Recovering Occlusion Boundaries from an Image

Derek Hoiem
Department of Computer Science
University of Illinois at Urbana-Champaign
dhoiem@illinois.edu

Alexei A. Efros Martial Hebert
Robotics Institute
Carnegie Mellon University
{efros, hebert}@cs.cmu.edu

Abstract

Occlusion reasoning is a fundamental problem in computer vision. In this paper, we propose an algorithm to recover the occlusion boundaries and depth ordering of free-standing structures in the scene. Rather than viewing the problem as one of pure image processing, our approach employs cues from an estimated surface layout and applies Gestalt *grouping principles using a conditional random field (CRF) model. We propose a hierarchical segmentation process, based on agglomerative merging, that re-estimates boundary strength as the segmentation progresses. Our experiments on the Geometric Context dataset validate our choices for features, our iterative refinement of classifiers, and our CRF model. In experiments on the Berkeley Segmentation Dataset, PASCAL VOC 2008, and LabelMe, we also show that the trained algorithm generalizes to other datasets and can be used as an object boundary predictor with figure/ground labels.*

1. Introduction

One major consequence of projecting the 3D scene onto the image plane is occlusion — each object blocks the view of the objects directly behind it. To understand the scene, we must detect and reason about these occlusions. In Figure 1, every object is involved in one or more occlusion relationships. These occlusions can make recognition difficult, but vision systems can compensate with effective occlusion reasoning. For example, we are not surprised that we cannot see the wheels of the truck in the background because they are occluded by the foliage. But, while occlusions may complicate recognition, they also provide valuable depth information. As Magritte playfully observes in his 1965 painting "The Blank Check" (Figure 2), scene interpretation would become quite difficult if objects did not reliably occlude each other. Neurological studies further emphasize the fundamental role of occlusion reasoning in vision. In macaque brains, Bakin et al. [7] find that occlusion boundaries and contextual depth information are represented in the early V2 processing area.

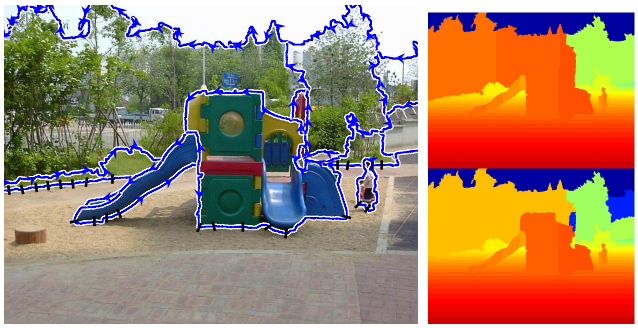


Figure 1. Given an image, we recover occlusion boundaries (left) and infer a range of possible depths (right) that are consistent with the occlusion relationships. In the center, blue lines denote occlusion boundary estimates, the region to left of the arrows is in front, and black hatch marks show where an object is thought to contact the ground. On the right, we display the minimum and maximum depth estimates (red = close, blue = far).

In this paper, we describe an algorithm to recover occlusion boundaries from a single image.[1] We take a machine learning approach. Starting with a large number of initial regions, we classify the boundaries between neighboring regions as occlusion or not. We also classify occlusion boundaries as figure or ground, specifying which region is in front. Unlikely boundaries are sequentially removed, and occlusion likelihoods are re-evaluated as the regions become larger. Our goal is to recover the boundaries and depth ordering of prominent objects in sufficient detail to provide an accurate sense of relative depth.

The greatest challenge is that objects are typically defined, not by homogeneity in appearance, but by physical connectedness. For example, in Figure 1 the most prominent objects are the jungle gym, the boy, and the vegetation. Of these three, only the vegetation can be identified as a single region based on local appearance. How do we have any hope of realizing that the black shorts, white shirt, and small

[1] This paper offers a more complete understanding of the algorithm first described in the conference version [27], providing further background, description, insight, analysis, and evaluation.

Figure 2. "The Blank Check", Magritte, 1965. By ignoring rules of occlusion, Magritte draws our attention to how much we rely on occlusion reasoning for scene interpretation.

circular region above the shirt actually form a single object? What is coherent about the blue slides and green portal of the jungle gym?

