of the training images via the Earth Movers Distance. By assuming the small image patches are locally planar and making them affine invariant Lazebnik et al's method is flexible to changes in viewpoint and deformations within non-rigid materials. They later go on in [55] to add neighborhood information, in the form of neighboring patch statistics within and in between classes. Using relaxation they are then able to segment a test image, possibly containing multiple textures, into its corresponding classes.

More recently, Hoiem et al [39] proposed starting with an over segmentation of the image into regions of uniform color and texture. Groups of these over-segmented regions, or super-pixels, are then represented by a large set of features describing local material,

shape, and geometric properties. A probabilistic classifier is then trained on these features so as to assign a higher confidence value to a given super-pixel when it is labeled with the correct class. Feature selection reduces the training space by only considering the most discriminative features. Given many initial segmentations, greedily generated from local similarities between super-pixels, they are able to choose the best one as the output. Unlike the method of Lazebnik et al. which attempts to directly account for changes in viewpoint, the level of viewpoint invariance is determined by the diversity of viewpoint found in the images of the training set. In addition the feature selection incorporated within this method allows one to throw a large number of features at a particular segmentation problem and allow the classifier to determine which ones are important.

2.3 Non-Local Properties

Local information can not capture relationships between various regions. An example would a smoothness constraint which preferred segmentations that had nearby regions labeled similarly. Non-local information is useful in clarifying ambiguous local information in the case when multiple labellings are equally adequate. It may also overturn local labellings if they are inconsistent with non-local observations. A few forms of non-local information that have been used are context [35, 54, 89, 106], symmetry [92, 109], and shape [8, 22, 23, 94].

In MRF frameworks the joint probability of an image and it's labels are modeled and used to assign an optimal labeling to novel images. The generative nature of MRF models however requires a large amount of data to estimate the parameters. In addition the locality used to make inference and parameter estimation practical makes the model inefficient at capturing long range interactions. Within CRF frameworks the goal is to instead estimate the conditional probability of image labels given the image, which is really what we are after in image segmentation. In addition conditional random fields require less training data and can depend on arbitrary sets of the observation data, not just that from neighboring regions. In

Kumar et al [53] spatial dependencies are incorporated into their model of natural and man-made regions by utilizing interaction potentials which consider features over the observations of several regions. In He et al [35] both features made up of several near by locations and global features made up of all image locations are used within a CRF model. The regional features capture local geometric relationships such as edges, corners and T-junctions. The global features capture the layout of image labellings.

In Rabinovich et al. [89] context in the form of class co-occurrences are used. The MSRC database used by Shotton et al. [106] contains 23 classes, however only a small number of them are present at any one time within a particular image. Certain classes tend to co-occur with other classes within the dataset more often than others. For example, a small yellowish region might be labeled a tennis ball or a lemon equally well. However, if the rest of the scene is labeled to indicate a tennis court then it is likely the yellowish region is a tennis ball. By incorporating class co-occurrence statistics into their CRF model Rabinovich et al. attempts to maximize object label agreement within a segmentation.

In Rousson et al. [94] a template shape is used to a guide the contour evolution such that curves matching the given shape are favored. The template is allowed to undergo affine transformations thus allowing some variability of the shape within the image. An energy function is proposed and the Euler-Lagrange equations of this are added to the segmentation method of Paragios et al [85], effectively constraining their material segmentation method to also fit the desired shapes. All the Euler-Lagrange equations are embedded into a level set function and iteratively updated. The parameters of the affine transformation, which come from the current level set function (i.e., labels), are what makes this a global function. Symmetry has also been used in this way [92] to prefer contours that have symmetric shapes. This approach is useful when the goal is to find objects that tend to have natural symmetry such as butterflies.

Recently, Hoiem et al [40] have proposed using their scene segmentation method [39] as a guide to an external object classifier. Specifically they have used the segmentation of

an outdoor scene into ground/vertical/sky to determine a likely ground plane and camera position within a particular image. This information is then used to guide a pedestrian or vehicle classifier by effectively limiting the position and scale of positives to ones consistent with viewing perspective.

It is along the lines of Hoiem et al that we attempt to incorporate global information so as to improve an initial local segmentation. Similar to their local method we wish to throw a large number of global features at our system and allow it to decide which features are important. We will do this with an Energy Based Model [57] trained using contrastive divergence [37] on synthetically generated data from the ground truth data in our training set. In chapter 5 we describe the details of this model and the global features that we use.

Chapter 3

Glass Edges

This chapter addresses the problem of finding glass objects in images, focusing on the identification of their internal and external image contours in the output of an edge detector. Typical glass objects such as bottles, glasses, plates, and vases have rather smooth surfaces, display strong highlights but weak diffuse reflectance, and—as pointed out by Murase [79]—reveal a distorted version of the texture of surfaces lying behind them (Figure 3.2). These properties are used to identify a number of visual cues associated with the image regions surrounding glass edges. Given a set of training images with hand-labeled glass edges, we then use these cues to learn a hierarchy of classifiers and identify fragments of glass boundaries in new pictures (see [49] for related work in the image segmentation domain). A global integration step merges these fragments and uses snakes to identify their support regions as potential glass objects.

3.1 Edge Detectors and Glass

We have observed that the edges of glass objects show up rather reliably in the output of the Canny edge detector [12]. As shown in Figure 3.1 the edges of a glass plate and tea cup are present in the edge detector's output. Though the edges of glass objects show up in many types of edge detectors the canny edge detector is particularly reliable. There are two reasons for this. First most glass objects that we encounter have physically smooth boundaries. This

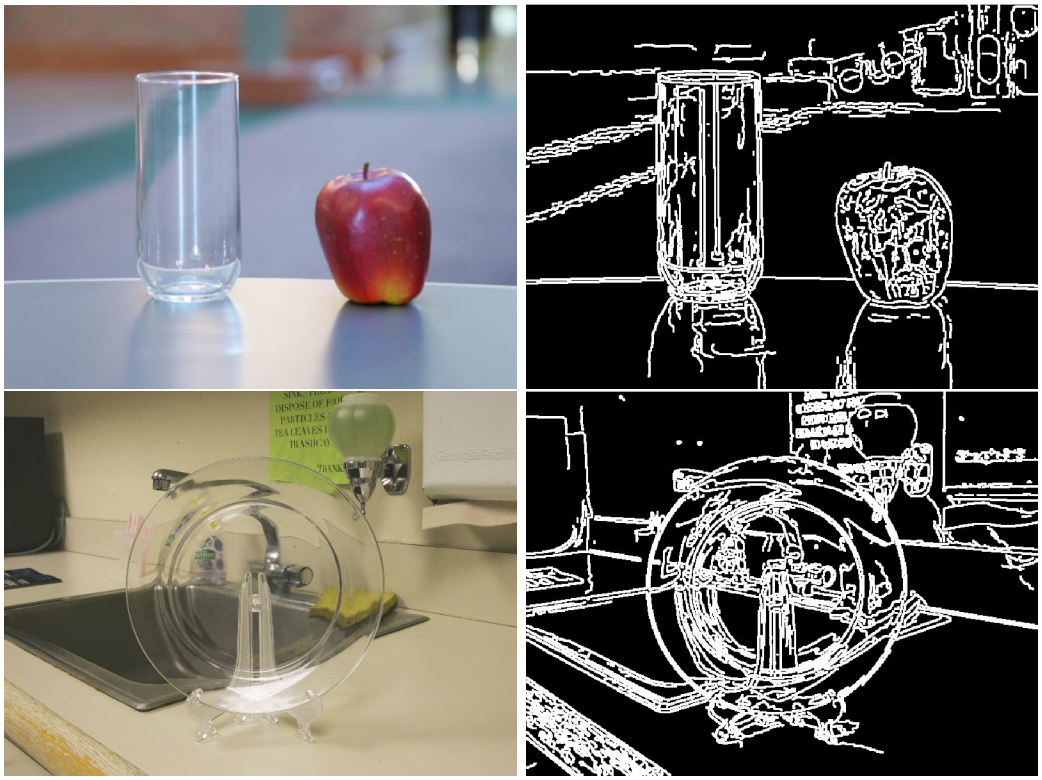

Figure 3.1: *The edges of glass objects show up rather reliably in the output of the canny edge detector. Given these edges the task then becomes identifying those that belong to transparent objects as opposed to opaque objects.*

combined with the reflective and refractive properties of glass results in many strong edges. For example since glass is highly specular the smooth transition across surface normals often results in a reflection angle with the light source and a resulting highlight. Secondly these boundaries tend to have strong refractive effects with their appearance depending on objects far off in the scene. The result again is a rather strong edge if the refracted part of the scene doesn't match the background at that position in the image (which is often the case). These strong edges will show up in most edge detectors however the hysteresis step within the canny edge detector will not only find these edges but recover weaker edges as well. As one might expect, when these reflective and refractive effects are not present, glass objects will have very weak edges (with the change in intensity being small across the edge since they appear the same on both sides). However with the strong highlights and refractions

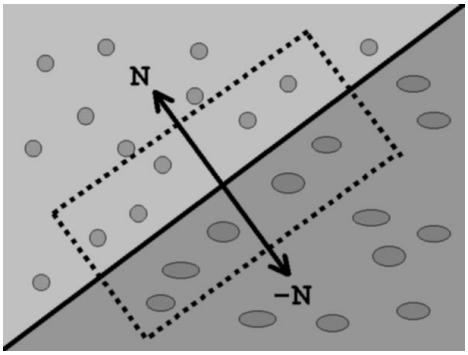

Figure 3.2: *An edge separating two regions. To determine if the edge is caused by a glass surface we examine a small part of the edge and compare the patches on both sides. If the edge is from a glass object then both sides should be similar and distorted. The distortion is required in order to distinguish the edge from other edges within textures.*

along the boundary a strong edge will allow the for the low threshold to be used as the edge is followed along the boundary and across any weak edges.

Because glass edges show up so reliably we attempt to identify glass objects within an image by labeling edges as resulting from glass or not. The edge detector output is broken into small fragments. Given a set of cues which search for properties of glass we then attempt to label these small edge fragments.

3.2 Edge Cues

Given contour fragments from an edge detector we would like to identify those that are associated with glass. Kaufhold and Hoogs [49] use various cues to remove edges that a person would not consider to be true object boundaries. Basically, the goal is to keep edges that are between two non-similar regions. Here we are not concerned with true object boundaries [67, 68] but instead with internal and external edges of glass regions. Since glass is usually clear, we focus on its refractive/reflective properties. Assuming that a sample of the background is visible and undistorted on one side of a glass edge, we identify five cues to the presence of such an edge.

- **Color Similarity (C):** the color tends to be similar on both sides (Section 2.2.1).
- **Blurring (B):** the texture on the glass side is more blurry (Section 2.2.2).
- **Overlay Consistency:** the intensity distribution on the glass side is constrained by intensity distribution on the non-glass side by alpha **(A)** and emission **(E)** values (Section 2.2.3).
- **Texture Distortion (D):** the texture on the glass side is slightly different (Section 2.2.4).
- **Highlights (H):** the glass side has a specularity (Section 2.2.5).

Note that internal edges in glass regions often correspond to sudden changes in object thickness or shape, and one can still consider that the texture on one side of the edge is a distorted version of the texture on the other side.

Each of the fives cues used is characterized by one or two scalar values that provide information relevant to glass properties. Similar color distributions and high alpha values indicate possible transparency. Blurring and distortion deal with refractive properties, while emission and highlights deal with reflective properties. Below we describe how the values for each cue are obtained.

3.2.1 Color Similarity

The value returned provides an indication of how similar the colors are across the edge. To measure this a histogram is constructed for each side containing twenty bins for range of hue values and twenty bins for the saturation values. The intensity component is ignored in order to make the measure somewhat invariant intensity differences. The two histograms are normalized to have a sum of one and the euclidean distance between them calculated. This distance is the value returned.

3.2.2 Blurring

In our experiments it has been observed that the side covered by glass is often a smoother version of the other side. We explain this as the result of impurities in the glass causing imperfect refraction as well as minor surface variations which create different optical properties along the surface. Thus the value returned by this cue should indicate how much smoother one side is compared to the other.

In Forsyth and Fleck [31] texture smoothness is measured by subtracting a smoothed version of an image from the original. Already smooth areas will have small differences where as highly textured areas will have large differences, thus this difference gives an indication of the smoothness of the texture. To measure the smoothness of a 2D sample we instead use the discrete cosine transform. Once the two sides are transformed we compute the mean of the frequency coefficients on each side. The difference among the two means can then be used to indicate relative smoothness (as being a shift into the lower frequency range of the spectrum). For this measure to make any sense we must consider how textured the background was to begin with. A small shift on a highly textured background could just be noise while on a smooth background is much more significant. We thus normalize the measured shifts by a measure of the texture entropy which we take to be the standard deviation of the intensities on that side of the edge. Another method for measuring texture smoothness used by Forsyth and Fleck [31] involves subtracting a smoothed version of image from the original. Already smooth areas will have small differences where as highly textured areas will have large differences.

3.2.3 Overlay Consistency

Rather than using the constraints of Adelson et al [2] we directly use the model given in Eq. (1). The two parameters α and e can be used to identify edges exhibiting transparent

overlay effects. An α value near 1 and a low value of e indicates that the edge is likely an intensity change in a textured region. An α less than 1 and a small e may indicate that the edge belongs to a transparent object. A low α and/or high e indicates a strong intensity change indicating an edge between two very different regions. Thus these two parameters will make up the values of this cue.

Equation (2.1) is a linear in two unknowns, and can thus be solved given at least two distinct intensities on each side of the edge. This is interesting and makes sense since given a homogeneous background there is no reason to assume the presence of a transparent overlay rather than just a change in intensity. Since Eq. (2.1) is a 1D affine transformation resulting in only a scaling and translation (and assuming the texture is consistent on both sides of the glass edge and that our sample is large enough) we can assign I and I_B by clustering the intensities on the two sides of the edge by percentages. In other words we can say that the top 10% brightest on one side is equal to the top 10% on the other side and so on. The mean intensity of the clusters of one side becomes the values of I_B in Eq. (2.1) and the means of the clusters on other side the values of I. The value of α and e can now be solved as a linear least squares problem. The value of α should be between 0 and 1, so if it is negative we simply switch the values of I_B and I.

One may ask that since glass is clear shouldn't α always be 1 and e always be 0 for glass edges. In practice the α value actually tends to be a bit less than 1 on glass edges, again likely due to the smoothness phenomena. Also, the emission value appears to correspond with weak reflections in the glass.

3.2.4 Texture Distortion

Because of refraction we can expect that a textured background will be magnified and/or skewed when seen through glass. The value returned from this cue should provide information as too how similar the texture is on both sides of the edge.

To measure the difference in texture we filter both sides of the edge with a bank of

gaussian filters consisting of six orientations and two scales [64]. Distributions of the filter outputs are created for both sides of the edge. The similarity of the texture on the two sides is then measured as the euclidean distance between the two distributions.

3.2.5 Highlights

Glass is known to be highly specular, making highlights a valuable cue. The value returned from this cues is a binary value indicating if a highlight was detected on one side of the edge.

As discussed by Klinker et al. [51, 113] highlights can be found in color images by assuming a dichromatic color model and looking for dog leg like structures in a plot of the colors observed in small windows. While an effective technique in areas containing monochrome surfaces, searching for the dog leg structures in textured areas can be a nuisance. Other techniques, such as that of Brelstaff et al. [11] also assumes uniform albedo. Since highlights are usually the brightest part of the image (being reflections of the light source) a simple technique is to threshold based on intensity (e.g. keeping pixels above 90% of the brightest value). However we can do better. Clearly it is desirable to set the threshold as low as possible in order to get as much of the highlight as possible. At the same time we do not want to introduce bright patterns resulting from non-specular reflections.

We use the following heuristic to detect highlights in a reliable fashion. First note that highlights on smooth shiny surfaces such as those of most glass objects tend to have a profile such as the one shown in Figure 3.3, where a sharp spike is overlaid on a (locally) smooth intensity profile due to diffuse reflection and/or, in the case of transparent objects, background texture. Various analytical models for the shape of this spike exist, from the non-physical Phong model commonly used in computer graphics, to physical models related to the microscopic roughness of the surface, such as those proposed by Healey and Binford [36] and Nayar et al. [80]. For our purposes, it is sufficient to note that, on each side of the spike and over most of its (small) extent, the intensity profile is close to a straight line. In turn, this means that the perimeter P of a two-dimensional highlight region found by thresholding

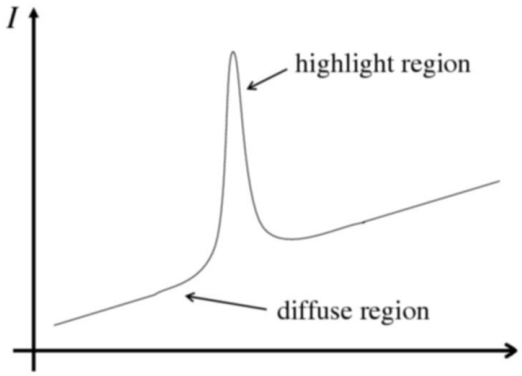

Figure 3.3: *Typical profile of a highlight on a smooth specular surface. The diffuse component increases fairly linearly with the changes in surface normals of smooth objects. The specular component however changes quickly and abruptly as the surface normal lines up so that the angle of reflection is opposite the direction to the light source.*

should also be a roughly affine function $P = aT + b$ of the threshold T. The perimeter of a bright diffuse region, on the other hand, tends to be (roughly) piecewise constant.

Thus to find the proper highlight threshold we create a plot of highlight perimeter as a function of intensity threshold. Once an image has been thresholded, such that pixels below the threshold are black and pixels above are white, we estimate the perimeter simply by counting the number of sides on each white pixel that face black pixels. As can be seen in the middle plot of Figure 3.4 the tip of a spike can be seen lying on its side near the end of the plot. As argued above, we can approximate the edge of the spike by a straight line. Thus to find the threshold we iteratively fit a line to the perimeter values , starting from a threshold of 1.0 and plot the fit error (see Figure 3.4). As can be seen the error is nearly 0 until the bottom of the spike region is hit (indicated by an arrow in the figure). An example is shown in Figure 3.5 where the threshold is set to 0.71 for a glass plate. Had the threshold been set to a fixed value of 90% of the brightest intensity, a value of 0.87 for this image, much of the highlights would have been lost.

By setting the threshold in this way we can set it as low as possible to get as much of the highlight without worry of including mostly bright Lambertian regions. For example if a white sheet of paper is placed in the image at an appropriate angle as to be fairly bright,

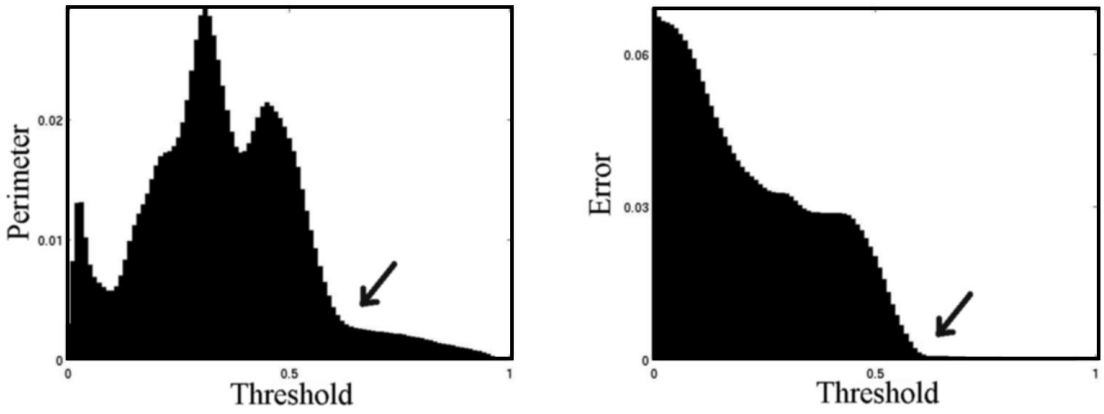

Figure 3.4: *The relationship between intensity threshold and perimeter of iso-intensity rings within highlights.* **Left:** *Plot of perimeter as a function of the intensity threshold.* **Right:** *Plot of line fit error to perimeter in the previous plot as a function of the intensity threshold. The intensity at which the error is no longer small is chosen as the threshold to determine highlights. The chosen threshold of 0.71 is indicated by an arrow.*

the threshold would be set just above its maximum brightness. While this is what we want it is undesirable for this to suppress highlights through out the image. We overcome this by examining sub-windows instead of the whole image at once.