We believe that the perception of these structures as individual objects arises from a physical interpretation of the 3D scene. Correspondingly, our approach aims to incorporate physically motivated cues, such as estimates of surface geometry, relative depth, and alignment, as well as traditional segmentation features, such as boundary strength and color differences of regions. Some of these features, such as region alignment and boundary continuity, are more helpful with larger regions than the initial small ones. Therefore, as the regions grow, we apply new classifiers that use features appropriate to the progress of the segmentation. Finally, because many combinations of boundary labels are unlikely or physically implausible, we employ a conditional random field (CRF) model for global consistency.

In summary, we offer several contributions: (1) A broad set of cues for occlusion and figure/ground labeling; (2) an agglomerative segmentation approach, in which boundary likelihoods are refined using updated predictors; and (3) a CRF model that leads to more consistent boundary estimates and enforces boundary closure. In experiments on the Geometric Context dataset, we investigate the impact of each of these contributions and demonstrate the overall quality of resulting scene interpretations. We also evaluate the boundary detectors performance on perceptual and object boundary tasks on three external datasets, providing some evidence for the general utility of the algorithm.

2. Background

In *Perception of the Visual World* [19], Gibson declares, "The elementary impressions of a visual world are those of surface and edge." We previously proposed a surface layout [25], that labels pixels according to surface geometry, such as "support" (e.g., road, grass), "vertical planar" (e.g., a building wall), "vertical non-planar porous" (e.g., vegetation or a mesh), "vertical non-planar solid" (e.g., a person or a car), and "sky". This paper complements our surface layout by inferring the edges of the scene and also builds on it by incorporating surface estimates as valuable cues.

Early computer vision successes in image understanding, such as Roberts' blocks world [59], encouraged interest in occlusion reasoning as a key component for a complete scene analysis system. Some success has been achieved by exploiting multiple images of the scene and apparent motion at the boundaries. This can be done based on the local surface near the boundary [73], the difference of the motion estimates on either sides of the boundary [8, 66, 70], or the responses of spatio-temporal filters [68, 69]. Although motion cues are extremely helpful, we are interested in the problem of occlusion reasoning from a single image, where motion cues are not available.

Traditionally, researchers have divided single-view occlusion reasoning into subtasks of segmentation and line labeling to be solved separately. Modern segmentation algorithms attempt to partition the image according to color or texture similarity or gestalt cues, but the resulting boundaries often do not to correspond to complete objects. Figure/ground labeling algorithms work well, but only when given perfect segmentations. Our work stands out as a unified approach to both find and label occlusion boundaries in natural scenes from one image.

2.1. Segmentation

In part, our goal is similar to that of traditional image segmentation — to partition the visual field into meaningful, coherent regions. The major difference is in how we define what makes a coherent region. Some segmentation methods rely on 2D brightness, color, or texture cues to group the image pixels into perceptually similar regions [64, 50, 58, 18, 5]. Though this grouping is sometimes performed based on pre-computed affinities (e.g., [64]), others advocate a gradual approach, such as the hierarchical segmentation techniques developed by Ahuja [1] and Arbelaez [5, 6], which we follow in our approach.

Other segmentation methods are based on the observation that many objects are delineated by closed, smooth contours [35]. These approaches typically compute affinities between edge fragments to form a graph from which

contours are computed. The affinities are based on computational realizations of the Gestalt principles of continuation, proximity and closure [76, 44, 15, 22]. The graph can be used directly through graph-based search techniques [15, 30, 3, 33] to find the contours. Alternatively, a global solution can be reached by finding a dominant component in the graph with spectral methods [67, 55, 41, 45].

Because all of these algorithms rely on 2D perceptual grouping cues, the boundaries of such segmentations could be due to reflectance, illumination or material discontinuities, as well as occlusions, and resulting regions often do not correspond to actual objects (see Berkeley Segmentation Dataset (BSDS) [50]). Our physical definition of boundaries provides a more concrete objective and allows us to build on the 3D surface estimation described in [25].

In using surface label classifier outputs as features, we relate to other works on object-based segmentation, such as [56, 36, 20]. However, our ideal segmentation into objects and major surfaces is different from a segmentation into the geometric classes of [25], and we use no object-specific models.

2.2. Figure/Ground Line Labeling

Much research has gone into assigning occlusion labels to boundaries, once a good segmentation has been achieved. In the domain of simple polyhedral objects, Guzman in 1968 proposed an elegant algorithm for assigning occlusion labels to line segments [21]. The approach, fully developed by Waltz [75] and others [11, 28, 29, 34, 14], defines a set of possible line labels (convex, concave, and occluding) and a set of allowed vertex types (T-junctions, L-junctions, etc). Constraint propagation was used to efficiently rule out globally-inconsistent geometric interpretations.

This line labeling paradigm has been very influential over the years, with extensions to handle curved objects (e.g., [31, 47]) as well as algebraic [71, 72] and MRF-based [61] reformulations. Marill [48] observes that optimization of a global numerical criterion can mimic human 3D interpretation of line drawings, motivating others to formulate the 3D interpretation problem as optimization over an objective function that favors planarity, symmetry, and other "natural" properties [38, 43, 65]. More recent approaches also incorporate topological constraints [10].

However, attempts to transfer these ideas from the world of line drawings to natural images have been largely unsuccessful. The main reason is that detecting boundaries in real images is in itself an extremely hard problem. Directly detected T-junctions are not as helpful as they would intuitively seem. In fact, recent psychophysics experiments [52] suggest that T-junctions may not be the cause of occlusion percepts, but rather their byproduct.

This may partially explain why recent methods to infer figure/ground labels in natural images [57, 40] tend to work much better for manually-provided boundaries than automatically detected boundaries. Following similar strategies to the 2.1D work of Nitzberg and Mumford [54], they approach the problem in two stages: 1) segment the image, 2) assign a figure/ground label to each boundary fragment according to local image evidence and global MRF-learned constraints. Given a perfect segmentation, their methods are able to produce impressive results on difficult natural images (about 80% accuracy, versus 50% chance accuracy). But without perfect segmentation, the performance drops dramatically (to about 70%). These works are also noteworthy for their investigations into a diverse set of figure/ground cues and their experimental support of CRF models. Note that the boundaries that we seek to recover (major boundaries of free-standing structures) are a subset of the occlusion boundaries considered in these works, which could include boundaries within objects, such as between a person's hair and face.

2.3. Single-View 3D Reconstruction

Our goal of recovering depth relates to recently proposed methods for single-view 3D reconstruction. Our surface estimates [25] can sometimes be used to reconstruct a coarse 3D model of a scene [24]. Saxena et al. [62, 63] train with range images to estimate depth from the image features directly. These methods, however, are likely to oversimplify the 3D model when the scene contains many foreground objects. By explicitly reasoning about occlusions, we enable much more accurate and detailed 3D models of cluttered scenes [26].

3. Algorithm Overview

Our goal is to assign occlusion and figure/ground labels to boundaries in images. This is a very difficult task, partly because occlusion is a physical phenomenon, and we have only a single image. To succeed, we must incorporate a broad range of cues. Our representation includes: color, position, and alignment of regions; strength and length of boundaries; 3D surface orientation estimates; and depth estimates. Most of these features are not local: they require plausible regions to be predictive. The question is how to progress from pixels to more plausible regions to the final occlusion boundaries.

Our strategy is to begin with a conservative oversegmentation into thousands of regions and slowly remove boundaries based on predictions from learned models. As the regions grow, spatial support for computing features improves, and certain features become much more useful. For