3.3 A Local Classifier

Each of the parameters, except for highlights which are either 0 or 1, should occupy a small range of values at a glass edge: color differences should be small, smoothness shifts small but non-zero, alphas near one, emissions low, and distortion somewhat high. To learn the space of parameters occupied by glass we use a support vector machine [10, 20, 116] and attempt to solve the following optimization problem:

$$\begin{aligned} min_{w,b,\xi} \quad & \frac{1}{2}w^T w + C\sum_{i=1}^{n}\xi_i \\ subject\ \ to \quad & y_i(w^T\phi(x_i) + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \end{aligned}$$

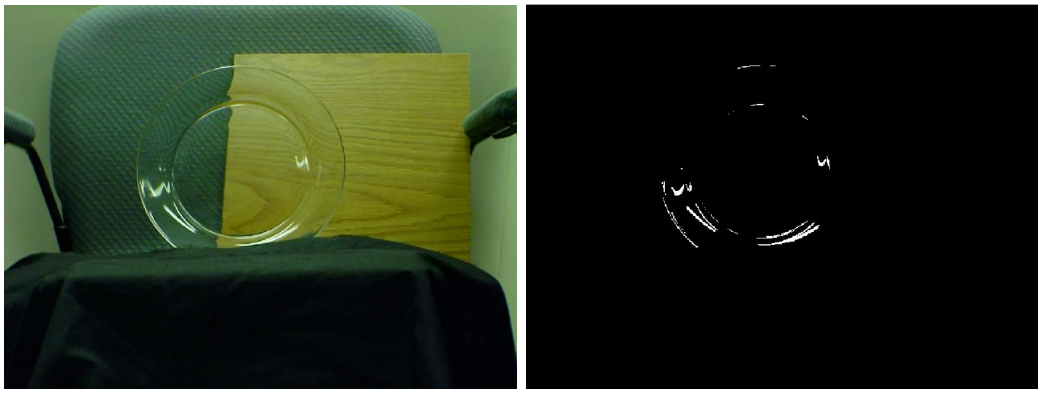

Figure 3.5: *A glass plate and its detected highlights at a threshold of 0.71.*

where x_i is a feature vector, y_i is its corresponding label, w and b represents a plane separating our positive and negative data. The desired plane is the one that maximizes the margin between the two classes. The variable ξ_i, known as a slack variable, allows for soft margins when a perfectly separating plane can not be found. The variable C regulates how costly this slack is. Separating planes in spaces of higher dimension than our feature vector can be considered by the mapping function $\phi(x_i)$. In the dual form of this problem [116] this takes the form of a kernel function of the form $K(x_i, x_j)$. For our experiments we utilize the following gaussian kernel:

$$K(x_i, x_j) = exp(-\gamma||x_i - x_j||^2)$$

This kernel is convenient in that it has only one parameter, γ, and is known to often out perform other kernel choices.

We must be very conservative when labeling an edge as glass. There are many situations in which non-glass edges are similar to glass edges as defined by the cues. One example is a lined surface such as the mat shown in Figure 4.7. Across the lines there is a similar color distribution, the possibility of a low alpha value and non-zero emission value (due to shading). Something similar could be said for shadows. However, we have another cue in that we know the glass in the image encloses a region. Thus if the precision of the classifier is somewhat high we can use a global integration step to link sparsely found glass edges

together (Section 5). Thus our goal when training the classifier should be to minimize the number of false positives.

A method of minimizing false positives is to have a large number of negative examples and weighting the negative data in the training set more than the positive data by giving negative examples a larger C value in the optimization function above. Thus to discourage calling such edges glass we can collect a large collection of negative samples embodying as many cases of such undesirable edges as possible and then give them a high weight. Even with a large weight forcing no false positives on the training data the learned classifier tends to have too many false positives on test data. We encounter this problem due to our quantized view of the space. We would like the number of false positives to be extremely low, perhaps one in one million edge samples. In order to reach such a rate we would have to acquire an amount of negative samples that would be impractical to train a classifier with.

A more practical option to limiting false positives on test images is to set the recall rate low enough such that only the strongest true glass edges are labeled as glass and the weaker glass edges along with these false glass edges are labeled as non-glass. To do this we turn to another technique to adjust the recall of an SVM which is to adjust the position of the hyperplane away from the negative data. The hyperplane produced by the above optimization has the form:

$$y = w^T \phi(x_i) + b$$

where w represents the planes normal and b represents its offset. By shifting the offset parameter b so that the decision boundary passes into the positively labeled data we can increase the number of negatively labeled data and hopefully minimize our number of false positives. In our experiments we iteratively shift b until the number of true positives drops below 30% on the training data.

3.4 Multiple Classifiers

Let us consider the range of values for each of the cues parameters in glass as logical propositions. From this we make the observation that for an edge to be glass only two of these propositions must always be true given that one of the other four is true as well:

$$\begin{aligned} glass \Leftarrow\ & similar\_color \wedge high\_alpha \wedge \\ & (low\_emission \vee highlight \vee \\ & smoother \vee distortion) \end{aligned}$$

The above statement can be re-written as four different statements of three propositions:

$$\begin{aligned} glass &\Leftarrow similar\_color \wedge high\_alpha \wedge low\_emission \\ glass &\Leftarrow similar\_color \wedge high\_alpha \wedge highlight \\ glass &\Leftarrow similar\_color \wedge high\_alpha \wedge smoother \\ glass &\Leftarrow similar\_color \wedge high\_alpha \wedge distortion \end{aligned}$$

The underlying structure comes from the fact that the color and alpha parameters are sufficient in eliminating most non-glass edges. What is left is to distinguish among the two types of edges that satisfy the proposition associated with these parameters, internal texture edges and glass edges. Thus we need information from at least one, not all, of the other cue parameters.

Since the problem has this structure it may prove beneficial to use it and learn four classifiers in the lower three dimensional spaces of the four statements above, rather than a single classifier using all six cue parameters. To combine the outputs of these classifiers we consider the three methods described below.

3.4.1 Logical OR

The most obvious means of combining the classifiers is to OR their outputs. That is if one classifier labels an edge as glass then the edge will be labeled glass.

3.4.2 Weighted Sum

An alternative to OR'ing the classifier outputs is to weight them and see if their sum is above some threshold. By combining the cues in this way we are able to give less weight to cues that do not perform as well as the rest. It may also be the case that sets of cues labeling an edge as glass is better than considering each one separately. If this is the case then no one weight will be above the threshold.

One method of finding the weights and threshold is to use gradient ascent to maximize the classifier's accuracy on the training set. However, one should notice that the parameters we are looking for define a support vector classifier with linear boundaries.

3.4.3 Exponential Model

One can combine boolean classifiers so as to return a probability of a given label given their outputs. Maximum likelihood is used to learn the parameters of an exponential model, whose form is given by entropy considerations as:

$$p(y|x) = Z(x)e^{\sum_i \lambda_i f_i(x,y)}$$

where $p(y|x)$ is the probability of a label y being assigned to the SVM classifier outputs x [7]. λ_i is a weight for a particular feature defined by the function $f_i(x, y)$. These feature functions return a value of 1 if x takes a particular form given y and 0 otherwise. $Z(x)$ is a normalizing factor. The parameters are found with generalized iterative scaling [26]. Once the parameters are found we can classify a particular set of sub-classifier outputs by

thresholding on a probability of 0.5.

3.5 Global Integration

Once a part of an edge has been classified as glass we should more easily believe that the rest of the edge is glass as well. We can increase the number of positive results by connecting two edge samples labeled as glass if they have a path connecting them. As will be described in the following section, edges from the edge detector are broken into smaller edge samples for classification. Some of these may be labeled as glass while some may not, even if the whole edge is glass. These false negatives can occur if the assumption is locally violated (i.e. the background is not the same on both sides) or because of the conservative training of the classifier. This hysteresis style of approach provides us with a means of recovering these falsely labeled glass edge samples.

We now have a number of long edges that are labeled as glass. To find the glass region that these edges enclose we can use an active contour approach such as the one described by Kass et al. [48]. Initialized at the boundary of the image a polygon is iteratively updated so as to minimize an internal energy function enforcing smoothness and an external energy function preferring to be near edges. Once the contour has converged all pixels within its boundary are labeled as glass.

3.6 Experiments

Given an image we find edges with the Canny edge detector. From an edge point we then walk 25 pixels in each direction, and using the center pixels normal we take a sample 25 pixels outward in the positive and negative direction. The edge samples are then transformed into a 6-vector using the described cues. The results from the various cues can be seen in Figure

Classifier	TP Rate	FP Rate	Precision
Single SVM	47.01%	3.09%	68.76%
Multiple SVM's + OR	88.30%	10.04%	56.04%
Multiple SVM's + Weighted sum	83.94%	8.53%	58.78%
Multiple SVM's + Exponential model	88.30%	10.04%	56.04%
Multiple SVM's + Weighted sum (sampled)	79.72%	4.12%	73.7%

Table 3.1: *Results from the described classifiers, all tested on a test set of 50 images where glass pixels were marked by hand. The true positive rate, or recall, indicates the percentage of glass pixels that were correctly identified. The false positive rate indicates the percentage of non-glass pixels that were identified as glass. Precision is the percentage of identified glass pixels that are actually glass.* **From top to bottom***: an SVM trained on all six values of the cues, a classifier consisting of the OR'ed output of the four sub-classifiers as described in section 4.1, a classifier consisting of a weighted sum of four sub-classifiers as described in section 4.2, and a classifier consisting of the outputs of four classifiers combined through an exponential model as described in section 4.3. The last row contains results from a classifier consisting of the weighted sum of four sub-classifiers trained on a subset of the training data as described in section 6.*

3.7.

To train the classifier 15 images were used. Six of these images contained glass objects in front of various backgrounds. Edges were manually labeled and broken into 333 positive training samples. The other nine images contained no glass. An edge detector provided the edges which were then broken into 4581 negative samples. To test the various classifiers a set of 50 images was used. Thirty five of these images contained glass objects in front of various backgrounds. Glass regions were manually labeled and stored in a separate image mask. The remaining fifteen images contained non-glass objects.

To measure how well the classifiers found glass each one was run on the set of 50 hand labeled test images. The regions identified as glass by the classifier are then compared with the masks associated with each image. True positives are measured as the number of pixels within the intersection of the two regions while false positives were measured as the number of pixels within the snake but outside the mask. The results can be seen in Table 4.1. The classifier trained using the six cues has a recall of 47% and a precision of 68%. The classifiers made of sub-classifiers have higher recall rates, all around 80%, but at a somewhat lower

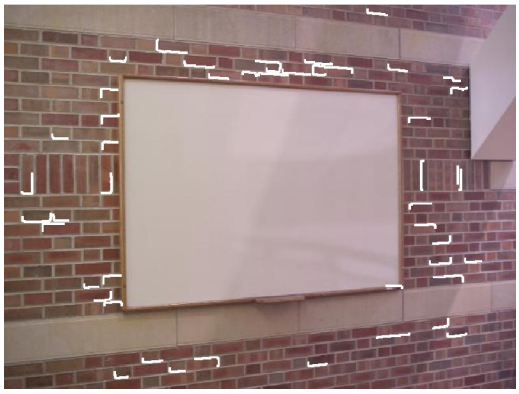

Figure 3.6: *Test image showing edge samples that are falsely labeled by the local classifier (shown in white).*

precision.

In addition to the four classifiers discussed before we experimented with the idea of training a classifier on subsets of the training data rather than the whole set. The subsets were chosen randomly so that we had 50 positive samples and 100 negative samples. A classifier trained on such a subset was considered good if it could correctly classify the edges on the difficult image discussed earlier of a glass plate on a lined mat. The best of these classifiers is shown in the last row of Table 4.1. This classifier out performed the others with a high recall rate as well as a high precision.

Six examples where the glass was successfully identified are shown in Figure 4.7. Figure 3.6 shows an example of where the classifier falsely labels edges as glass. The falsely labeled edges are the result of the large texture pattern in the image. Since we are looking at each side of the edge through a window of finite size we fail to get accurate statistics from larger texture patterns. In this example the color is similar on both sides of the edges along the internal of the brick pattern. The alpha value is likely high as well. Due to our limited view, the texture appears to be different on each side of the edge, thus these non-glass edges are labeled as glass. It should be pointed out that in this image there are a significant amount of edges within the brick that were correctly classified as non-glass.

3.7 Discussion

This chapter has described a method for identifying glass edges in an image. It is not limited to glass since most of the cues we have used are valid for other smooth transparent media. It does, however, assume that the background is similar on both sides of all glass edges. This is of course not true when refraction is too strong, or when transparency is overwhelmed by very bright highlights. Assuming there is one transparent object present the edges identified to be glass can be grouped together to produce a region within the image.

Unlike traditional material and texture recognition methods the features used here are inherently binary, comparing one side of the edge to the other. We will expand on this in the next chapter and move away from identifying glass edges and towards identifying glass regions. This will also allow us to segment multiple glass objects within the same image as no assumptions about the number of glass objects present will be necessary.

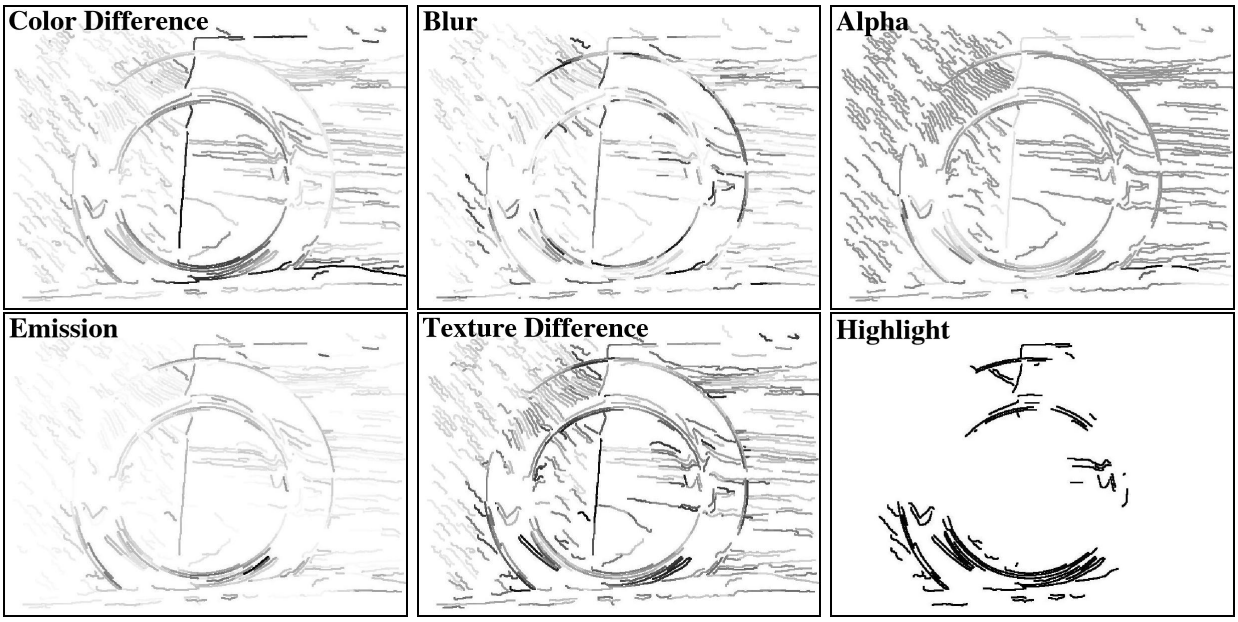

Figure 3.7: *The values provided by the six cues on the image of a glass plate in Figure 3.5. In each image darker edges indicate higher values.* **Top left:** *The difference in the color distributions across the edge as measured by the method in section 3.1. Notice the highest value along the edge of the wooden board and fabric behind it which has a very different color distribution.* **Top middle:** *The amount of blurring across the edge as measured by the method in section 3.2. There are few edges with large blurring values within consistent texture regions.* **Top right:** *Alpha values across edges as measured by the method of section 3.3. The edge of the board and fabric is not visible since being very different in intensity distribution the alpha value here is very low.* **Bottom left:** *Emission values across edges as measured by the method in section 3.3. Again the edge of the wooden board is visible as well as much of the glass plate.* **Bottom middle:** *Measure of difference in texton distributions as measured by the method of section 3.4. Notice here that the lowest values exist within a consistent texture.* **Bottom right:** *Highlights, a binary value determined by the method of section 3.5. A value of one exists along edges that have a detected highlight to one side.*

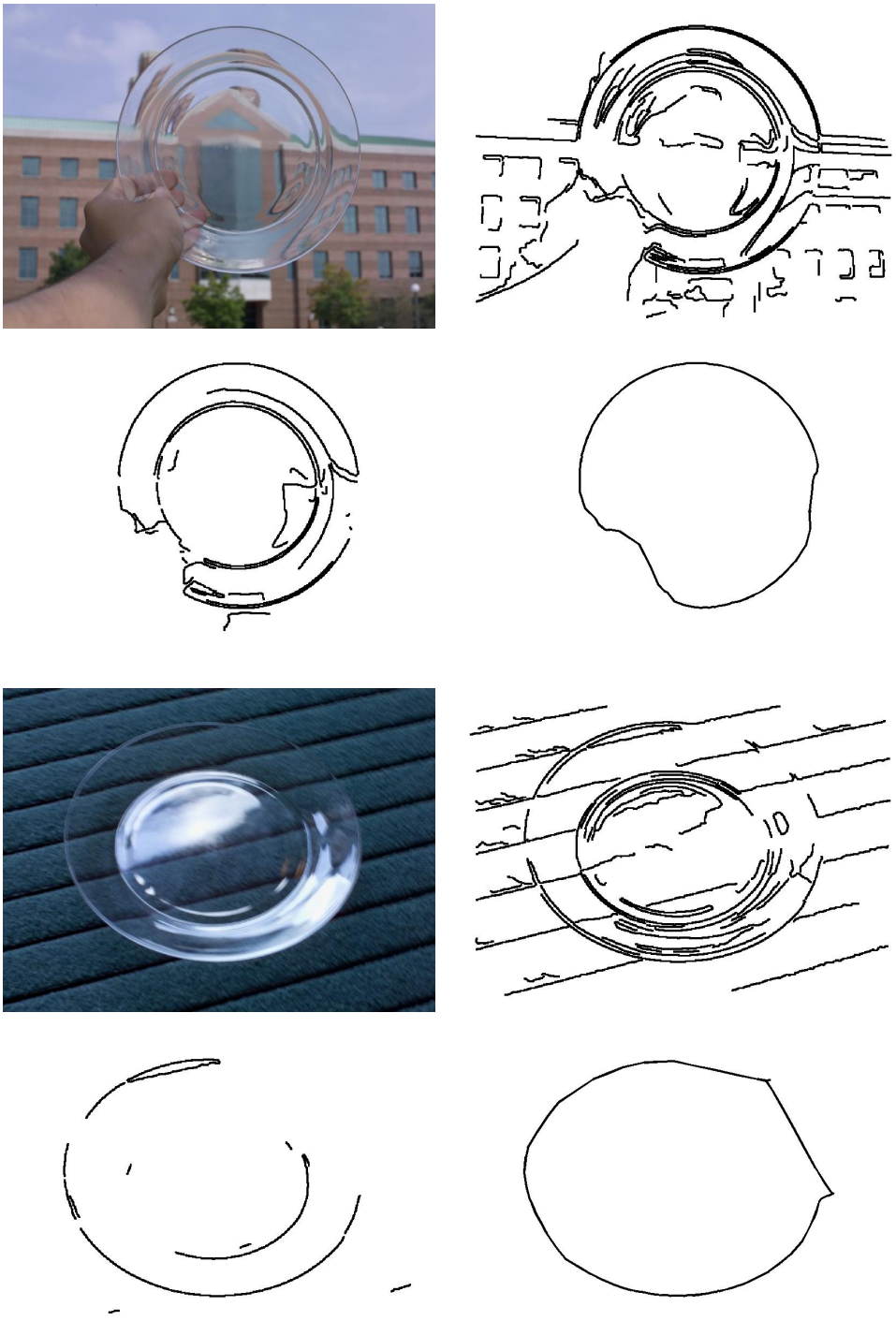

Figure 3.8: *Test images and output from the described system.* **Top left:** *Test image,* **top right:** *edges from edge detector,* **bottom left:** *edges that have been labeled as glass,* **bottom right:** *regions of glass found by shrinking a snake around the labeled glass edges.*

Figure 3.9: *Test images and output from the described system.* **Top left:** *Test image,* **top right:** *edges from edge detector,* **bottom left:** *edges that have been labeled as glass,* **bottom right:** *regions of glass found by shrinking a snake around the labeled glass edges.*

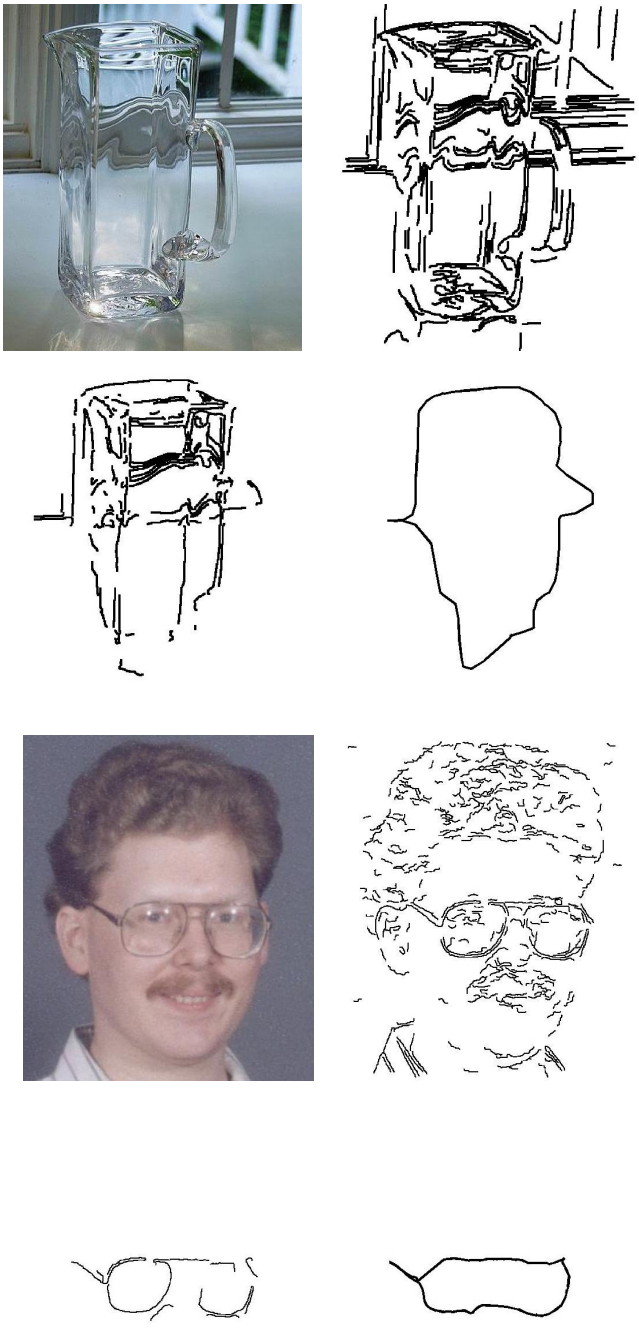

Figure 3.10: *Test images and output from the described system.* **Top left:** *Test image,* **top right:** *edges from edge detector,* **bottom left:** *edges that have been labeled as glass,* **bottom right:** *regions of glass found by shrinking a snake around the labeled glass edges.*

Chapter 4

Glass Regions

This chapter describes an alternative, region-based approach to the problem of segmenting images into glass and non-glass components. By focusing on regions, we alleviate the need for a final grouping stage as needed in an edge-based method, and thus do not have to assume that only one glass object is present. Our method can be outlined as follows: Since the appearance of glass objects depends for the most part on what lies behind them, we propose to use binary criteria ("are these two regions made of the same material?") rather than unary ones ("is this glass?") to guide the segmentation process. After an initial set of homogeneous image regions has been identified by a traditional segmentation algorithm [29], pairs of regions are related by an *affinity* measure, providing some indication as to how well two regions belong to the same material; and a *discrepancy* measure based on how much one region looks like a glass-covered version of the other, providing a basis for separating regions made of different materials. We combine these two complementary measures into a single objective function. Though optimizing this function is a combinatorial problem, we show that it can be modified to fit into the geodesic active contour framework [15, 16], allowing a solution to be found efficiently.