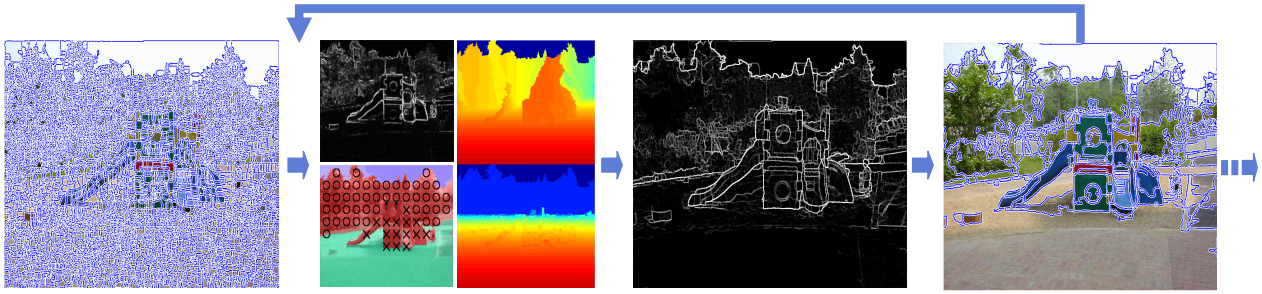


Figure 3. Illustration of our occlusion algorithm. Beginning with an initial oversegmentation into thousands of regions, we gradually progress towards our final solution, iteratively computing cues over boundaries and regions in the current segmentation, estimating a soft boundary map by performing inference over our CRF model, and using the boundary map to create a new segmentation by agglomerative region merging until the weakest boundary is above threshold. At the end of this process, we achieve the result shown in Figure 1.

this reason, we refine boundary predictions as the segmentation progresses, drawing from a set of classifiers that are trained for different stages of the segmentation.

This suggests an iterative procedure, which is illustrated in Figure 3. Each iteration consists of three steps based on the image and the current segmentation: 1) compute cues; 2) assign confidences to boundaries; and 3) remove weak boundaries, forming larger regions for the next segmentation.

The initial boundaries are created using watershed segmentation with the soft boundary map provided by the pB algorithm of Martin et al. [49] (without non-maxima suppression, as suggested by [5]). This will typically produce thousands of regions, preserving nearly all true boundaries. Using ground truth labels on these boundaries, we train classifiers that predict occlusion and figure/ground labels given the features described in Section 4. One classifier predicts the label of each boundary independently; another makes predictions given the labels of neighboring boundaries. These classifiers are incorporated into a CRF model, described in Section 5, that encourages continuity and enforces closure, providing a more consistent and plausible solution. The CRF produces a confidence of occlusion and of the figure/ground label for each boundary. Weak boundaries, according the occlusion confidence, are sequentially removed, as detailed in Section 6, until the weakest boundary has confidence greater than some threshold. Then, new classifiers are trained that can make appropriate use of the increased spatial support, and the merging process continues.

4. Cues for Occlusion Reasoning

We want to train classifiers that predict occlusion and figure/ground labels for hypothesized boundary fragments. Below, we describe a variety of cues (i.e., features) that represent statistics over the boundary or the regions on either side (Figure 4). Our occlusion cues are listed in Table 1 and described below. Although each classifier uses all of these features, some, such as similarity of region colors, are more useful for the occlusion vs. non-occlusion prediction, while others, such as signed difference of surface confidences, are more helpful for figure/ground labeling. The effectiveness of these features is experimentally analyzed in Section 8.

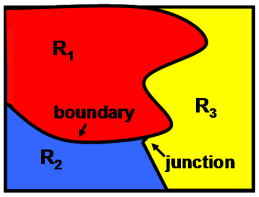


Figure 4. **Illustration of regions, boundaries, and junctions.** Beginning with an oversegmentation, we attempt to determine whether each boundary is an occlusion boundary and, if so, which region is in front. We classify the boundary based on 2D and 3D cues computed over the boundary and neighboring regions. In a conditional random field (CRF) model, we encourage continuity and enforce closure by defining appropriate energy terms over the junctions.

4.1. Boundary Cues

Long, smooth boundaries with strong color or texture gradients are more likely to be occlusion boundaries than short boundaries with weak gradients. To represent boundary strength (B1), we take the mean Pb [49] (probability of boundary) value along the boundary pixels, without applying non-maxima suppression. We also provide a measure of surroundedness (B2): the ratio of boundary length to the perimeter of the smaller region (e.g., $\frac{\text{length}(e_{12})}{\text{length}(e_{12})+\text{length}(e_{23})+\text{length}(e_{24})}$ in Figure 10). We measure smoothness (B3) as the ratio of boundary length to L1 distance between endpoints and orientation (B4) as the difference between the boundary angle (the angle between the endpoints) and the angle of adjacent boundaries (B5). Fi-