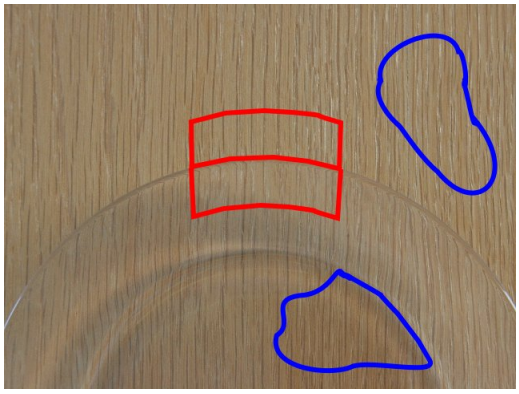

Figure 4.1: *In the previous chapter region samples around edge snippets are compared in order to classify the snippet as glass or not glass based on similarity and distortion between them. Here, our goal is to check whether one region is a glass-covered version of another, and we may potentially compare any two (blue) regions in the image.*

4.1 Characterizing Glass Regions

In the previous chapter, rectangular samples are taken from opposite sides of small edge snippets from the output of the Canny edge detector and compared using a series of cues associated with characteristics of glass: transparency, refraction and reflection. The cues operate on the assumption that one sample should be some arbitrary piece of background and check whether the other looks like a glass-covered version of that piece. The two regions should look similar in terms of color, and somewhat similar in terms of texture. Since glass is refractive, we might expect the texture to be slightly different due to distortion. Because of reflection, we might also expect to see specularities as well.

While these cues hold most strongly at edges between adjacent regions, the same cues can in fact be applied between any pair of image samples (Figure 4.1). Rather than focusing on rectangular region samples along image edges, we use the homogeneous texture/color regions found by the graph-based segmentation algorithm of Felzenszwalb et al. [29] as an initial partition of the image. These regions that are then compared as explained in the rest of this section to identify glass/non-glass pairs. It is important in our case that the initial regions provide an *over*-segmentation of the image, which is easily achieved by appropriately setting

the parameter that governs the behavior of the algorithm described in [29]. In addition, we have modified the measure of texture/color discrepancy between pixels proposed in [29] to add weight to edge pixels and prevent regions from bleeding over Canny edges, which have been shown in the previous chapter to be reliable cues for glass boundaries. Finally, small and thin regions are merged with their neighbors.

Ideally, given these initial regions, we would like to define a single homogeneity measure expressing how much regions from the same material belong together. Since we are separating glass from non-glass, this means giving high values to regions within glass, high values to regions not within glass, and low values to regions when one is in glass and one is not. As explained below, this does not seem possible, due to the optical properties of glass and other transparent materials. We propose instead to quantify how much pairs of regions don't belong together, or do, using two separate *discrepancy* and *affinity* terms.

4.1.1 Discrepancy

In the previous chapter, we proposed to find glass edges using a classifier trained on cues between two regions incident to the same edge, assuming one is a glass-covered version of the other. We will again utilize these cues this time to help identify glass regions. We do not use the blur cue here since it relies on the discrete cosine transform which is appropriate for rectangular samples but not for arbitrary region shapes.

The five cues (A, C, D, E, H) together provide us with a five-dimensional space to characterize the *discrepancy* between two regions —that is, a measure of how these regions are unlikely to come from the same material, in terms of one of them looking like a glass-covered version of the other. Rather than train a classifier in this five-dimensional space, the cues are grouped into three different classifiers of three terms each: ECA, DCA, HCA, and their outputs combined using a fourth linear support vector classifier which effectively weights the reliability of these sub-classifiers.

It has been shown in [87] that the value returned by a support vector classifier —that is,

the signed distance of a data point to the dividing hyperplane— can be used as a probability with a proper mapping. This is done by fitting a sigmoid to the training data, which tends to give the positive training data very high probabilities and the negative ones very low probabilities. The range of values for which a test point can take on an intermediate value is small, and values increase quickly. This is fine when the classifier is rarely wrong, but for our purposes it is not desirable to have most of the assigned probabilities at the far extremes. In addition, our setting does not require discrepancy (or for that matter, affinity) measures that can be interpreted as probabilities, so we drop the log function and simply scale the SVM output so that on the training data the minimum output is 0 and the maximum is 1. This will be our discrepancy measure D_{ij} between regions i and j.

D_{ij} can be thought of as a distance separating regions depending on how much one looks like a glass-covered version of the other. If this is the case, a high value is returned. When the two regions look completely different, a low value is returned. Unfortunately, a low value in no way means that the two regions are both glass or both background. Consider the four regions shown in Figure 4.2. Let us call the two background types A, B and their glass-covered versions A', B' respectively. In this example, we expect $D_{AA'}$, $D_{BB'}$ to be high and D_{AB}, $D_{A'B'}$ to be low since they look very different. But what of $D_{AB'}$ and $D_{A'B}$? These pairs also look very different so will have low values, possibly lower values than D_{AB}, $D_{A'B'}$ depending on the background appearances. Thus, if we try to separate the regions into two groups, glass and non-glass, based on this measure alone, we may have many solutions. In the above example, both the correct segmentation (AB)(A'B') and an alternative (AB')(A'B) are likely just as good. In fact, we see that we can really only trust D_{ij} when its value is high.

4.1.2 Affinity

Since glass edges tend to show up reliably in images it is reasonable to try and use them to relate regions. In particular, when two regions are connected by an edge, and they are on

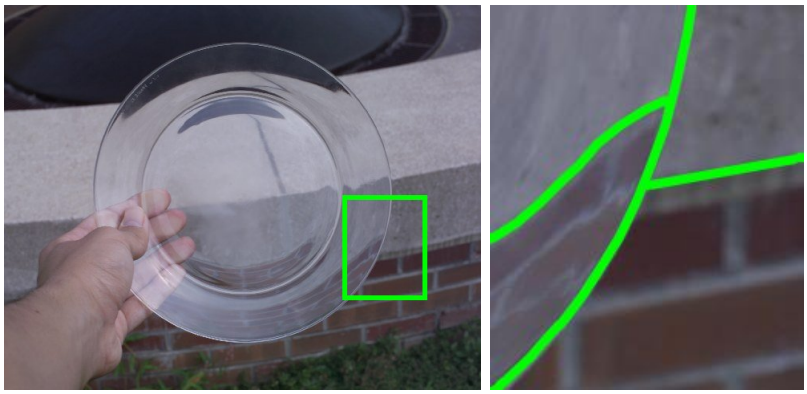

Figure 4.2: *Comparing two background regions to their glass-covered versions we expect to get a high discrepancy value. We will get low values when comparing the two background regions, the two glass regions, and the glass regions to the background region that does not correspond to it. This will allow for two possible segmentations, the correct glass/background segmentation and another with one glass region and the background that is not its own.*

the same side of that edge, they can be correlated by some *affinity* value A_{ij}. Determining whether a path exists between two regions is a simple matter of walking along the edgels that come near one region's border and following the edge to see which other region borders it comes close to. When there is more than one way to approach another region, we consider only the shortest path. The process of finding all the connecting paths can be implemented efficiently by converting the edge output into a graph where the nodes are edgels that are junctions (more than two neighbors) or ends (one neighbor), and the graph edges are the edgel paths connecting these nodes (possibly empty). In our implementation, we consider a region to be near an edgel when it has a point less than four pixels away. This too can be done efficiently by simply dilating each region by four pixels.

We would like the value of A_{ij} to be high when two regions come from the same material (i.e., glass or non-glass), and low otherwise. To do this, let us consider what can happen on the edge path between two regions. If this path is rough, with many large changes in orientation, there is a good chance that it is going through a textured region, possibly merged by smoothing and hysteresis, or has left one object and entered another. On the other hand, if the edge path is smooth, there is a good chance we are following a single object's contour.

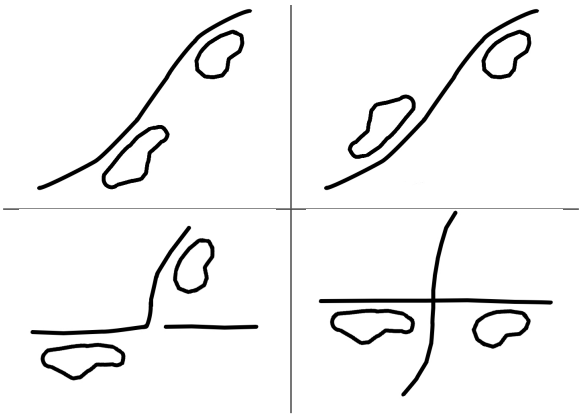

Figure 4.3: *Four possible scenarios that may occur when connecting two regions by an edge path.* **Upper left:** *Both regions are on the same side of a smooth edge path.* **Upper right:** *The regions are on opposite sides of an edge path.* **Lower left:** *Both regions are on the same side of a non-smooth path.* **Lower right:** *Both regions are on the same side of smooth path that is intersected by another path.*

Thus we define the affinity as

$$A_{ij} = 1 - \frac{a_{ij}}{\pi}$$

where a_{ij} is the maximum angle between consecutive edgel tangents on the path from region i to j. Straight paths will receive a value of 1, and paths with sharp turns will receive lower values. If there is no path between two regions, A_{ij} is set to 0. Edge detectors are known to behave oddly at junctions, possibly breaking one object's contour and connecting it to that of another. In many of these cases, an abrupt turn will result, thus it is beneficial to down weight sharp turns indicating our lack of confidence that they are true connections (Figure 4.3).

One may ask what happens when a background contour intersects a glass object. Would such an occurrence strongly correlate glass and non-glass regions? Due to the highly refractive nature of glass, most contours passing behind a glass object will appear broken (Figure 4.4). In turn, this will force a sharp change in direction on any joining paths between the two regions, resulting in a low affinity. Of course, under the right conditions —perhaps a flat object like a glass plate lying on top of a striped table— such intersecting

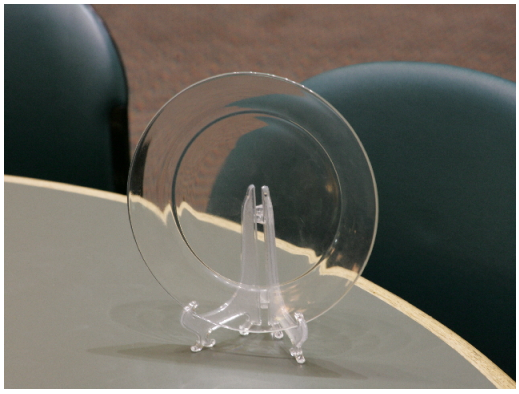

Figure 4.4: *Checking to see whether two regions are on the same side of a smooth edge is a good way of linking together regions that are both glass or both not glass. Edges that could incorrectly link glass and background regions by passing behind the glass object are usually broken by refraction, thus making their edge path non-smooth.*

contours will result in an incorrectly high affinity. We can spot such hazards with the edge graph already built. When the path between two regions comes across an intersection which two high correlating paths could pass through, for example two paths at right angles,[1] we can do one of two things. Following the proposed edge based method, we could evaluate each of the good paths through the junction and only allow information to pass along the most glass-like path, or just prevent all information from passing through the junction. In our experiments we do the latter. We also extend the notion of affinity to pairs of regions that don't have a direct edge path between them, but are connected through several edge fragments along adjacent regions.

Figure 4.5 shows typical examples of affinity strengths between sample regions and all others. Let us point out that there is a second advantage to glass having a high refractive component in that the area within and just outside a glass object's contour tend to have the glass boundary as the longest smoothest path around. Thus glass regions tend to be better correlated to one another than background regions are to each other.

We can think of A_{ij} as a measure of affinity between two regions, giving high values

[1]The topology of an edge detector's output is unreliable in most circumstances. However in the case of glass, it is a viable means of relating regions.

to regions within the same contour. However, a low value does not necessarily mean two regions belong in different groups. In fact in most cases it will mean that there is no edge path between them. As before, we can only really trust A_{ij} when its value is high.

4.2 Combining the Measures

We have been calling D_{ij} and A_{ij} discrepancy and affinity measures, but they may be better thought of as measures of certainty of discrepancy and affinity: When the discrepancy is 1, the regions compared are as likely to consist of different materials as can be inferred from the training data. On the other hand, when it is low, we cannot ascertain whether one region is glass, and the other is a piece of background. Similarly, when the affinity A_{ij} is high, the two regions are very likely to be part of the same material. On the other hand, a low value does not inform us much. Thus, it makes sense that a correct segmentation should maximize a combined certainty criterion drawn from the two measures at our disposal. To do this, we could maximize the following objective function:

$$E = \sum_{i \in G, j \in O} D_{ij} + \sum_{i \in G, j \in G} A_{ij} - \sum_{i \in G, j \in O} A_{ij} \tag{4.1}$$

with respect to the region labels ("(G)lass" or "(O)ther"). The first term enforces large discrepancies between glass regions and non-glass regions. The second term enforces large affinities internal to the glass. The final term penalizes a segmentation for breaking any large affinities between the glass and non-glass. Unfortunately, maximizing this function is a combinatorial problem: In practice, even though most initial segmentations only leave a few hundred regions, having to enumerate an exponential number of combinations of these is still far too expensive.

We propose instead to change the original objective function definition and concentrate

on the discrepancy between adjacent regions. In this context, it is convenient to define

$$D_{xy} = b_{xy} D_{i_{xy}j}$$

to express the local discrepancy at the pixel level. Here i_{xy} is the region that contains point x, y and b_{xy} is an indicator variable whose value is 1 if point x, y is on the border of this region. The variable j is then the region on the opposite side of the border. Similarly, we define

$$A_{xy} = b_{xy} \left(\sum_{j \in G} A_{i_{xy}j} - \sum_{j \in O} A_{i_{xy}j} \right)$$

to express the pixels' affinity to the glass set.

Optimizing an objective function based on A_{xy} and D_{xy} over all possible pixel labels is still a combinatorial problem. Thus, we treat every pixel as a sample of an underlying continuous function and relax the original combinatorial problem into a continuous one. For this, we use the geodesic active contour framework of Caselles et al. [15], and define an objective function over the smooth boundary (*contour*) $\mathcal{C}_s$ of an evolving image region:

$$E(\mathcal{C}) = \int_0^1 g(B_{\mathcal{C}_s}) |\dot{\mathcal{C}}_s| ds,$$

where g is a monotonically decreasing function (e.g. Gaussian), and $|\dot{\mathcal{C}}_s|$ is a regularization term that tends to minimize the length of the contour and maximize its smoothness. For regions defined in the discrete image domain, $B_{\mathcal{C}_s}$ is replaced by a term B_{xy} (boundary strength) given in our case by:

$$B_{xy} = \alpha D_{xy} + (1 - \alpha) A_{xy}$$

where α weights the significance of the discrepancy and affinity terms. We have defined D_{xy} and A_{xy} only at region borders since we want a subset of whole regions as our output.

Figure 4.5: **From left to right:** *a test image, its initial segmentation based on color and texture, two selected regions (red) and the affinity values of all other regions. Brighter intensities indicate higher values.*

We now wish to minimize this function. This can be done with a gradient descent method by differentiating over time [15]:

$$\frac{\partial}{\partial t}\mathcal{C} = g(B_{\mathcal{C}_s})\mathcal{K}\vec{\mathcal{N}} - (\nabla g(B_{\mathcal{C}_s}) \cdot \vec{\mathcal{N}})\vec{\mathcal{N}}$$

where $\mathcal{K}$ is the Euclidean curvature with respect to the curve and $\vec{\mathcal{N}}$ the normal. This can be solved in a level set implementation by embedding this within a function ϕ whose 0 level set is the current contour. Given an initialization for ϕ we adjust it with the following evolution equation [15]:

$$\frac{\partial}{\partial t}\phi = g(B_{\mathcal{C}_s})\mathcal{K}|\nabla\phi| + \nabla g(B_{\mathcal{C}_s}) \cdot \nabla\phi$$

In our implementation we initialize ϕ so that its 0 level set is near the border of the image and set α to 0.25 (Figure 5.4).

The evolution equation implicitly requires the curve to be smooth. We find this beneficial since trouble areas such as texture will usually have rough boundaries.

Note that by changing our objective function to fit the geodesic active contour framework, our final result may not be on the initial region boundaries (even though we restricted D_{xy} and A_{xy} to border pixels). This is wasted information, since we want a subset of these regions as our output. To account for this, we reset the contour every few iterations to match the boundary of the regions within it. This can be done by looking at the percentage of each region inside the current contour and removing all regions that have more than 50 % of its

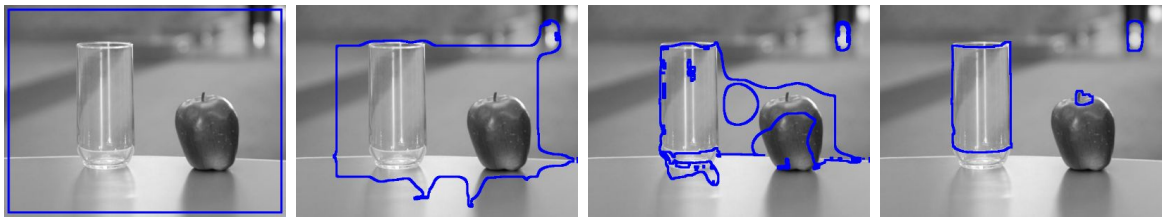

Figure 4.6: *Evolution of the level set (blue) as it segments the image into glass and non-glass. Note that two small regions have been classified incorrectly.*

pixels outside.

We further enforce the smoothness of the boundary with a simple post-processing step where we examine each of the regions just outside the border of what has been labeled as glass. If adding this region reduces the overall perimeter of the object then it too is classified as glass. This is repeated until no further regions can be added.

4.3 Experiments

We compare the proposed method to the ones proposed in previous chapter on the test data set of fifty images. Thirty five of the images contain glass objects in front of various backgrounds. Fifteen of the images contain no glass objects. In addition we train the glass classifiers used in D_{ij} on the same fifteen training images, six with glass objects in front of various backgrounds and nine with no glass at all. The results are shown in Table 4.1.

The proposed method obtains a precision (percentage of pixels labeled glass that are actually glass) of 77.03% which is higher than any of the edge based methods. Some sample segmentations are given in Figure 4.7. Notice the two distinct glass objects in the third row, which could not be found with the edge based method. A couple of false positives can be seen in the second row, though most of the image is segmented correctly. In the first row there is a conspicuous piece missing out of the right side of the plate. Though its difficult to see in the small image one of the external building regions partially bleeds into the plate allowing for some of the glass regions to have a somewhat high affinity with this region. When this

Method	TP Rate	FP Rate	Precision
SVM on all cues	47.01%	3.09%	68.76%
Multiple SVM's + Weighted sum	83.94%	8.53%	58.78%
Multiple SVM's + Exponential model	88.30%	10.04%	56.04%
Multiple SVM's + Weighted sum (sampled)	79.72%	4.12%	73.70%
Proposed method	61.73%	2.67%	77.03%

Table 4.1: *Results of the various edge based classifiers all tested on a test set of 50 images where glass pixels were marked by hand. True positive rate, or recall indicates the percentage of glass pixels that were correctly identified. The false positive rate indicates the percentage of non-glass pixels that were identified as glass. Precision is the percentage of identified glass pixels that were actually glass.* **From top to bottom:** *an SVM on all six values of the cues, a classifier consisting of a weighted sum of the sub-classifiers described in section 2.1, a classifier consisting of the outputs of the sub-classifiers combined through an exponential model, a classifier consisting of a weighted sum of the sub-classifiers and trained on only a sample of the training data, and the segmentation method proposed in this chapter.*

region is eventually removed it weakens those regions allowing the contour to pass by their boundaries. Similarly in the fourth row. In this case the orange region making up one of the wall planks crosses into the right bottle from time to time. Glass regions incorrectly linked to to this region are eventually weakened when the plank is removed. Of course if the initial segmentation merges some of the glass into the background we can not separate it later on. This can be seen in row two, where most of the base of the tea glass is merged into the table. Again in row five, the unmarked thin areas at the top of the bowl are part of the the blue region just above it. In row three an incorrect merging of part of the glass with the opening of the mug and the base of the bottle with part of the table results in false positives. In general, the initial segmentation does a good job of separating the region into uniform pieces of background and glass.

4.4 Discussion

This chapter has described a region-based segmentation method for finding objects made of transparent materials such as glass. Unlike the previous edge based method no assumptions

need to be made about the number of transparent objects present in the image.

As argued by Adelson in [1], the visual vocabulary of materials includes much more than just texture. The recognition of transparency is one such example, but finding metallic, dull, or even greasy spots in an image would require more than just texture as well. While characterizing a material's luster as metallic or mirror-like may be difficult when looking at a single image region, it might be accomplished using a binary measure as we have here. In the case of mirror-like surfaces, we might also compare two regions to see if one could be a reflected version of the other (possibly with some distortion).

We would like to point out that the general approach that we have described here can also be applied to the task of finding shadows [28, 44, 50] with different, appropriate cues to define the discrepancy measure (Figure 4.8). Color cues, such as as the ratios of their components [4] have been used to remove shadows [63]. In addition it has been observed that some color spaces tend to have channels that are invariant to shadows [96]. This property lends itself to another shadow cue based on color. Of course, texture should be a cue since a shadow should not modify an objects texture, thus the texture would have to be very similar in two regions if one is to be considered a shadow-covered version of the other.

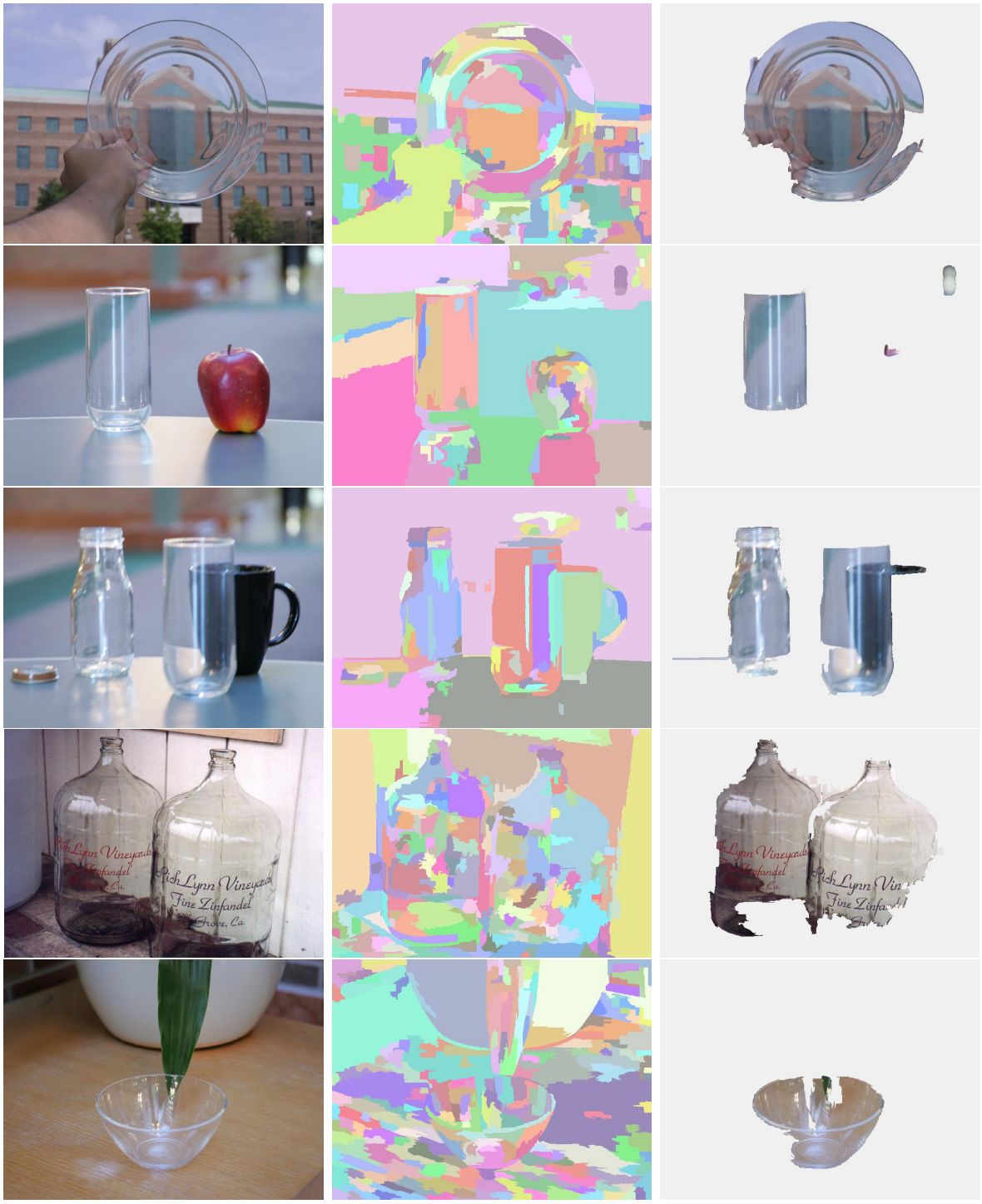

Figure 4.7: *Test images and output from the described method.* **Left:** *Five sample test images,* **Middle:** *The initial segmentation based on color and texture,* **Right:** *Regions labeled glass.*

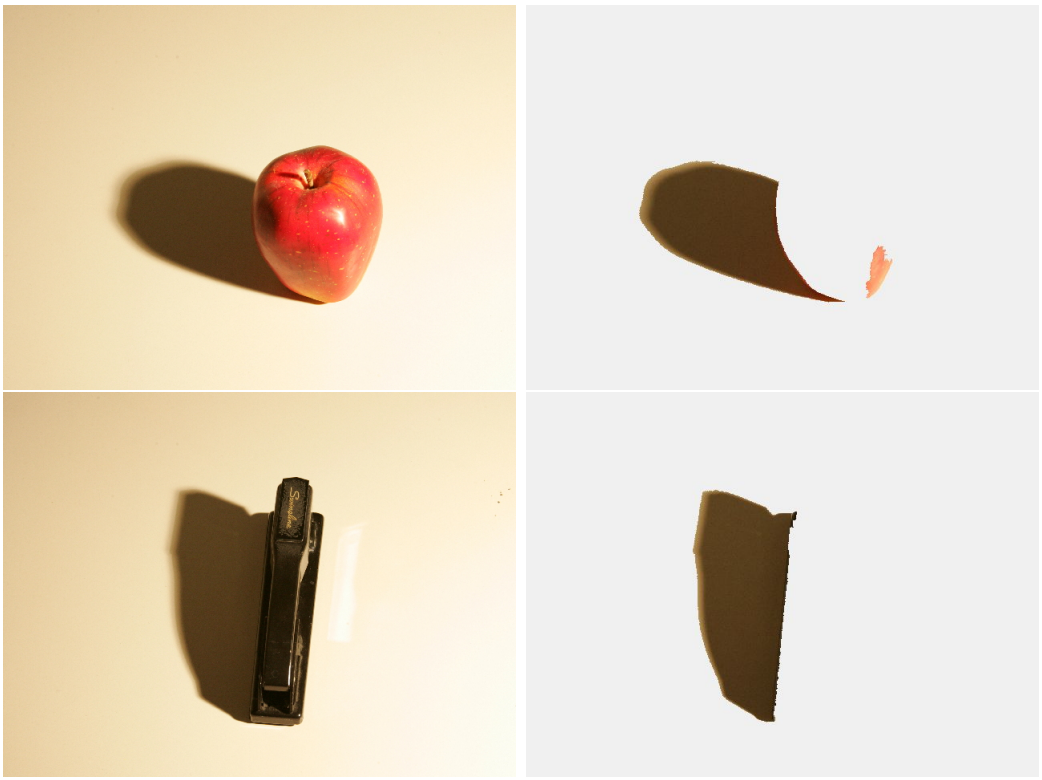

Figure 4.8: *A proof-of-concept illustration showing that the outlined technique can be generalized to find shadows with a discrepancy based on the appropriate cues. It is interesting to note that the stapler is actually darker than its shadow in the image.*

Chapter 5

Scene Segmentation and Estimating Segmentation Quality

Local information cannot capture all of the image/scene constraints available for image segmentation. Imposing global constraints such as context and shape priors are known to often improve segmentation results. In this chapter we consider two things. First we propose using global information, in the form of global features over a segmentation, to derive a means of estimating the quality of that segmentation. Second we attempt to use these quality estimates to improve an initial segmentation, in effect imposing learned global constraints. Given images that are initially over-segmented into regions of nearly uniform color and texture we use a set of global features on these regions and their class assignments to learn an energy function. This energy-based model is trained so as to assign lower energies to segmentations that have a larger percentage of correctly labeled pixels. The resulting energy function is then used to refine a given segmentation constructed from local features of the initial (over-segmented) regions. We demonstrate our approach with quantitative and qualitative results.

5.1 Characterizing Objects within a Scene

Hoiem et al. [40] have proposed using their scene segmentation method [39] as a guide to an external object classifier. Specifically they have used the segmentation of an outdoor

scene into ground/vertical/sky to determine a likely ground plane and camera position. This information is then used to guide a pedestrian or vehicle classifier by effectively limiting the position and scale of positives to ones consistent with viewing perspective. Though Hoiem et al. use this global information obtained from their segmentation to aid a separate classifier, this needs not be the case. For example in the case of indoor scene segmentation, knowledge of the ground plane would be useful in labeling regions such as chair, since chairs should be on the floor.

We attempt to use non-local information in the form of global features to estimate the quality of a given segmentation. Segmentations that are more correct (i.e. have more of their pixels labeled correctly) should match a series of global constraints specific to their particular scene type (in our case indoor scenes). Given this quality estimate we then attempt to improve a segmentation by choosing changes in class assignments that optimize the estimated quality. Segmenting indoor scenes is a challenging task since most objects present (chairs, desks, etc.) do not have very characteristic textures or colors. With regards to less local information such as context these scenes are also challenging as nearly all images contain all of the classes making class co-occurrences which are prevalent in datasets like the MSRC dataset of little use. Given a set of global features that attempt to capture various global scene properties we learn an energy-based model [57] such that better segmentations have lower energies. This energy function is then used to iteratively refine our initial segmentation. To our knowledge, our approach is unique in using an energy-based model trained with synthetically generated contrastive data to evaluate the quality of segmentations. Rather than working at the pixel level, we use regions from an initial over-segmentation of the image which we obtain from the graph based method of Felzenswalb et al. [29]. Regions obtained from an over-segmentation, also called super-pixels, have been shown to respect region boundaries [91, 65] and, unlike pixels, they provide us with some spatial support in order to calculate local features. The algorithm of Felzenswalb et al. contains within it a parameter k which affects the size of the resulting regions. A reasonable over-segmentation

of an image can be obtained by setting k to a low enough value.

In the following sections we describe the local features used to describe the regions obtained from our initial over-segmentation and the classifier used to assign labels to individual regions. We then describe our set of global features, our energy-based model and how it is used to improve a given segmentation. In addition we describe the problem of error amplification that occurs as one tries to iteratively refine an image and propose a means to minimize it's effects.

5.2 Local Model

5.2.1 Features

We wish to capture as much information about the region as possible in the type and number of features that we utilize. Some of the features will be more discriminant than others. However, limiting ourselves to a subset of the available features, perhaps in order to save computation time, would increase the possibility that two of the classes become indistinguishable. We instead use feature selection to determine the best features during the learning process. The features listed in Table 5.1 attempt to capture material properties, spatial properties, shape characteristics, and some geometric properties.

To model the materials making up the classes, we use color and texture. For color, we convert to the LAB color space so that Euclidean distance captures the similarity between colors and store the mean and standard deviation of each channel within each region. For texture, we use the MR filter bank of [117] which has the property of being rotationally invariant. Each image is convolved with a filter bank consisting of spot, edge, and bar filters at two scales ($\sigma = 0.5, 1$). The edge and bar filters are generated at 6 orientations between 0 and π. To make the result rotation invariant we only keep the maximum response of the 6 orientations. Thus the result consists of 2 spot filter responses and 2 edge/bar filter responses at 2 scales for a total of 6 values. Rotationally invariant texture features are important when

Local Feature Descriptions	#
Color	**6**
LAB values: mean and std	6
Texture	**6**
DOOG filters: mean response	6
Location	**10**
Mean x and y	2
Min x and y	2
Max x and y	2
# of points on top border	1
# of points on bottom border	1
# of points on left border	1
# of points on right border	1
Shape	**4**
2nd moment matrix	3
Size	1
Geometry	**36**
Longest Contour Line: mean x and y	2
Longest Contour Line: orientation	1
Longest Contour Line: length	1
Long Lines: total number in region	1
Long Lines: weighted total in region	1
Long lines: % of nearly parrallel pairs	1
Line Intersections: mean intersection	2
Line Intersections: histogram over 8 orientations, entropy (near center)	9
Line Intersections: histogram over 8 orientations, entropy (far from center)	9
Line Intersections: histogram over 8 orientations, entropy (very far from center)	9

Table 5.1: *The features used to represent each region. The "#" column indicates the dimensionality of the feature.*

the goal is to distinguish materials. Consider two similar pieces of wood, one rotated so that the grain is vertical and the other rotated so that the grain is horizontal. The important texture features is the grain, not the grain's orientation.

Some classes can take advantage of discriminative spatial information. For example ceiling regions tend to be at the top of the image and floor regions at the bottom. This information is obtained by storing the mean x and y values of the pixels within each region. Since the mean position depends on the size and extent of the region we also store the minimum and maximum of the x and y values. Both doors and walls tend to occupy the top of many images but wall regions tend to cover more of the top pixels. Thus we also store the number of pixels each region has at the top, bottom, left and right borders of the image.

Many classes have discriminative shape information. For example door regions tend to be fairly large, due to being of a uniform material, and are often elongated in the vertical direction. To capture this shape information we use the 2nd moment matrix of the pixel positions within each region and store the major and minor axis length as well as the orientation of the corresponding ellipse. In addition, region sizes (i.e., the number of pixels within the region) are stored.

Most of our classes are planar or made up of several planar parts. The intersection of two planes in 3D will project onto a line in the image plane. We attempt to identify regions that are likely to correspond to planes by finding the longest line along their contour. From such a line we store the mean x and y values along with its orientation and length. Knowing the orientation of these planes in 3D would be extremely helpful. Orientation information can be directly estimated from homogeneous textures [66, 24], however we attempt to indirectly capture this information as was done in [39] by looking at long near-parallel lines within each region and observing their intersections to get an idea of the planar surfaces vanishing line [32, 52].

$\underline{\text{AdaBoost.L}(X, Y):}$

for $t \leftarrow 1$ to T

- $w_i^t = \frac{1}{1+\exp\left(Y_i \sum_{s=1}^{t-1} h^s(X_i)\right)}$
- Train h^t to minimize $\sum_{i=1}^{n} w_t(i) e^{-y_i h_t(x_i)}$.

return final classifier: $f(x) = \sum_{t=1}^{T} h^t(x)$

Figure 5.1: *Logistic AdaBoost.*

5.2.2 Probabilistic Model

Regions from our training images are represented by their 62-dimensional feature vectors x_i. In this chapter, we use the logistic regression version of Adaboost ("Adaboost.L", see [18, 103]) to learn the posterior probability $P(y|x)$ that some observed feature vector x belongs to some class y. One of these boosted classifiers is constructed for each of our classes. In each case training is done in a one versus rest manner.

An overview of this algorithm is given in Fig. 5.1 for the case of binary classification $y \in \{-1, +1\}$. At each iteration t, a weak classifier h^t is trained on the examples x_i and labels y_i with importance given by weights w. Weights are recomputed so that examples with low confidence receive higher weights in the next iteration. The final model is $P(y|x) = f(x)$, where $f(x) = \sum_{t=1}^{t} h_t(x)$. We learn such a model for each class, and denote the probabilistic classifier for class i by P_i.

The next question is how to choose the base classifiers h. We have experimented with both SVMs and decision trees [39], and the latter have empirically given better results achieving higher accuracies. Let us consider a simple two-level decision tree (or *stump*) which makes a decision based on a threshold on one feature. A classification hypothesis $h(x_i)$ is produced by selecting a feature j and a threshold τ assigning all examples such that $x_i(j) < \tau$ to one class and all examples such that $x_i(j) \geq \tau$ to the other class. Concretely, let $I_+ = \{i \in 1..n | x_i(j) < \tau\}$ and $I_- = \{i \in 1..n | x_i(j) \geq \tau\}$, the feature and threshold are chosen so as to maximize the following classification score, which measures how well the two

classes are separated, using as before the weights w_i to measure the relative importance of the examples [41]:

$$\frac{1}{\sum_{i=1}^{n} w_i} \max \left(\sum_{i \in I_+} w_i \delta(y_i, +1), \sum_{i \in I_-} w_i \delta(y_1, -1) \right),$$

where δ is the Kronecker symbol defined by $\delta(x, y) = 1$ if $x = y$ and $\delta(x, y) = 0$ otherwise. In the case of boolean attributes this score provides the same ordering of chosen features as Quinlan's original information gain [88] while being considerably simpler.

We assign to each leaf node a probability that the corresponding examples are correctly classified:

$$\begin{cases} P_+ = \frac{1}{\sum_{i \in I_+} w_i} \sum_{i \in I_+} w_i \delta(y_i, +1), \\ P_- = \frac{1}{\sum_{i \in I_-} w_i} \sum_{i \in I_-} w_i \delta(y_i, -1). \end{cases}$$

If the decision tree classifier were perfect, one of these probabilities would be 1, and the other would be 0. In general this is not the case, and we use the sign of $\log(P_+/P_-)$ (which is $+1$ when $P_+ > P_i$ and -1 otherwise) as the output of the classifier. The same principles are easily extended to larger trees in order to learn non-linear relations among the features.