Occlusion Cue Descriptions	Num
Boundary	**7**
B1. Strength: average Pb value	1
B2. Length: length / (perimeter of smaller side)	1
B3. Smoothness: length / (L1 endpoint distance)	1
B4. Orientation: directed orientation	1
B5. Continuity: minimum diff angle at each junction	2
B6. Long-Range: number of chained boundaries	1
Region	**18**
R1. Color: distance in L*a*b* space	1
R2. Color: difference of L*a*b* histogram entropy	1
R3. Area: area of region on each side	2
R4. Position: differences of bounding box coordinates	10
R5. Alignment: extent overlap (x,y,overall,at boundary)	4
3D Cues	**34**
S1. GeomContext: average confidence, each side	10
S2. GeomContext: signed difference of S1 between sides	5
S3. GeomContext: sum absolute S2	1
S4. GeomContext: most likely main class, both sides	1
S5. GeomTJuncts: two kinds for each junction	4
S6. GeomTJuncts: if both junctions are GeomTJuncts	2
S7. Depth: three estimates (log scale), each side	6
S8. Depth: diffs of S7, abs diff of first estimate	4
S9. Depth: diff of min overestimate, max underestimage	1

Table 1. **Cues for occlusion labeling.** The "Num" column gives the number of variables in each set. We determine which side of a boundary is likely to occlude (neither, left, right) based on estimates of 3D surfaces, properties of the boundary, and properties of the regions on either side of the boundary. Some information (such as S1) is represented several ways to facilitate learning and classification.

nally, we apply a simple chaining algorithm to link approximately (within 45 degrees) continuous boundaries together (B6).

4.2. Region Cues

It is also helpful to consider features of the regions that are separated by the boundary. Adjacent regions that have similar colors or are well-aligned are more likely to correspond to the same object. Position can be a valuable cue for figure/ground labeling, as the lower region is more likely to be foreground. We represent color in L*a*b* space, and we use as cues the difference of mean color (R1 in Table 1) and the difference between the entropy of histograms (8x8x8 bins) of the individual regions versus the regions combined (R2). We also represent the area (R3), position and differences of the bounding box coordinates (R4), and the alignment of the regions (R5). The positional features are illustrated and detailed in Figure 5.

4.3. Surface Layout Cues

The surface estimates from [25] are highly predictive of occlusion boundaries and figure/ground labels. For example, a woman standing in front of a building is a solid, non-planar surface occluding a planar horizontal surface (the ground) and a planar vertical surface (the building wall). We can take advantage of our work in [25] to recover surface information, which we represent as the average confidence (S1-S4) for each geometric class (horizontal support, vertical planar, vertical solid non-planar, vertical porous, and sky) over each region. These are illustrated in Figure 6.

T-junctions, which occur when one boundary ends on another boundary, have long been used as evidence for an occlusion event [21]. Such junctions, however, are only reliable indicators when they occur at the boundaries of surfaces, not within them. As a cue, we record the event of *geometric T-junctions* (S5-S6) by finding where a boundary chain (B6) transitions from a ground-vertical or sky-vertical to a vertical-vertical boundary, according to the most likely surface labels (S4). In Figure 10, the junctions (e_{12}, e_{23}, e_{13}) and (e_{23}, e_{24}, e_{13}) exemplify the two types of geometric T-junctions.

4.4. Depth-based Cues

Under strong assumptions, we can estimate the depth or depth range (Figure 7) of regions using surface estimates and occlusion boundaries, if we can see where they contact the ground. Although coarse and defined up to a scale, these estimates can provide a good qualitative sense of depth (Figure 15) and convincing 3D reconstructions [26]. These estimates can also be used as features, incorporating information from non-neighboring regions and potentially encoding, for instance, that small pixel misalignments are more significant for objects that are further away. However, we forewarn that our experiments do not confirm the usefulness of these depth cues.