5.2.3 Segmentation

We use the *coupled geodesic active region* (CGAR) approach of Paragios et al. [85] to construct an initial image segmentation from the probabilities provided in the above local model. The CGAR attempts to label each pixel so as to maximize the probability of each class assignment while at the same time maximizing boundary smoothness. Briefly, we evaluate the quality of an image segmentation L such that $L(x, y)$ indicates the class label

at the point (x, y) using the following energy function:

$$E(L) = \alpha \sum_{i=0}^{N} \iint_{x,y:L(x,y)=i} g(P_{R_i}(x,y))dxdy + (1-\alpha) \sum_{i=1}^{N} \int_{s=0}^{1} g(P_{B_i}(\mathcal{C}_i(s)))|\mathcal{C}_i'(s)|ds$$

where $P_{R_i}(x, y)$ is the local probability of region i (i.e., class i) at the given point, $P_{B_i}(x, y)$ is the probability of a border for region i at the given contour point $\mathcal{C}_i(s)$, g is a Gaussian function, N is our number of classes, and α regulates the relative importance of the region and boundary forces. Here $P_{R_i}(x, y)$ is taken from our trained probabilistic classifier P_i and P_{B_i} is derived from this as described in [85]. Applying the Euler-Lagrange equation to the above energy function and embedding the result into a level set function [16, 121] we get:

$$\begin{aligned}
\frac{\partial}{\partial_t}\phi_i(\vec{\mathbf{x}}) &= \beta \sum_{j=1}^{N} H_i(j, \phi_j(\vec{\mathbf{x}}))|\nabla\phi_i(\vec{\mathbf{x}})| + \\
&\quad \gamma(g(P_{R_i}(\vec{\mathbf{x}})) - g(P_{K_i}(\vec{\mathbf{x}}))|\nabla\phi_i(\vec{\mathbf{x}})| + \\
&\quad \delta(g(P_{B_i}(\vec{\mathbf{x}}))\mathcal{K}_i(\vec{\mathbf{x}})|\nabla\phi_i(\vec{\mathbf{x}})| + \\
&\quad \nabla g(P_{B_i}(\vec{\mathbf{x}})) \cdot \frac{\nabla\phi_i(\vec{\mathbf{x}})}{|\nabla\phi_i(\vec{\mathbf{x}})|})|\nabla\phi_i(\vec{\mathbf{x}})|
\end{aligned}$$

where $\vec{\mathbf{x}}$ is a point (x, y), $\mathcal{K}$ is the Euclidean curvature with respect to the curve, $\vec{\mathcal{N}}$ is the curves normal, and $\phi_i(\vec{\mathbf{x}})$ is class i's level set function. $P_{K_i}(\vec{\mathbf{x}})$ is the probability of some class other than i being present at the given point. The coupling force $H_i(j, \phi_j(\vec{\mathbf{x}}))$ is taken to be

$$\begin{cases}
0 & \text{if } j = i, \\
H_a(\phi_j(\vec{\mathbf{x}})) & \text{if } j \neq i \text{ and } \phi_j(\vec{\mathbf{x}}) \leq 0, \\
\frac{1}{N-1}H_a(\phi_j(\vec{\mathbf{x}})) & \text{if } j \neq i \text{ and } \phi_k(\vec{\mathbf{x}}) > 0 \text{ for all } k \neq i
\end{cases}$$

where:

$$H_a(x) = \begin{cases}
+1, & x > a \\
-1, & x < -a \\
\frac{1}{\tan(1)}\tan(x/a), & |x| \leq a
\end{cases}$$

Global Feature Descriptions	#
Compactness	**1**
Number of connected groups	1
Location	**43**
Mean position (per class)	14
Min position (per class)	14
Max position (per class)	14
Mean P(class|y-coordinate)	1
Shape	**29**
% of image covered (per class)	7
Total boundary length internal to image	1
Length of boundary internal to image (per class)	7
Length of boundary on image boundary (per class)	7
Total boundary length to area ratio (per class)	7
Neighborhood Statistics	**56**
% of each other class nearby (per class)	56
Confidence	**1**
Sum of P(class|region)	1

Table 5.2: *The features used to represent a correctly labeled image.*

This coupling force ensures that every pixel is given a unique class label. The parameters β, γ, and δ govern the relative importance of the coupling, region and boundary forces respectively. Using the above update equation we can use gradient descent to evolve the contour until it converges to a local optimum. Since class labels are assigned to pixels we need to add a step to retrieve labels for our regions. Since the original over-segmentation tends to be reliable this also tends to improve things in the case where the curve evolution incorrectly eats away at a region. To do this we simply have each active region vote once for every pixel it occupies within a region. The class of the active region with the most votes is then assigned to the region.

5.3 Global Model

5.3.1 Features

The previous sections only considered local information in terms of the features within individual regions. However there is useful information that can not be captured locally. For example, floors are usually at the bottom of an image and ceilings at the top. If we see a large group of regions labeled floor at the bottom, a large group of regions labeled ceiling at the top, and one region labeled floor among the ceiling regions, then we should expect that this one region may be wrongly labeled. We capture this information with global features of the image labels (Table 5.2).

A correctly segmented image should be compact in the sense that there are a handful of connected groups of similarly labeled regions. For example, an image that has had its regions randomly assigned labels would not likely be compact since there would likely be small groups of floor scattered throughout the image. We capture this idea of compactness by counting the number of connected groups.

As in the local case, position is a useful global cue. This time, however, we are concerned with the set of regions assigned to a particular class. We again capture position information by storing the mean, standard deviation, min, and max of the x and y coordinates of each class. As an example consider the class floor, which one expects to be at the bottom of an image. If in a given image most of what is labeled floor is at the bottom except for one or two regions which are labeled near the top, the mean position of the class would be shifted upward. A more correct segmentation would not have these out of place regions and would have a lower mean position. In this case the lowering of the mean position is helpful in distinguishing one segmentation as being better than another.

Size is a useful feature as well. In most indoor images we would expect that most of what we see be either floor or wall. To capture this we store the percentage of the image covered by each class. Shape, though a usual cue, would be too complex too directly use as a feature

within our setup. We attempt to indirectly capture characteristics of shape by looking at the length of the boundary of each class and also looking at the ratio of the boundary versus the area. Consider the difference between tables and chairs. Tables are larger objects with smooth boundaries and thus should have a small boundary as compared to its area. Chairs on the other hand have sharp bends at the seat, arms, and legs. Combined with their smaller area, chairs should have a higher boundary to area ratio.

Next, we include neighborhood statistics. An example of this is that chairs tend to be near tables. Another could be that chairs usually are not surrounded by wall, which might mean that the chair is flying. To capture this for each class we count the percentage of each other class along its borders. We include image boundary as a class here, giving us 8 numbers for each of the 7 classes.

Lastly, to prevent the learned global model from completely overriding the local model we add the following local confidence term:

$$\sum_i n_i P(y_i|x_i)$$

where x_i is the local feature vector for region i, y_i is the class currently assigned, n_i is the number of pixels within the region, and P is the local probability of region i being class y_i.

5.3.2 Energy-Based Model

We attempt to learn an energy function [57] such that better segmentations will have lower energies than worse ones. Learning to assign an absolute energy to each segmentation is difficult so we use the idea of contrastive divergence [37] to instead learn locally within the feature space which segmentations are better and should have lower energies than nearby worse segmentations. We begin with an energy function of the form:

$$E(w, z_i) = w^T z_i$$

where z_i is a global feature vector for image i and w is the vector of parameters for the energy function. We choose a linear function for it's simplicity; however more complex functions can be used (Appendix B). Within our training data we have a set of feature vectors z_i, which we wish to have low energy values, as they represent good segmentations (perhaps the ground truth or a slightly noisy version of it). Relative to these we have a set of contrastive examples z_i' that are nearby, in the sense that they come from segmentations that are for the most part the same as those of z_i except that some of the label assignments have been changed to incorrect values. These contrastive examples should have higher energies than the good examples. Our goal is to set w so as to maximize the number of contrastive examples with energies higher than nearby positive examples.

We can set w by minimizing the negative log-likelihood loss described in LeCun et al. [57]:

$$L_{nll}(w, z_i) = E(w, z_i) + \log \left(\int_{z' \in z_i'} e^{-E(w,z')} \right)$$

The above loss function is derived by maximizing the probability of the good examples, given in the form of a Gibbs distribution, over their relative contrastive examples. This is done by minimizing the negative log-likelihood, which is why it is referred to as the negative log-likelihood loss function. Differentiating it with respect to the parameters, w, yields the following update equation [57, 112]:

$$w \leftarrow w - \eta \left(\frac{\partial E(w, z_i)}{\partial w} - \sum_{z' \in z_i'} \frac{\partial E(w, z')}{\partial w} P(z', w) \right)$$

Using gradient descent we find a w which is a local minima of the loss. The learning rate, η, is determined by a line search at each iteration of the gradient descent. Within contrastive divergence learning, $P(z', w)$ is usually estimated via MCMC. This estimation requires that we generate new contrastive examples at each iteration. For our setup this is impractical as it would require a somewhat costly process over the training set (Section 5.3.2). Thus, we

instead estimate $P(z', w)$ through the Gibbs distribution:

$$P(z', w) = \frac{e^{-E(w,z')}}{Z}$$

The normalizing constant Z is difficult to calculate in practice so we instead estimate it as the sum of $e^{-E(w,z')}$ over our training examples.

Training data

The training data is generated in the following manner. Let z be the feature vector representing a ground truth segmentation s of our training set. We now select a random number of regions (super-pixels) in the image and change each of their classes to one of the other 6 possibilities, yielding a new segmentation s' with feature vector z'. The segmentation s is an improvement over s' since one or more regions of s' are mislabeled (this will be our contrastive example). We can generate our positive and contrastive training data in this manner by randomly changing regions to other classes. In addition to using ground truth segmentations as the basis for improved segmentations we also use versions of the ground truth with noise added. Noise is added by randomly selecting a number of regions and changing their class assignments. When the worse segmentation is generated for the contrastive example we make sure not to change any of these mislabeled regions back to the correct class. The idea behind this is that we will likely encounter these less than perfect segmentations.

5.3.3 Segmentation Improvement

Given an initial scene segmentation we attempt to improve it by enforcing the learned global constraints captured within the energy-based model above. We adopt a simple greedy method to refine the segmentation. For a given iteration we change each region in turn to each of the 7 classes and record the resulting energy. We then choose the region/class change with the lowest energy and repeat the process. We have also experimented with MCMC and

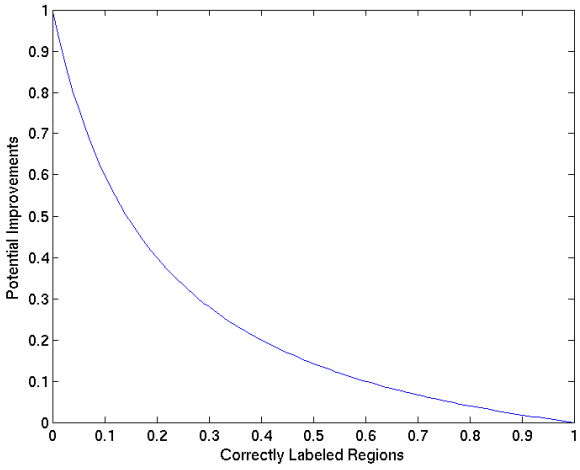

Figure 5.2: *Possible improvements vs. percentage of regions correctly labeled.*

a tree based search, choosing some n of the top changes at each iteration, however the greedy approach has performed the best.

Error Amplification

Initial experiments had shown an unexpected side effect of iteratively improving a segmentation region by region. We have observed that the amount of the improvement obtained appeared much smaller for segmentations that were mostly correct. In fact it can be shown that it becomes harder and harder to iteratively choose regions to change and improve a segmentation as more of the regions are labeled correctly.

Consider a local segmentation of an image containing n regions. Let m be the percent of regions correctly labeled (with $m < 1$) and c be the number of classes. The number of potential improvements (i.e. good changes) is $n(1-m)$ since there is only one good change for each incorrectly labeled region (i.e. the correct one). The potential degradations on the other hand have $nm(c-1)$ possibilities, since every correctly labeled region can be changed to every non-correct class for a degradation. From this we can write the percentage of potential improvements in the total number of changes, p, as a function of the percent of the regions

already correctly labeled:

$$p = \frac{n(1-m)}{n(1-m) + nm(c-1)} \tag{5.1}$$

Canceling the n and setting c to 7 (the number of classes we consider) we plot p vs. m (Figure 5.2).

From Figure 5.2 we see that as the segmentation improves the percentage of potential improvements drops rapidly. This does not yet take into account the actual model/classifier used to determine the improvements. If the model were perfect, correctly labeling all degradations and improvements, then the increased number of degradations produced with better segmentations would not matter. Realistically this is not the case as we will not have a perfect model and mistakes will be made. In Appendix C we take a closer look at this problem and consider how fewer potential improvements increases the number of false positives we see. For the rest of this chapter however we will simply attempt to minimize it's effect.

Quality Prediction

One way to deal with the above situation is to focus on the worst segmentations. Using the same global features we train a boosted tree classifier. The training data again consists of randomly damaged ground truth segmentations. However, this time we make a positive dataset with those segmentations that have greater than or equal to some percentage of the image labeled correctly, in our experiments 80%. The negative dataset will consist of all other segmentations. The local segmentations from test images are first passed through this classifier. If classified as having a classification rate less than the threshold value we attempt to improve the segmentation using the EBM described in the previous section.

5.4 Experiments

Our data consists of 105 annotated indoor images, kindly provided by Toyota Motor Corporation. The images were taken in locations which varied in appearance (styles and furniture),

under various lighting conditions and arbitrary viewpoints. In our experiments, we randomly generate 10 splits of this data, each with 75% used as training data and the remaining 25% as test data.

For the local classifiers, we run the logistic version of Adaboost for 200 iterations, using three-level decision trees as weak classifiers. A validation set is used to verify that the model is not over-fitting. We biased the positive data so that classifying them correctly was 3 times as important as the negative data. In addition, the importance of classifying each region correctly is biased by the region's size. For the global energy model we train with 20 positive examples per training image, one the ground truth and 19 noisy versions of the ground truth. Each of these in turn has 200 contrastive examples. This training data is generated once at the beginning of training.

We begin by testing the effect of the error amplification. Rather than starting with the segmentation provided by the local classifiers and the CGARs we randomly damage the ground truth segmentations of the test images in the manner specified for the EBM training data. The amount of damage is varied to create three categories: one with low, one with medium, and one with a high amount of damage. The low category will have relatively fewer incorrectly labeled regions as compared to medium and high and so on. We then apply the greedy global refinement discussed in Section 5.3.3 to these initial segmentations. We do not consider the local term in these experiments, so none of the observed improvements

Class	**Color**
ceiling	light blue
chair	light green
desk	yellow
door	dark green
floor	dark blue
wall	orange
other	black

Table 5.3: *The indoor scene classes considered and the corresponding colors used to render them in segmentation experiments.*

	High	Medium	Low
Initial	62.11%	76.76%	89.38%
Final	75.46% (13.35%)	83.73% (6.97%)	89.10% (-0.28%)

Table 5.4: *Initial and post global refinement classification rates for test images with damaged initial segmentations. High, medium and low indicate the amount of damage imposed to the ground truth segmentations. The values in the "initial" row indicates the resulting average percentage of correctly labeled pixels of these damaged segmentations. The values in the "final" row indicate the average percentage of correctly labeled pixels after the global refinement. The value in parenthesis indicates the improvement from the initial value.*

are a result of simply trying to maximize the local classification results. Also, we have observed that the greedy refinement is only reliable when the change in energy is large, thus we stop making changes when the change in energy becomes small. The results are shown in Table 5.4. What should be noted from this synthetic test is the amount of improvement seen in each case. In the low category, with little initially incorrect, the segmentation actually gets worse. On the other hand the other two categories show considerable improvement.

For the rest of our experiments we start with an initial segmentation produced by the local classifiers and 1000 iterations of the CGARs. These segmentations are greedily refined using our global energy model. We experiment with and without the local confidence term in the EBM. The weight assigned to the local term regulates the relative importance of the local and global features (shown in Figure 5.3 for one of the dataset splits). If the magnitude of this weight is high then only local information is considered and the improvement will be 0 since nothing will change from the initial segmentation. If the magnitude is too low then the global information completely overrides the local information which sometimes results in worse segmentations. When set to a more appropriate value, chosen by the training of the EBM, the weight of the local term allows improvements from the global information without completely ignoring the initial local information.

Improvement results obtained by starting from these local segmentations are shown in Table 5.5. Overall, the CGARs labeled 76.55% of the pixels correctly. With the use of the global refinement, we were able to increase the percent of correctly labeled pixels by

Method	Classification Rate	Classification Rate after Quality Prediction
Initial	76.55%	68.83%
Final	78.12% (1.66%)	72.08% (3.25%)
Final (no local term)	78.21% (1.31%)	72.28% (3.45%)

Table 5.5: *Classification rates for the various methods on the Toyota data set: local features only, global features + local term, global features only (initialized by local segmentation). Middle column gives results averaged over all test images. Right column gives results on test images classified as being initially worse than* 80% *correct via the quality prediction discussed in Section 5.3.3. Values in parentheses are the average improvements over the local segmentations. Note, the quality prediction step labeled* 30.3% *of the test images as being less than* 80% *correct (an accuracy of* 69.1%*).*

an average of 1.66%. Like Shotton et al. [106] who use context to improve segmentation results, we see a seemingly small numerical improvement overall. While in their case small improvements appear to be distributed over all their test images, our case involves large improvements that occur on the worst initial segmentations (as much as 20%, see Figure 5.5). We also note that image context alone, a strong cue within the MSRC dataset used in [106], would likely not lead to improvements on the Toyota dataset as every image contains nearly every class. If we use the quality prediction described in Section 5.3.3 and only attempt to improve those segmentations that were classified as being below 80% correct we achieve an average improvement of 3.25%. As an interesting side note, the last row of Table 5.5 uses global information alone without the local term (i.e. no image information) and achieves and average improvement of 1.31%.

Qualitative results are shown in Figures 5.5 and 5.6. In particular, consider the images in the first rows of each. In Figures 5.7 and 5.8, we show these two images, their over-segmentations, and the local probabilities assigned for each class. The results from the local segmentations can be seen in the third column of Figures 5.5 and 5.6. In Figure 5.8, the local probabilities are not too bad, and after the global refinement the overall recall only improves slightly, from 78.8% to 79.44% (fourth column). On the other hand, in Figure 5.7 large portions of the desk are assigned small probabilities of being desk and high probabilities as

Method	Classification Rate	Classification Rate after Quality Prediction
Initial	74.08%	70.68%
Final	75.64% (1.56%)	72.83% (2.15%)
Final (no local term)	75.64% (1.56%)	72.83% (2.15%)

Table 5.6: *Classification rates for the various methods on the UIUCi data set: local features only, global features + local term, global features only (initialized by local segmentation). Middle column gives results averaged over all test images. Right column gives results on test images classified as being initially worse than* 80% *correct via the quality prediction discussed in Section 5.3.3. Values in parentheses are the average improvements over the local segmentations. Note, the quality prediction step labeled* 34.5% *of the test images as being less than* 80% *correct (an accuracy of* 65.0%*).*

to being other. In this case the improvement from the global refinement is considerable, with the recall going from 63.2% to 83.13%. The global refinement, concerned with the quality of the overall labeling of the image, is able to recover most of the desk even though the local probability of being desk is low. Similarly in rows 2 through 7 of Figure 5.5, pieces of walls, floors, and chairs are recovered despite being misclassified by the local information.

5.5 Discussion

We have proposed a means of estimating the quality of a segmentation and using this estimate to attempt to improve it. This is performed via an energy-based model trained with global features so as to prefer segmentations that are more correct. In addition to the Toyota dataset discussed in this chapter we have conducted experiments on our own larger indoor dataset (Table 5.6), containing 232 images and the same 7 classes. Results are similar with an average improvement of 1.56% overall and 2.15% with quality prediction (where the quality prediction labeled the worst segmentations with an accuracy of 65.0%).

Our experiments have revealed an error amplification that is inherent to any iterative refinement method such as the greedy refinement described in Section 5.3.3. We deal with this by proposing a quality prediction step that allows us to focus our efforts on the lower quality segmentations which are in most need of improvement. Lastly, adding new global

features is simple and convenient within the EBM paradigm. Finding useful global features however is part of ongoing research.

5.6 Figures

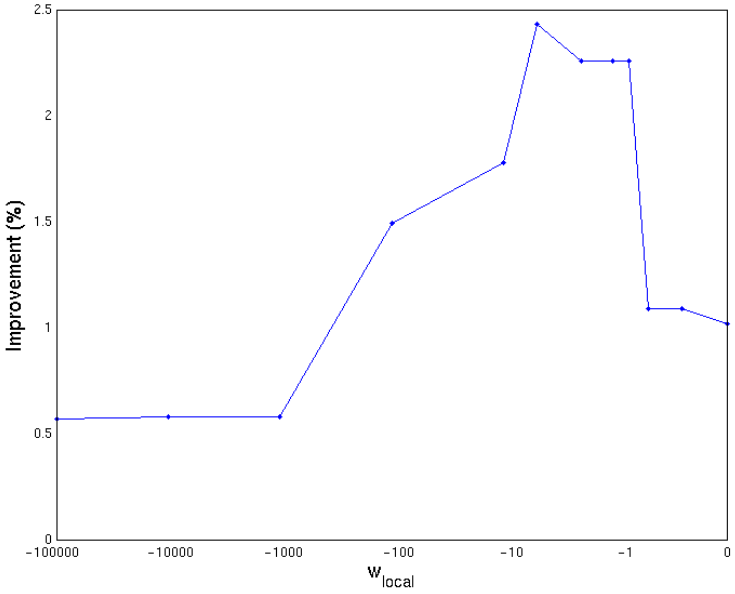

Figure 5.3: *The effect of the local term on the improvements obtained. The weight, w_{local}, assigned to the local term within the EBM is a negative value whose magnitude indicates how much trust should be given to the local classifier results. As the magnitude of the weight gets larger, left on the x-axis, the improvement becomes smaller as this tends to ignore the global information and effectively returns the original local segmentation. As the magnitude of the weight gets smaller, right on the x-axis towards 0, the improvement also gets smaller. In this case the global information completely overrides the local information which can result in worse segmentations. The trained EBM sets w_{local} to -1.051 which is near the peak on the plot.*

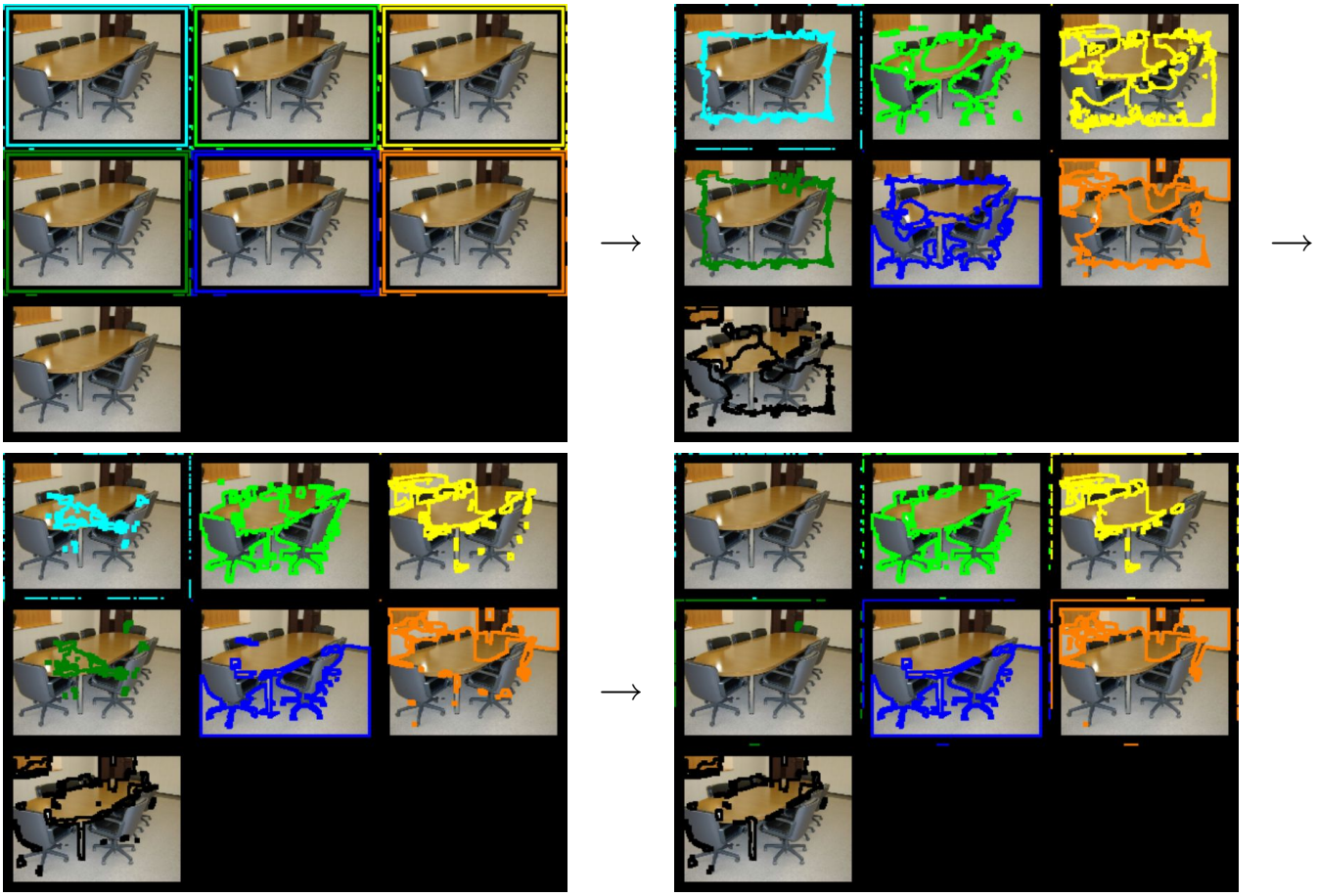

Figure 5.4: *Coupled geodesic active region evolution after 1000 iterations. Class colors are given in the Table 5.3.*

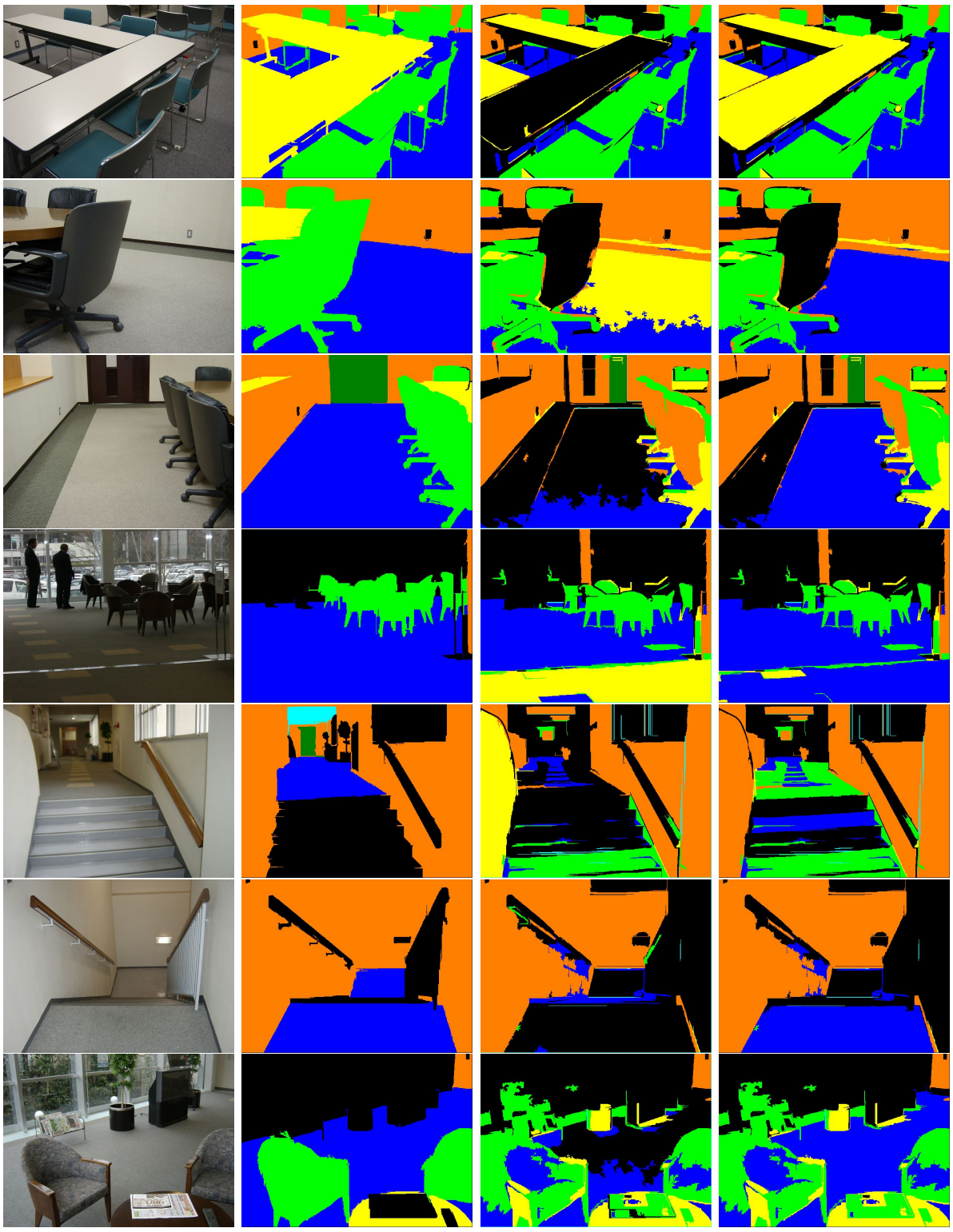

Figure 5.5: *From* **left** *to* **right:** *test image, ground truth, output from CGAR, output from CGAR + Global Refinement. Initial vs. final error rates (from* **top** *to* **bottom***: 63.2% to 83.13%, 52.4% to 74.02%, 40.1% to 62.3%, 69.6% to 88.32%, 60.5% to 72.34%, 55.3% to 75.3%, 58.5% to 73.62%.*

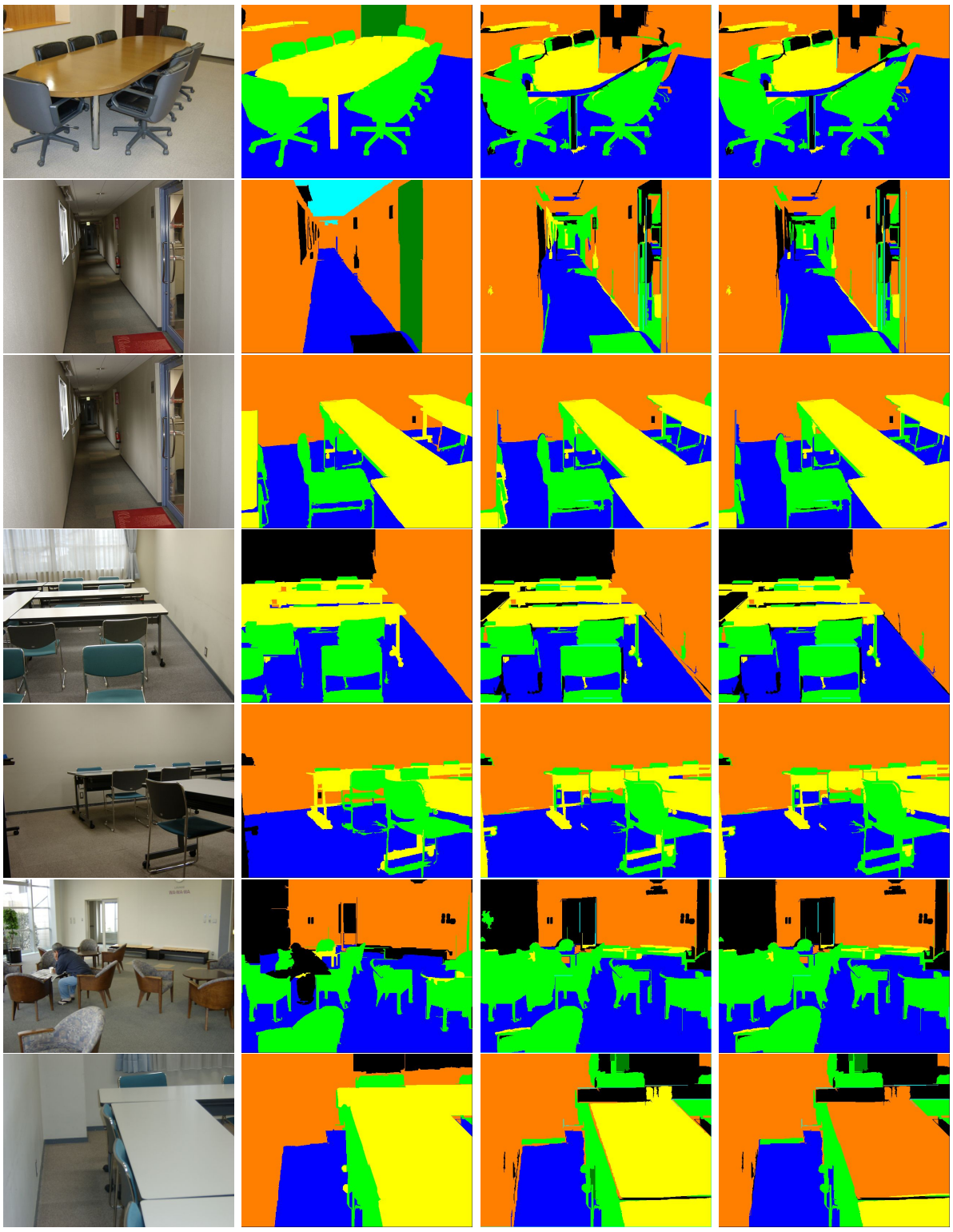

Figure 5.6: *From* **left** *to* **right:** *test image, ground truth, output from CGAR, output from CGAR + Global Refinement. Initial vs. final error rates (from* **top** *to* **bottom***): 78.8% to 79.44%, 74.5% to 74.13%, 91.8% to 92.16%, 94.6% to 94.11%, 91.5% to 91.17%, 80.2% to 80.85%, 86.1% to 62.4%.*

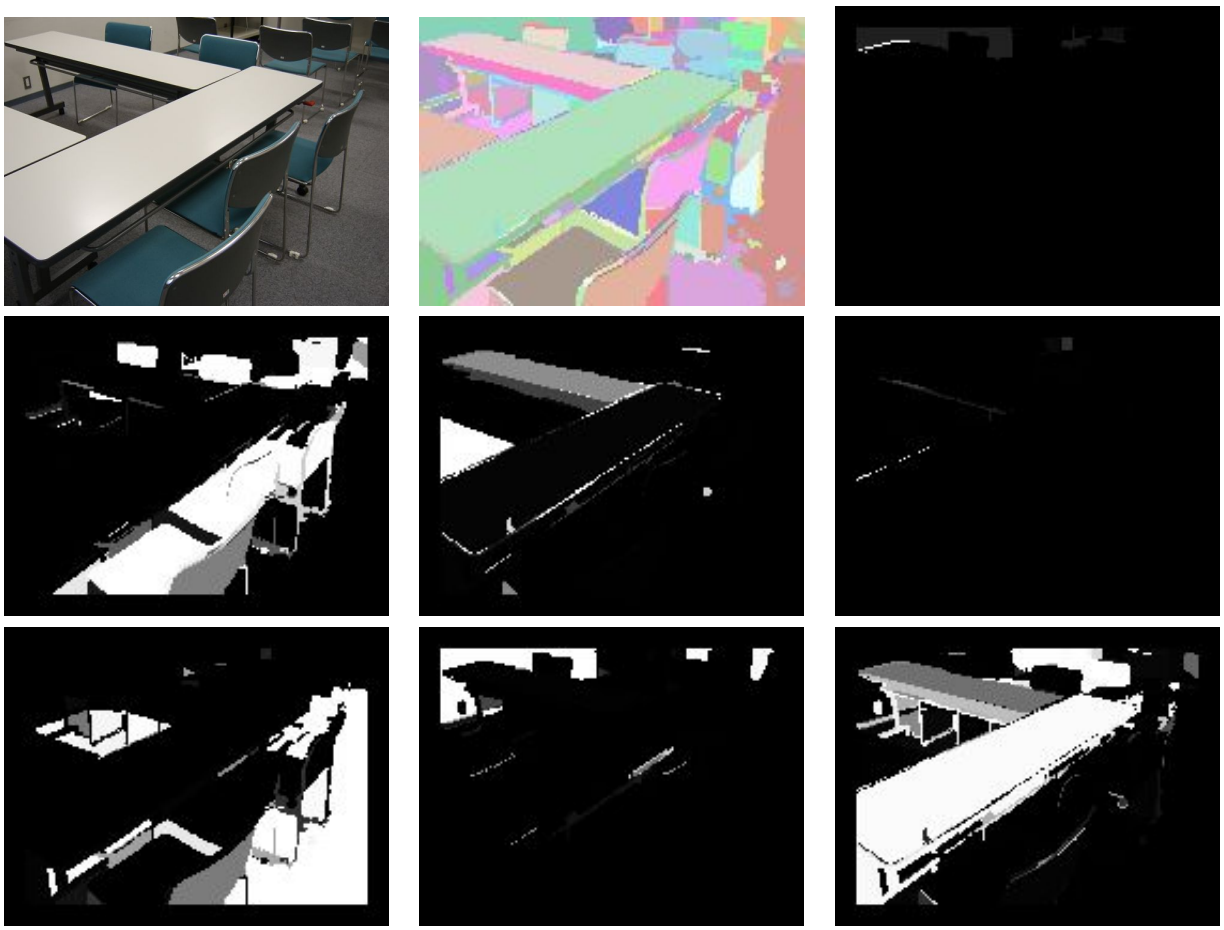

Figure 5.7: **Top left:** *a test image in which the local information is mis-leading,* **top-middle:** *regions,* **top-right:** *probability of each region being of class ceiling,* **middle-left:** *probability of being of class chair,* **middle-middle:** *probability of being of class desk,* **middle-right:** *probability of being of class door.* **bottom-left:** *probability of being of class floor,* **bottom-middle:** *probability of being of class wall,* **bottom-right:** *probability of being of class other. Brighter values indicate higher probabilities.*

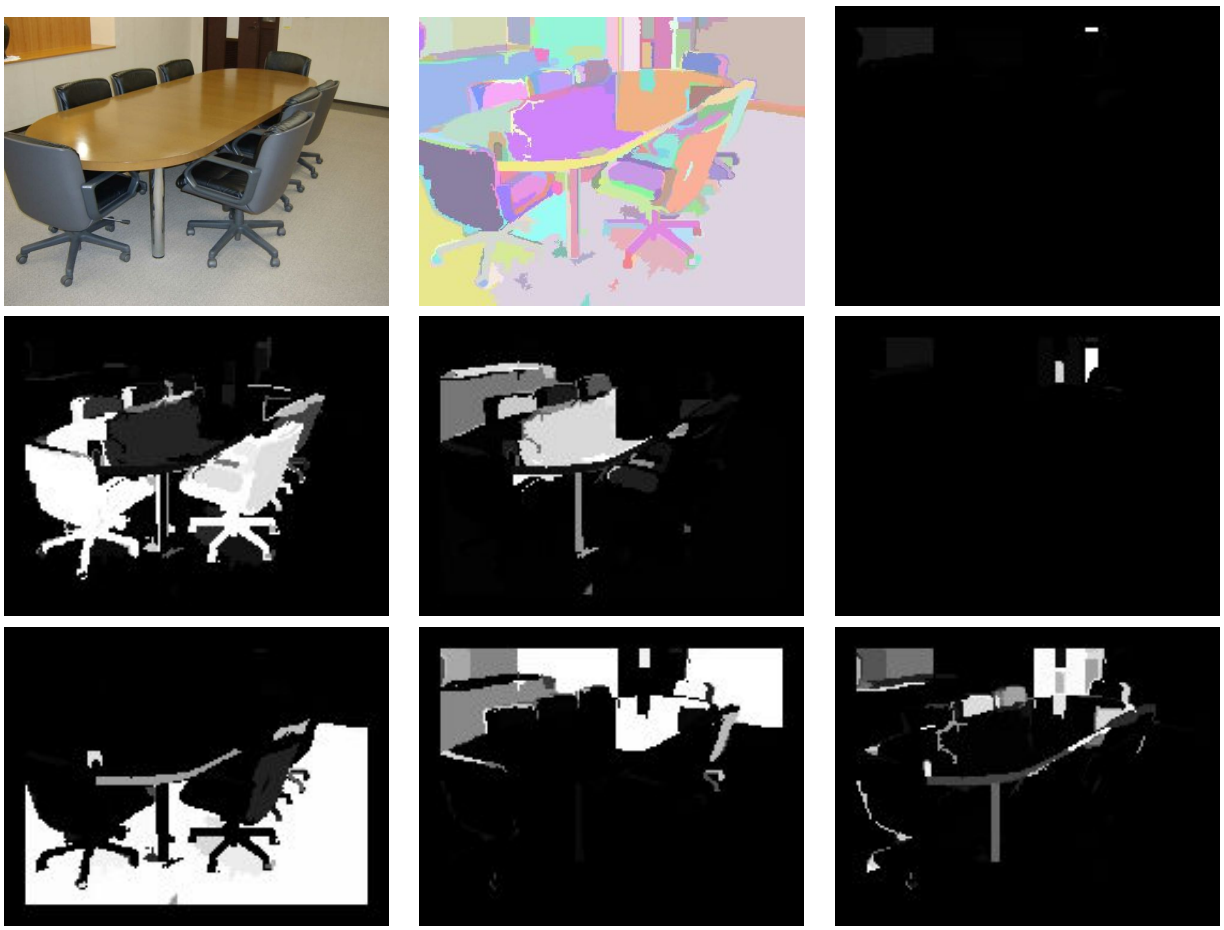

Figure 5.8: **Top left:** *a test image in which local information is fairly accurate,* **top-middle:** *regions,* **top-right:** *probability of each region being of class ceiling,* **middle-left:** *probability of being of class chair,* **middle-middle:** *probability of being of class desk,* **middle-right:** *probability of being of class door.* **bottom-left:** *probability of being of class floor,* **bottom-middle:** *probability of being of class wall,* **bottom-right:** *probability of being of class other. Brighter values indicate higher probabilities.*