Under assumptions of no camera roll, unit aspect ratio, zero skew, and an approximately level camera and ground, the depth of a point at ground level at pixel row v_i is given by $z = \frac{fy_c}{v_i - v_0}$, where f is the camera focal length, y_c is the camera height, and v_0 is the horizon position. Therefore, given the horizon, the ground-vertical-sky surface labels, and ground-vertical contact points, we can approximate the depth of an image region, up to the scale fy_c. The log ratio depth of two such ground points is $\log(z_2) - \log(z_1) = -\log(v_2 - v_0) + \log(v_1 - v_0)$. Note that, while these estimates are suspect due to strong assumptions, the algorithm need not rely on them.

Given the most likely surface labels, we estimate the horizon to be below the lowest sky label, above the highest

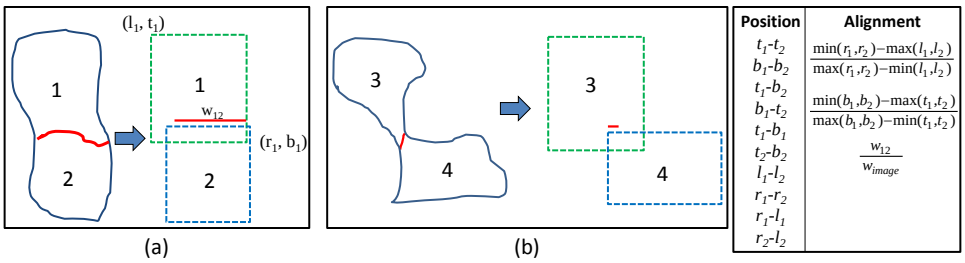


Figure 5. **Illustration of region position features.** The regions in (a), being better aligned, are more likely to be part of the same object than the regions in (b). If the boundaries are due to occlusion, then regions 1 and 3 likely correspond to objects behind those of regions 2 and 4, respectively, because further objects tend to appear higher in the image. On the right, the specific features for position and alignment are detailed, with the bounding box of each region i given by (l_i, t_i) for the left-top and (r_i, b_i) for the right-bottom. The term w_{ij} denotes the horizontal length of the boundary. Note that r is used for right coordinate of the bounding box here but denotes region label elsewhere. We omit details of a fourth alignment feature because it is difficult to concisely describe and not found to be helpful in experiments.

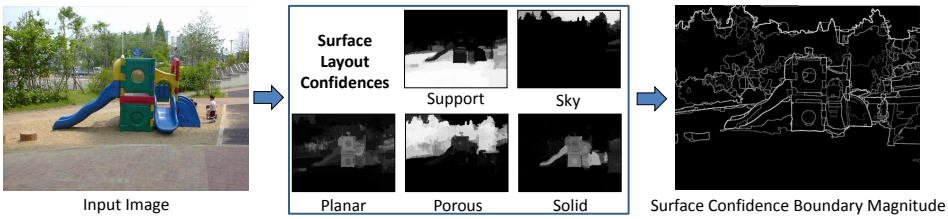


Figure 6. **Illustration of surface layout features.** The method in [25] provides pixel confidences for "support", "vertical planar", "vertical non-planar porous", "vertical non-planar solid", and "sky". These confidences are shown in center, with brighter pixels as more confident in the given label. On the right, the intensity of each boundary pixel is equal to the sum absolute difference of the surface confidences of the regions that it separates. As can be seen, this provides a useful feature for classifying occlusion boundaries. The signed difference of the confidences is very useful for figure/ground classification. Often, the figure/ground label will be very obvious where an object region is adjacent to a ground or sky region, and inference on our CRF model helps to propagate the label into less obvious boundaries.

ground pixel, and as close to the image center as possible. We estimate the ground-vertical contact points using a decision tree classifier based on the shape of the perimeter of a region, as described by Lalonde et al. [37]. For each region, we provide three estimates of depth (S7-S9) corresponding to three guesses of the ground-contact point. The first is estimated by computing the closest ground pixel directly below the object, giving a trivial underestimate of depth. The second assigns the depth of objects without visible ground-contact points as the maximum depth of the objects that occlude it, giving a more plausible underestimate of depth. The third assigns such objects the minimum depth of objects that it occludes, giving an overestimate of depth. The depth range images displayed in our results depict the second and third of these estimates. These estimates are produced by first estimating depth for regions that contact the ground and then iteratively estimating the depth range for the remaining regions based on its occlusion relationships.