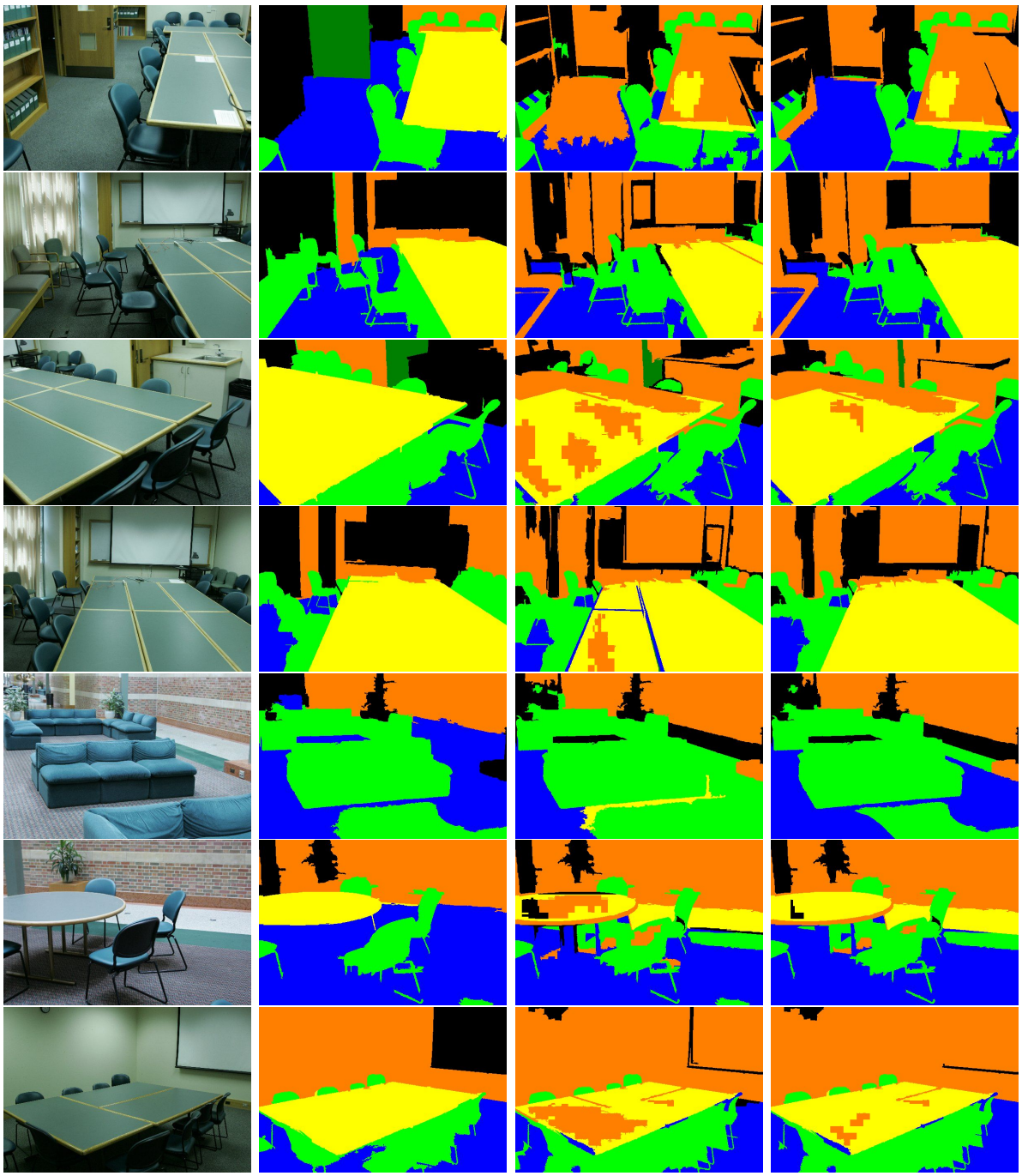

Figure 5.9: *From* **left** *to* **right:** *test image, ground truth, output from CGAR, output from CGAR + Global Refinement. Initial vs. final error rates (from* **top** *to* **bottom***): 47.5% to 58.88%, 60.2% to 63.24%, 63.3% to 67.54%, 69.1% to 72.24%, 77.9% to 87.05%, 80.0% to 84.64%, 76.3% to 82.13%.*

Chapter 6

Conclusion

In this chapter we summarize the contributions of our research and discuss potential directions for future work.

6.1 Summary

In this dissertation we examined the task of supervised image segmentation. In particular we have taken a look at various non-local features for the purpose of capturing information beyond local appearance. Initially we considered the problem of segmenting transparent materials. Unlike most opaque materials, transparent materials lack any locally distinctive information. While one might be tempted to use the highly specular nature of glass objects to identify them by their highlights this information is by no means indicative of transparency as many opaque materials also are highly specular [82]. Instead we focused on the nature of transparency itself, that being the ability to see through something. We argue that transparency is an inherently binary property as it implies perceiving two things: a background scene and something interfering/distorting that background scene. In order to identify transparency within images we proposed the use of binary cues which consider two regions simultaneously. These cues allow us to effectively ask the question "Is this region a glass covered version of this other region?" and detect the glass and background scene at the same time. In Chapter 3 we constructed a classifier around a set of such cues and use

this to label edges that appeared to be from a transparent glass object. In Chapter 4 we extend this method so that regions can be labeled as glass or non-glass.

In Chapter 5 we considered the task of improving a segmentation based on local features with scene constraints obtained from global features. Here our scenes contained objects made of opaque materials so local features were possible. However there is information that cannot be captured by local cues alone. For example, the observation that chairs are often seen near desks can not be captured by examining a single region. Rather than imposing some predetermined global constraint on our scene we allowed the system to learn these constraints by providing it with a set of global features that captured various global properties and allowing the system to decide which ones were important. From a training set we synthetically generate a large number of contrastive examples that are relatively worse in quality as compared to a given ground truth segmentation. These positive and contrastive examples are then used to train an energy based model so as to assign smaller energies to segmentations which are more accurate. This model, which provides an estimate of a segmentations quality, is then used to refine the initial segmentation.

6.2 Future Work

There are two potential directions for future work. Of particular interest is the arity of the features used. We have used binary features to identify transparency and global features to improve local segmentations. Higher arity features seem to be a promising direction in obtaining the non-local properties and habits of materials as described by Adelson et al. (e.g. luster, transparency, resinous and columnar nature, etc...). Secondly the types and number of features used in this work is flexible. We have demonstrated how our method in Chapter 4 can be altered to identify shadows given appropriate binary features. Mirrored surfaces may also be recognizable through binary features, as we would be looking for a reflection distorted version of some other part of the scene. Finding useful (and efficiently

computed) features is an important area of research.

Appendix A

Active Contours

In this appendix we review snakes, geodesic active contours, and supervised geodesic active regions.

A.1 Snakes and Geodesic Active Contours

The active contour method known as snakes [48] is performed by minimizing two energies:

$$E(\mathcal{C}(q)) = \int_0^1 \alpha|\mathcal{C}'(q)|^2 + \beta|\mathcal{C}''(q)|^2 dq - \lambda \int_0^1 |\nabla I(\mathcal{C}(q))| dq \tag{A.1}$$

where $E(\mathcal{C}(q))$ is the total energy of the curve $\mathcal{C}(q)$. I is an image and $I\nabla(\mathcal{C}(q))$ is the intensity gradient/edge strength along the curve $\mathcal{C}(q)$. From here on we will write $\mathcal{C}(q)$ as $\mathcal{C}$. α, β, and λ are constants which govern the importance of the various terms. The first two terms, called the internal energy, enforces that the curve is C1 and C2 continuous respectively (i.e. smooth). The last term, called the external energy, draws the curve near strong edges. Enforcing C2 continuity is often found to be unnecessary and thus dropped:

$$E(\mathcal{C}) = \int_0^1 \alpha|\mathcal{C}'|^2 dq - \lambda \int_0^1 |\nabla I(\mathcal{C})| dq$$

We can rewrite this using a function g which has the properties that $g(x) \to 0$ as $x \to \infty$.

$$E(\mathcal{C}) = \alpha \int_0^1 |\mathcal{C}'|^2 dq + \lambda \int_0^1 g(|\nabla I(\mathcal{C})|)^2 dq$$

This expression resembles the Lagrangian [15]:

$$\mathcal{L}(\mathcal{C}) = \mathcal{T}(\mathcal{C}) - \mathcal{U}(\mathcal{C})$$

where $\mathcal{T}(\mathcal{C})$ and $\mathcal{U}(\mathcal{C})$ are the kinetic and potential energies respectively of a curve at a given position. Making use of this observation we can rewrite our energy equation as:

$$E(\mathcal{C}) = \int_0^1 \mathcal{L}(\mathcal{C}) dq$$

where

$$\begin{aligned} \mathcal{T}(\mathcal{C}) &= \alpha |\mathcal{C}'|^2 \\ \mathcal{U}(\mathcal{C}) &= -\lambda g(|\nabla I(\mathcal{C})|)^2 \end{aligned}$$

Letting the momentum be $p = 2\alpha\mathcal{C}'$ we can write the Hamiltonian as:

$$\begin{aligned} H &= \mathcal{T}(\mathcal{C}) + \mathcal{U}(\mathcal{C}) \\ &= \frac{p^2}{4\alpha} + \mathcal{U}(\mathcal{C}) \end{aligned}$$

Maupertuis' principle says that curves $\mathcal{C}(q)$ in Euclidean space which are extremal corresponding to the Hamiltonian $H = \frac{p^2}{4\alpha} + \mathcal{U}(\mathcal{C})$, and have a fixed energy level E_0 (law of conservation of energy), are geodesics, with non-natural parameter, with respect to the new

Riemannian metric:

$$g_{ij} = 4\alpha(E_0 - \mathcal{U}(\mathcal{C}))\delta_{ij}$$

where δ_{ij} is the Kronecker delta and $i, j = 1, 2$ represent the components of the curve (i.e. x and y in 2D Euclidean space). Taking E_0 to be 0 we are left with:

$$\begin{aligned} g_{ij} &= -4\alpha\mathcal{U}(\mathcal{C})\delta_{ij} \\ g_{ij} &= 4\alpha\lambda g(|\nabla I(\mathcal{C})|)^2\delta_{ij} \end{aligned}$$

Note that in Euclidean space $g_{ij} = \delta_{ij}$. In our Riemannian space minimizing $E(\mathcal{C})$ is equivalent to minimizing:

$$\begin{aligned} \int_0^1 \sqrt{g_{ij}\mathcal{C}_i'\mathcal{C}_j'}dq &\Rightarrow \int_0^1 \sqrt{g_{11}\mathcal{C}_1'^2 + g_{12}\mathcal{C}_1'\mathcal{C}_2' + g_{22}\mathcal{C}_2'^2}dq \\ &\Rightarrow \int_0^1 \sqrt{g_{11}\mathcal{C}_1'^2 + g_{22}\mathcal{C}_2'^2}dq \\ &\Rightarrow \int_0^1 \sqrt{(4\alpha\lambda g(|\nabla I(\mathcal{C})|)^2)(\mathcal{C}_1'^2 + \mathcal{C}_2'^2)}dq \\ &\Rightarrow \int_0^1 \sqrt{4\alpha\lambda}g(|\nabla I(\mathcal{C})|)|\mathcal{C}'|dq \end{aligned}$$

Without loss of generality we can set $4\alpha\lambda = 1$ to get:

$$L_R = \int_0^1 g(|\nabla I(\mathcal{C})|)|\mathcal{C}'|dq$$

The above expression gives us the length, L_R, of the curve in a Riemannian space derived from features in the image. In order to minimize L_R we need direction of steepest descent

and thus compute the Euler-Lagrange [110] of the above:

$$\mathcal{C}_t = g(\mathcal{C})\kappa\vec{\mathcal{N}} - (\nabla g(\mathcal{C}) \cdot \vec{\mathcal{N}})\vec{\mathcal{N}} \tag{A.2}$$

where κ is the Euclidean curvature at a point on the curve and $\vec{N}$ is the unit inward normal. We represent this using the level-set approach [83]. Assume that the curve $\mathcal{C}$ is a level-set of a 2D function u for $u = 0$. Consider a planar curve evolving according to:

$$\mathcal{C}_t = \beta\vec{\mathcal{N}}$$

for a given function β. Consider the zero level-set of u at time t:

$$u(\Gamma, t) = 0$$

where Γ is a set 2D points. We have to find an evolution of $u(t)$ such that the evolving curve $\mathcal{C}(t)$ is represented by the evolving zero level-set $\Gamma(t)$ (i.e. $\mathcal{C}(t) = \Gamma(t)$). Differentiating the above we obtain:

$$\nabla u \cdot \Gamma_t + u_t = 0$$

and noting that for any level-set the following holds:

$$\frac{\nabla u}{|\nabla u|} = -\vec{\mathcal{N}}$$

The above equation relates the function u to the curve $\mathcal{C}$. Using the above equations we get:

$$\begin{aligned} u_t &= -\nabla u \cdot \Gamma_t \\ &\Rightarrow -\nabla u \cdot \mathcal{C}_t \\ &\Rightarrow \mathcal{C}_t \cdot \vec{\mathcal{N}} |\nabla u| \\ &\Rightarrow \beta \vec{\mathcal{N}} \cdot \vec{\mathcal{N}} |\nabla u| \\ &\Rightarrow \beta |\nabla u| \end{aligned}$$

Thus rewriting our curve evolution equation with $\beta = g(\mathcal{C})\kappa - (\nabla g(\mathcal{C}) \cdot \vec{\mathcal{N}})$ we get the following evolution equation:

$$\begin{aligned} \frac{\partial u}{\partial t} &= (g(\mathcal{C})\kappa - (\nabla g(\mathcal{C}) \cdot \vec{\mathcal{N}}))|\nabla u| \\ &= g(\mathcal{C})|\nabla u|\kappa + \nabla g(\mathcal{C}) \cdot \nabla u \end{aligned} \tag{A.3}$$

where the steady state solution ($\frac{\partial u}{\partial t} = 0$) will be our result.

Given an initialization of u we can iteratively apply this equation until convergence. The benefit of this level-set approach is that it removes any assumptions about the number and topology of the final shape of object(s) in the image.

A.2 Supervised Geodesic Active Regions

Given an image I and probability distributions p_A, p_B giving the probability of a pixel belonging to a region A or B, we can define the following energy function [84, 85]:

$$\begin{aligned} E(\partial\mathcal{R}(c)) &= (1-\alpha) \iint_{\mathcal{R}_A} g(p_A(I(x,y)))dxdy + (1-\alpha) \iint_{\mathcal{R}_B} g(p_B(I(x,y)))dxdy \\ &\quad + \alpha \int_0^1 g(p_C(I(\partial\mathcal{R}(c))))dc \end{aligned} \tag{A.4}$$

where $\mathcal{R}_A, \mathcal{R}_B$ are the pixels of the two regions, ∂R is the partition boundary, and p_C is the probability distribution of a pixel to belong to the boundary:

$$p_C(x,y) = \frac{p(I(N(x,y))|Boundary)}{p(I(N(x,y))|Boundary) + p(I(N(x,y))|\overline{Boundary})}$$

where $N(x,y)$ is a local neighborhood partition into regions N_R and N_L and the conditional distributions are given by:

$$\begin{aligned} p(I(N(x,y))|Boundary) &= p_A(I(N_R(x,y)))p_B(I(N_L(x,y))) + p_B(I(N_R(x,y)))p_A(I(N_L(x,y))) \\ p(I(N(x,y))|\overline{Boundary}) &= p_A(I(N_R(x,y)))p_A(I(N_L(x,y))) + p_B(I(N_R(x,y)))p_B(I(N_L(x,y))) \end{aligned}$$

When implemented multiple neighborhood partitions are tried and the one giving the maximum boundary probability is used. Applying Green's theorem [47] which relates double integrals over regions in 2D to integrals along contours we again use the Euler-Lagrange to determine the direction of steepest descent and yield a curve evolution equation:

$$\begin{aligned} \frac{\partial(\partial\mathcal{R})}{\partial t} &= (1-\alpha)(g(p_A(I(\partial\mathcal{R}))) - g(p_B(I(\partial\mathcal{R})))) \\ &\quad +\alpha(g(p_C(I(\partial\mathcal{R})))\kappa - \nabla g(p_C(I(\partial\mathcal{R}))) \cdot \vec{\mathcal{N}})\vec{\mathcal{N}} \end{aligned} \tag{A.5}$$

Again g is a function such that $g(x) \to 0$ as $x \to \infty$. Embedding the above curve evolution as the zero level-set of a function u we obtain the following surface evolution equation:

$$\begin{aligned} \frac{\partial u}{\partial t}(x,y) &= (1-\alpha)(g(p_A(I(x,y))) - g(p_B(I(x,y))))|\nabla u(x,y)| \\ &\quad +\alpha(g(p_C(I(x,y)))\kappa(x,y)|\nabla u(x,y)| - \nabla g(p_C(I(x,y))) \cdot \nabla u(x,y)) \end{aligned} \tag{A.6}$$

Though layed out here as a segmentation of two regions this method can be applied to the separation of many regions. One should notice that the portion of Equation-A.6 dealing with the boundary is the geodesic active contour equation.

Appendix B

Energy Functions for Energy Based Models

The form of the energy function is fairly arbitrary. In LeCun et al. [57] they are mostly concerned with regression and classification in the case where you have a set of feature vectors x and associated values y. In the case of classification a particular y_i would provide the class for example x_i. LeCun et al. suggests the following energy function form for regression:

$$E(w, y_i, x_i) = \frac{1}{2}((G(w, x_i) - y_i)^2$$

and the following for classification:

$$E(w, y_i, x_i) = -y_i G(w, x_i)$$

where G is a function with parameters w. In the case of regression the energy function E is minimal when the function $G(w, x_i)$ returns a value matching y_i. For classification E is minimal when $G(w, x_i)$ returns a value with the same sign as y_i. The form of G is left open, a few possibilities are:

$$\begin{aligned} G_{Linear}(w, x_i) &= \sum_j w_j x_{ij} \\ G_{Quadratic}(w, x_i) &= \sum_j \sum_k w_{jk} x_{ij} x_{ik} \end{aligned}$$

It should be noted that one can also learn an energy function exclusively on the features:

$$E(w, x_i) = G(w, x_i)$$

This type of function can then be trained to assign higher energies to examples similar to the contrastive training examples (in our case worse segmentations).

Appendix C

Error Amplification

In this appendix we examine the error amplification that occurs when we choose region/class changes as more of the regions are labeled correctly.

C.1 Discrete Outcomes

Let us consider an arbitrary classifier used to decide if a change is an improvement or a degradation. Consider the percentage of true positives, false positives, true negatives and false negatives over a set of examples containing a percentage p of potential improvements:

$$\begin{aligned} \mathbf{tp} &= \alpha p \\ \mathbf{fp} &= \beta(1-p) \\ \mathbf{tn} &= (1-\beta)(1-p) \\ \mathbf{fn} &= (1-\alpha)p \end{aligned}$$

where α is the percentage of potential improvements which are labeled as improvements by the classifier and β is the percentage of degradations that are mis-labeled as improvements. The error we are most interested in is the percentage of false positives so we consider the

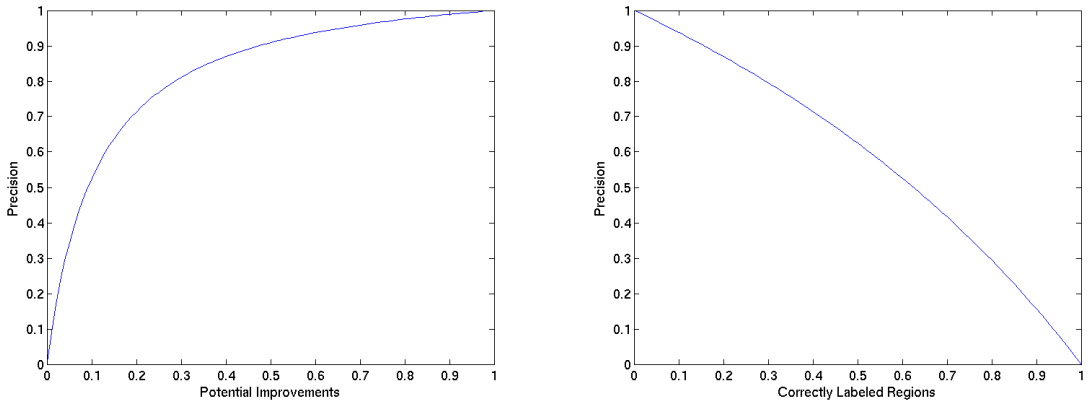

Figure C.1: *Precision vs. possible improvements (***left***) and precision vs. percentage of regions correctly labeled (***right***) for* $c = 7$, $\alpha = 1$, $\beta = 0.1$.

precision:

$$precision = \frac{\mathbf{tp}}{\mathbf{tp} + \mathbf{fp}} = \frac{\alpha p}{(\alpha - \beta)p + \beta}$$

Plugging in Equation 5.1 we can write the precision as a function of the percent of correctly labeled regions:

$$precision = \frac{\alpha(1 - m)}{\alpha(1 - m) + \beta(mc - m))}$$

Let us now consider a very good classifier which on a validation set achieved an $\alpha = 1$ (accurately labeling all improvements) and making few errors among the negative examples with $\beta = 0.1$. In Figure C.1 we use these values to plot both precision vs. potential improvements and precision vs. percent of correctly labeled regions. From the plot on the right we see that even with a small percentage of false positives among the negative examples, when the percent of correctly labeled regions is high we have a very low precision. For example, with 80% of the regions correctly labeled the precision we would get is only 0.3 (where a value of 1 would be ideal giving us no false positives and 0 the worst giving us only false positives). In Figure C.2 we see that as β gets larger the precision drops much more rapidly as the segmentation improves.