5. CRF Model for Occlusion Reasoning

Once we have computed cues over the boundaries and regions from the current segmentation, the next step is to estimate the likelihoods of the boundary labels ("0" for no boundary or the region number of the occluding side) and of the surface labels (into "support", "planar", "porous", "solid", and "sky"). Our conditional random field (CRF) model enables joint inference over both boundary and surface labels, modeling boundary strength and continuity and enforcing closure and surface/boundary consistency. Figure 8 illustrates how a CRF can improve global consistency of the boundary labeling. A poor independent boundary prediction can often be improved by considering the other boundary labels. We note that the boundary terms have much more impact on boundary labeling performance.

We represent the model with a factor graph, in which the probability of the boundaries and surfaces is given by

$$\mathrm{P}(\mathbf{labels}|\mathrm{data}) = \frac{1}{\mathrm{Z}} \prod_{\mathrm{j}}^{\mathrm{N_j}} \phi_{\mathrm{j}} \prod_{\mathrm{e}}^{\mathrm{N_e}} \gamma_{\mathrm{e}} \qquad (1)$$

where in shorthand notation we denote junction factor ϕ_j and surface factor γ_e, with N_j junctions and N_e boundaries in the graph and partition function Z. Boundaries are directed because they have assigned figure/ground labels and can be denoted with an arrow. By our convention, the region to the left of the arrow is in front. It is convenient to speak of boundaries as *incoming* (arrow points towards junction) or *outgoing* (arrow points away from junction).

The junction factors encode the likelihood of the label of

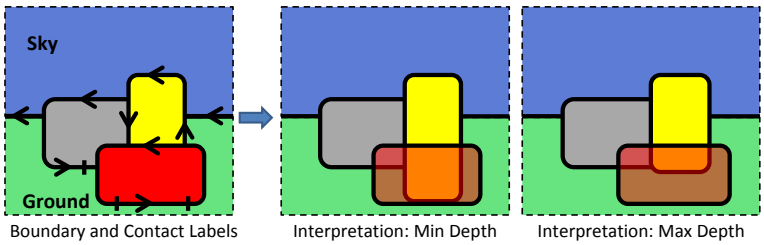


Figure 7. **Illustration of depth range.** In the left image, left side of the arrow is in front, so the depth ordering from front to back, according to the figure/ground labels is red box, yellow box, gray box, ground, and sky. Under simplifying assumptions, we can estimate the relative depth of the red and gray boxes because we can see where they touch the ground plane (shown by hatch marks). We know that the yellow box is no closer than the red box and no further than the gray box, providing a depth range.

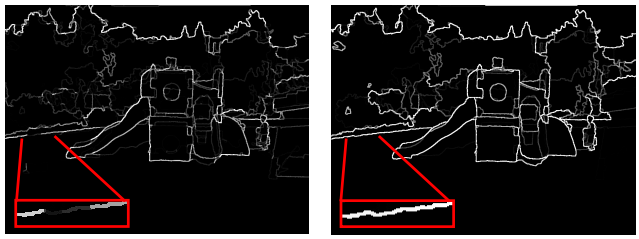


Figure 8. **Example of how the CRF model helps.** On the left, we show the occlusion boundary confidence from independent boundary classifications. On the right, we show the confidence after inference with our CRF. Note, in the magnified boundary, how independent predictions can be physically implausible (two occlusion boundaries with nothing connecting them). Such small errors could lead to an unstable hierarchical segmentation, but inference with the CRF model is often able to correct them. The post-inference predictions are more consistent and, on average, more accurate.

each boundary according to the data, conditioned on its preceding boundary if there is one. They also enforce closure and continuity. Though there are 27 possible labelings of boundary triplets, there are only five valid types of three-junctions up to a permutation. At a scene T-junction, there will be three occlusion boundaries, with at least one outgoing and at least one incoming. Along the edge of an object, there will be one incoming and one outgoing occlusion boundary. Within an object, there will be no occlusion boundary. Other labels are not physically plausible (i.e., other cases imply accidental alignment of object boundaries or interweaving surfaces). Four-junctions are handled in a similar manner. Because edges are defined along cracks in the pixel grid, junctions with more than four edges are not possible.