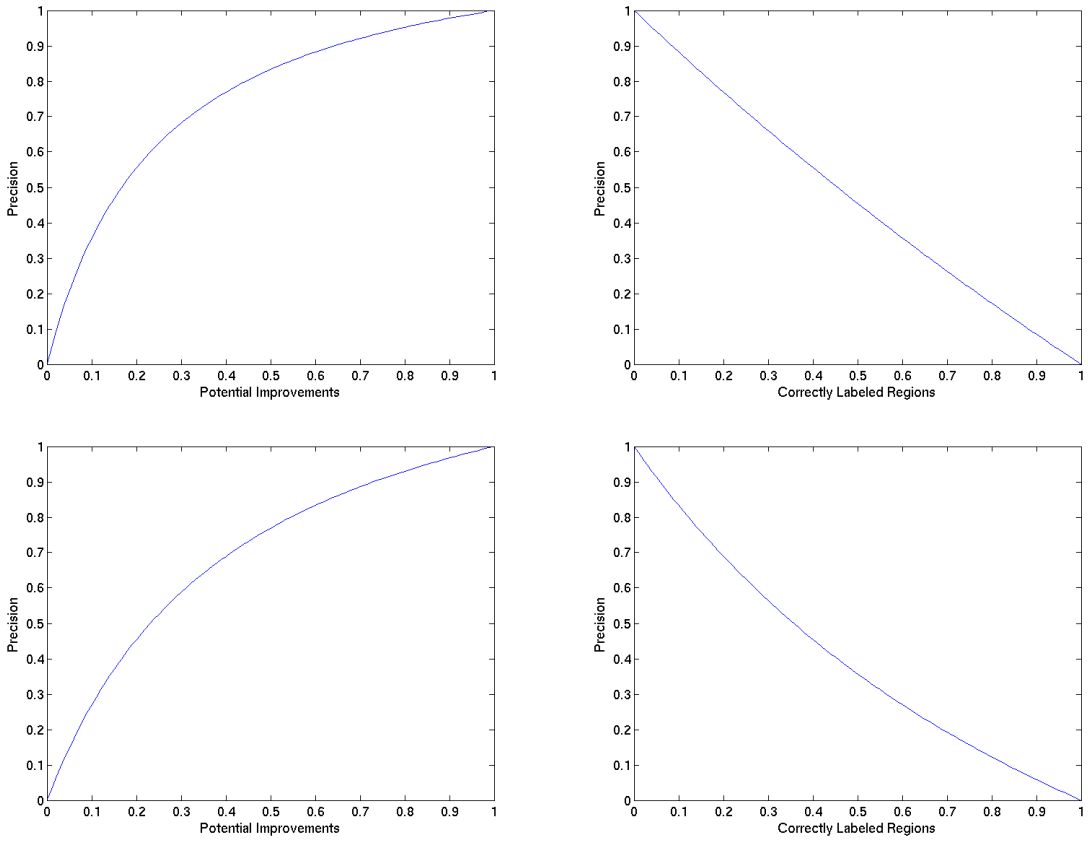

Figure C.2: *Precision vs. possible improvements (***left***) and precision vs. percentage of regions correctly labeled (***right***) for* $c = 7$, $\alpha = 1$, $\beta = 0.2$ *(***top***) and* $c = 7$, $\alpha = 1$, $\beta = 0.3$ *(***bottom***).*

C.2 Continuous Outcomes

Figure C.1 gives us a good idea of the problems we encounter with better segmentations. However, we have ignored the process of actually choosing the classification used as the final output (the one to actually be made). In the case of the boosted trees each classification is in the form a confidence value, such that higher confidences should be considered before lower ones. Similarly with EBM's which assigns energies to changes. In this case however changes with lower energies should be considered before changes with higher energies.

Formally analyzing this process of selecting one change/classification is difficult so instead we simulate the results. For ease of analysis let us assume that improvements are assigned

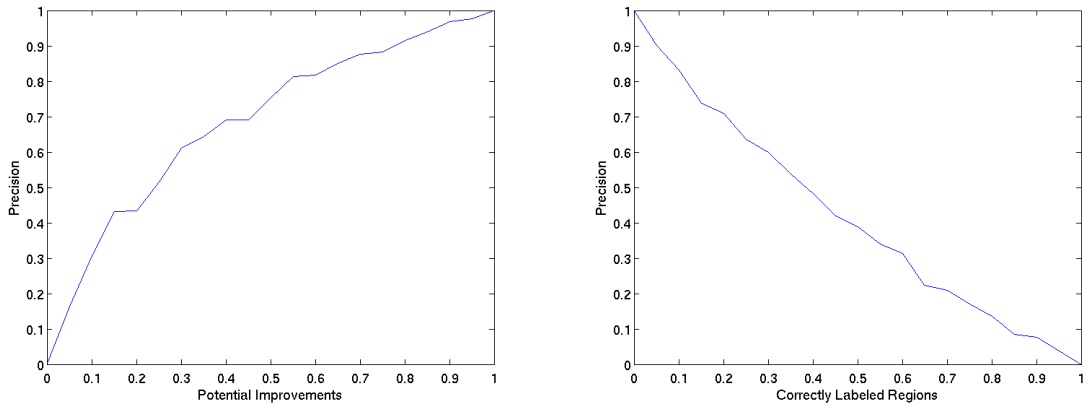

Figure C.3: *Precision vs. possible improvements (***left***) and precision vs. percentage of regions correctly labeled (***right***) for simulated model which assigns values to improvements from a gaussian distribution with* $\mu_i = 0.1, \sigma_i = 0.5$*, and values to degradations from a distribution where* $\mu_d = 0.9, sigma_d = 0.5$

values from a gaussian distribution with mean μ_i and standard deviation σ_i. Similarly let degradations be assigned values from a guassian distribution with mean μ_d and standard deviation σ_d. For various sampled percentages of correctly labeled regions we produce 1000 sets of values for improvements and degradations by using Monte Carlo sampling from the respective distributions. We then tally the number of times improvements had the lowest value among their set to calculate the precision. In Figure C.3 we show the results for a fairly good model which assigns improvements a mean value of 0.1 and degradations a mean value of 0.9. The standard deviation used is 0.5. Note that though the bulk of the value masses are quite separate (i.e. the model is pretty good), the precision still drops fairly rapidly as the segmentation becomes more correct.

References

[1] E. Adelson. On seeing stuff: The perception of materials by humans and machines. *SPIE*, pages 1–12, 2001.

[2] E. Adelson and P. Aniandan. Ordinal characteristics of transparency. *In the Proceedings of the National Conference on Artifical Intelligence*, pages 77–81, 1990.

[3] S. Agarwal, S. Mallick, D. Kriegman, and S. Belongie. On refractive optical flow. *In the Proceedings of the European Conference on Computer Vision*, pages 483–494, 2004.

[4] K. Barnard and G.D. Finlayson. Shadow identification using colour ratios. *8th Color Imaging Conference*, 2000.

[5] M. Ben-Ezra and S. Nayar. What does motion reveal about transparency? *In Proceedings of the International Conference on Computer Vision*, pages 1025–1032, 2003.

[6] J. Bergen, P. Burt, R. Hingorani, and S. Peleg. A three-frame algorithm for estimating two-component image motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1992.

[7] A. Berger, S. Pietra, and V. Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, pages 39–71, 1996.

[8] E. Borenstein and S. Ullman. Class-specific, top-down segmentation. *In the Proceedings of the European Conference on Computer Vision*, 2002.

[9] E. Borenstein and S. Ullman. Learning to segment. *In the Proceedings of the European Conference on Computer Vision*, 2004.

[10] B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, 1992.

[11] G. Brelstaff and A. Blake. Detecting specular reflections using lambertian constraints. *In Proceedings of the International Conference on Computer Vision*, pages 297–302, 1988.

[12] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1986.

[13] B. Caputo, E. Hayman, and P. Mallikarjuna. Class-specific material categorisation. *In Proceedings of the International Conference on Computer Vision*, 2005.

[14] C. Carson, M. Thomas, S. Belongie, J. Hellerstein, and J. Malik. Blobworld: A system for region-based image indexing and retrieval. *Third International Conference on Visual Information Systems*, 1999.

[15] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *In Proceedings of the International Conference on Computer Vision*, 1995.

[16] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 1997.

[17] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1995.

[18] M. Collins, R. Schapire, and Y. Singer. Logistic regression, adaboost and bregman distances. *Machine Learning*, 2002.

[19] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002.

[20] C. Cortes and V. Vapnik. Support-vector network. *Machine Learning*, 1995.

[21] D. Cremers, S. Osher, and S. Soatto. Kernel density estimation and intrinsic alignment for knowledge-driven segmentation: Teaching level sets to walk. *Pattern Recognition Proceedings of DAGM*, 2004.

[22] D. Cremers and S. Soatto. A pseudo-distance for shape priors in level set segmentation. *2nd IEEE Workshop on Variational, Geometric and Level Set Methods in Computer Vision*, 2003.

[23] D. Cremers, N. Sochen, and C. Schnorr. Towards recognition-based variational segmentation using shape priors and dynamic labeling. *International Conference on Scale Space Theories in Computer Vision*, 2003.

[24] A. Criminisi and A. Zisserman. Shape from texture: Homogeneity revisited. *In Proceedings of the British Machine Vision Conference*, 2000.

[25] T. Darrell and E. Simoncelli. "nulling" filters and the separating of transparent motion. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1993.

[26] J. Darroch and D. Ratcliff. Generalized iterative scaling for log-linear models. *The Annals of Mathematical Statistics*, pages 1470–1480, 1972.

[27] A. DelPozo and S. Savarese. Detecing specular surfaces on natural images. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[28] H. Etemadnia and M. Alshari. Automatic image shadow identification using lpf in homomorphic processing systems. *In the Proceedings of Digital Image Computing: Techniques and Applications*, 2003.

[29] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, pages 167–181, 2004.

[30] R. Fleming, R. Dror, and H. Adelson. How do humans determine reflectance properties under unknown illumination. *In Proceedings of the IEEE Workshop on Identifying Objects Across Variations in Lighting: Psychophysics and Computation*, 2001.

[31] D. Forsyth and M. Fleck. Finding naked people. *In the Proceedings of the European Conference on Computer Vision*, pages 593–602, 1996.

[32] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[33] S. Hata, Y. Saitoh, S. Kumamura, and K. Kaida. Shape extraction of transparent object using genetic algorithm. pages 684–688, 1996.

[34] E. Hayman, B. Caputo, M. Fritz, and J. Eklundh. On the significance of real-world conditions for material classification. *In the Proceedings of the European Conference on Computer Vision*, 2004.

[35] X. He, R. Zemel, and M. Carreira-Perpinan. Multiscale conditional random fields for image labeling. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[36] G. Healey and T. Binford. Local shape from specularity. *Proc. Image Understanding Workshop*, pages 874–887, 1987.

[37] G. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 2002.

[38] D. Hoiem, A. Efros, and M. Hebert. Automatic photo pop-up. *ACM SIGGRAPH*, 2005.

[39] D. Hoiem, A. Efros, and M. Hebert. Geometric context from a single image. *In Proceedings of the International Conference on Computer Vision*, 2005.

[40] D. Hoiem, A. Efros, and M. Hebert. Putting objects in perspective. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[41] W. Iba and P. Langley. Induction of one-level decision trees. *9th International Conference on Machine Learning*, pages 233–240, 1992.

[42] M. Irani, B. Rousso, and S. Peleg. Computing occluding and transparent motions. *International Journal of Computer Vision*, 1994.

[43] A. Jepson and M. Black. Mixture models for optical flow computation. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1993.

[44] C. Jiang and M. Ward. Shadow identification. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1992.

[45] A. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1999.

[46] S. Ju, M. Black, and A. Jepson. Skin and bones: Multi-layer, locally affine, optical flow and regularization with transparency. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1996.

[47] W. Kaplan. *Advanced Calculus*. Addison-Wesley, 1991.

[48] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1:321–331, 1987.

[49] J. Kaufhold and A. Hoogs. Learning to segment images using region-based perceptual features. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2:954–961, 2004.

[50] E. Khan and E. Reinhard. A survey of color spaces for shadow identification. *ACM Symposium on Applied Perception in Computer Graphics and Visualization*, 2004.

[51] G. Klinker, S. Shafer, and T. Kanade. A physical approach to color image understanding. *International Journal of Computer Vision*, 4:7–38, 1990.

[52] J. Kosecka and W. Zhang. Video compass. *In the Proceedings of the European Conference on Computer Vision*, 2002.

[53] S. Kumar and M. Hebert. Discriminative fields for modeling spatial dependencies in natural images. *In the Proceedings of the Conference on Neural Information Processing Systems*, 2003.

[54] S. Kumar and M. Hebert. Discriminative randome fields: A discriminative framework for contextual interaction in classification. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[55] S. Lazebnik, C. Schmid, and J. Ponce. Affine-invariant local descriptors and neighborhood statistics for texture recognition. *In Proceedings of the International Conference on Computer Vision*, 2003.

[56] S. Lazebnik, C. Schmid, and J. Ponce. Sparse texture representation using affine-invariant regions. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[57] Y. LeCun, S. Chopra, F. Huang, and M. Ranzato. *Predicting Structured Outputs*, chapter A Tutorial on Energy Based Learning. MIT Press, 2006.

[58] T. Leung and J. Malik. Detecting, localizing and grouping repeated scene elements from an image. *In the Proceedings of the European Conference on Computer Vision*, 1996.

[59] T. Leung and J. Malik. Recognizing surfaces using three-dimensional textons. *In Proceedings of the International Conference on Computer Vision*, 1999.

[60] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 2001.

[61] A. Levin, A. Zomet, and Y. Weiss. Learning ot perceive transparency from the statistics of natural images. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[62] A. Levin, A. Zomet, and Y. Weiss. Seperating reflections from a single image using local features. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[63] M. Levin and J. Bhattacharyya. Removing shadows. *Pattern Recognition Letters*, pages 251–265, 2005.

[64] J. Malik, S. Belongie, T Leung, and J. Shi. Contour and texture analysis for image segmentation. *In Proceedings of the International Conference on Computer Vision*, pages 7–27, 2001.

[65] T. Malisiewicz and A. Efros. Improving spatial support for objects via multiple segmentations. *In Proceedings of the British Machine Vision Conference*, 2007.

[66] C. Marions and A. Blake. Shape from texture: the homogeneity hypothesis. *In Proceedings of the International Conference on Computer Vision*, 1990.

[67] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using brightness and texture. *In Advances in Neural Information Processing Systems*, 2002.

[68] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004.

[69] K. McHenry and J. Ponce. A geodesic active contour framework for finding glass. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[70] K. McHenry and J. Ponce. Global features and energy-based models for improving local scene segmentations. *submitted to IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[71] K. McHenry, J. Ponce, and D. A. Forsyth. Finding glass. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[72] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[73] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 2004.

[74] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *IJCV*, 2005.

[75] D. Miyazaki, M. Kagesawa, and K. Ikeuchi. Polarization-based transparent surface modeling from two views. *In Proceedings of the International Conference on Computer Vision*, pages 1381–1386, 2003.

[76] D. Miyazaki, M. Kagesawa, and K. Ikeuchi. Transparent surface modeling from a pair of polarization images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004.

[77] D. Miyazaki, R. Tan, K. Hara, and K. Ikeuchi. Polarization-based inverse rendering from a single view. *In Proceedings of the International Conference on Computer Vision*, 2003.

[78] G. Mori, X. Ren, A. Efros, and J. Malik. Recovering human body configurations: Combining segmentation and recognition. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[79] H. Murase. Surface shape reconstruction of an undulating transparent object. *In Proceedings of the International Conference on Computer Vision*, pages 313–317, 1990.

[80] S. Nayar, K. Ikeuchi, and T. Kanade. Surface reflection: Physical and geometrical perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17, 1991.

[81] P. Nillius and J. Eklundh. Classifying materials from their reflectance properties. *In the Proceedings of the European Conference on Computer Vision*, pages 366–376, 2004.

[82] M. Osadchy, D. Jacobs, and R. Ramamoorthi. Using specularities for recognition. *In Proceedings of the International Conference on Computer Vision*, pages 1512–1519, 2003.

[83] S. Osher and J. Sethian. Front propogation with curvature dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 1988.

[84] N. Paragios and R. Deriche. Geodesic active regions for supervised texture segmentation. *In Proceedings of the International Conference on Computer Vision*, 1999.

[85] N. Paragios and R. Deriche. Coupled geodesic active regions for image segmentation: A level set approach. *In the Proceedings of the European Conference on Computer Vision*, 2000.

[86] N. Paragios and R. Deriche. Geodesic active regions and level set methods for supervised texture segmentation. *International Journal of Computer Vision*, 2002.

[87] J. Platt. *Advances in Large Margin Classifiers*, chapter Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods, pages 61–74. MIT Press, 2000.

[88] J. Quinlan. Induction of decision trees. *Machine Learning*, 1986.

[89] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. *In Proceedings of the International Conference on Computer Vision*, 2007.

[90] T. Randen and J. Husey. Filtering for texture classification: A comparative study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1999.

[91] X. Ren and J. Malik. Learning a classification model for segmentation. *In Proceedings of the International Conference on Computer Vision*, 2003.

[92] T. Riklin-Raviv, N. Kiryati, and N. Sochen. Segmentation by level sets and symmetry. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[93] M. Rousson and R. Deriche. A variational framework for active and adaptive segmentation of vector valued images. *IEEE Workshp on Motion and Video Computing*, 2002.

[94] M. Rousson and N. Paragious. Shape priors for level set representations. *In the Proceedings of the European Conference on Computer Vision*, 2002.

[95] M. Roussson, T. Brox, and R. Deriche. Active unsupervised texture segmentation on a diffusion based feature space. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[96] E. Salvador, A. Cavallaro, and T. Ebrahimi. Shadow identification and classification using invariant color models. *ICASSP*, pages 1545–1548, 2001.

[97] S. Savarese, M. Chen, and P. Perona. Recovering local shape of a mirror surface from reflection of a regular grid. *In the Proceedings of the European Conference on Computer Vision*, 2004.

[98] S. Savarese, L. Fei-Fei, and P. Perona. What do reflections tell us about the shape of a mirror? *Proceedings of the 1st Symposium on Applied Perception in Graphics and Visualization*, 2004.

[99] S. Savarese and P. Perona. Local analysis for 3d reconstruction of specular surfaces. *ICCV*, 2001.

[100] S. Savarese and P. Perona. Local analysis for 3d reconstruction of specular surfaces - part ii. *ECCV*, 2002.

[101] A. Saxena, S. Chung, and A. Ng. Learning depth from single monocular images. *In the Proceedings of the Conference on Neural Information Processing Systems*, 2005.

[102] F. Schaffalitzky and A. Zisserman. Geometric grouping of repeated elements within images. *Shape, Contour and Grouping in Computer Vision*, 1999.

[103] R. Schapire, M. Rochery, M. Rahim, and N. Gupta. Incorporating prior knowledge into boosting. *19th International Conference on Machine Learning*, 2002.

[104] J. Shi and J. Malik. Normalized cuts and image segmentation. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1997.

[105] M. Shizawa and K. Mase. A unified computational theory of motion transparency and motion boundaries based on eigenenergy analysis. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1991.

[106] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearence, shape and context modeling for multi-class object recognition and segmentation. *In the Proceedings of the European Conference on Computer Vision*, 2006.

[107] M. Singh and X. Huang. Computing layered surface representations: An algoritm for detecting and seperating transparent overlays. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2:11–18, 2003.

[108] J. Solem, H. Aanaes, and A. Heyden. Pde based shape from specularities. *LNCS*, pages 401–415, 2003.

[109] J. Stahl and S. Wang. Globally optical grouping for symmetric boundaries. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[110] G. Strang. *Introduction to Applied Mathematics*. Wellesley Cambridge Press, 1986.

[111] R. Szeliski, S. Avidan, and P. Aniandan. Layer extraction from multiple images containing reflections and transparency. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2000.

[112] Y. Teh, M. Welling, S. Osindero, and G. Hinton. Energy-based models for sparse overcomplete representations. *Journal of Machine Learning Research*, 2003.

[113] S. Tominaga and N. Tanaka. Estimating reflection parameters from a single color image. *Computer Graphics and Applications*, 2000.

[114] Z. Tu, X. Chen, A. Yuille, and S. Zhu. Image parsing: Unifying segmentation, detection and recognition. *In Proceedings of the International Conference on Computer Vision*, 2003.

[115] Z. Tu, X. Chen, A. Yuille, and S. Zhu. Image parsing: Unifying segmentation, detection and recognition. *International Journal of Computer Vision*, 2005.

[116] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.

[117] M. Varma and A. Zisserman. Classifying images of materials: Achieving viewpoint and illumination independence. *In the Proceedings of the European Conference on Computer Vision*, pages 255–271, 2002.

[118] S. Yu. Segmentation using multiscale cues. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 247–254, 2004.

[119] S. Yu and J. Shi. Segmentation with pairwise atrraction and repulsion. *In Proceedings of the International Conference on Computer Vision*, 2001.

[120] R. Zabih and V. Kolmogorov. Spatially coherent clustering using graph cuts. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[121] S. Zhu and A. Yuille. Region competition: Unifying snakes, region growing, and bayes/mdl for multi-band image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1996.

Author's Biography

Kenton Guadron McHenry was born in Sacramento, California in 1978. He received his B.S. degree in Computer Science from California State University San Bernardino in 2001. He is completing the Ph.D. degree under the supervision of Prof. Jean Ponce. His research interests include computer vision, computer graphics, and robotics.