In the CRF, a very small probability is assigned to invalid junctions. Otherwise, the junction likelihood term is given by the product of the label likelihood of each outgoing boundary, given any corresponding incoming boundary, and the square root of the likelihood of each non-occlusion boundary. This is shown in Figure 9 for three-junctions, with boundary label between regions i and j denoted by e_{ij}. The square root is used to avoid double-counting, since each boundary connects two junctions.

In Figure 10, we illustrate the junction factors with an example of a simple scene. Note that the factor graph, when given a valid labeling, can be decomposed into one likelihood term per boundary. This nice property allows us to learn boundary likelihoods using standard machine learning techniques, such as boosted decision trees, without worrying about CRF interactions (see Section 7 for implementation details). The reasoning over junctions and the definition of valid junctions is reminiscent of the much earlier line labeling work of Waltz [75]. But while Waltz used shading to resolve ambiguities in polyhedral scenes, we learn region and boundary cues to label occlusions in general scenes.

The surface factors encode the likelihood of the surface label of each region according to the data and enforce consistency between the surface labels and the boundary labels. We set penalty term ($\rho_{inconsistent} = \exp(-1)$) for the lack of a boundary between different geometric classes, for the ground region or sky occluding a vertical region, and for sky occluding the ground. We impose a weaker penalty term ($\rho_{floating} = \exp(-0.25)$) for a vertical region being entirely surrounded by another vertical region or by sky (which would imply that the former is floating). Examples of the surface factors are shown in Figure 11. The unary region label likelihood $\mathrm{P}(\mathrm{r_i}|\mathbf{data})$ is computed as the mean surface confidence over the region (S1 in Table 1). To write one surface factor term per boundary, the unary region terms are set to $\mathrm{P}(r_i|\mathbf{data})^{\frac{1}{n_i}}$, where n_i is the number of boundaries surrounding region i. Alternatively, the factor graph could be set up so that each region has a unary factor equal to $\mathrm{P}(r_i|\mathbf{data})$, and the factor of each boundary is used only to impose the "inconsistency" or "floating" penalty.

Even approximate max-product inference over our model is intractable due to the high closure penalties, but the Heskes et al. [23] Kikuchi free energy-based sum-product algorithm, combined with the mean field approximation of raising factors to the $1/T$ ($T = 0.5$ in our experiments),

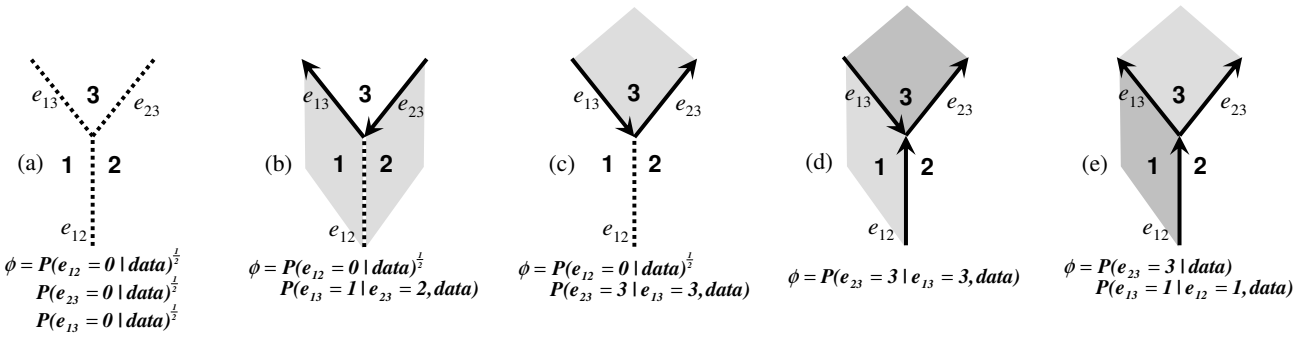


Figure 9. **Illustration of five valid junctions and corresponding likelihoods.** The label of the boundary between regions i and j is denoted as e_{ij} and assigned to 0 (no occlusion), i (region i occludes), or j (region j occludes). By our convention the foreground region (shaded) is to the left of the directed edge. Dotted lines indicate non-occlusion boundaries.

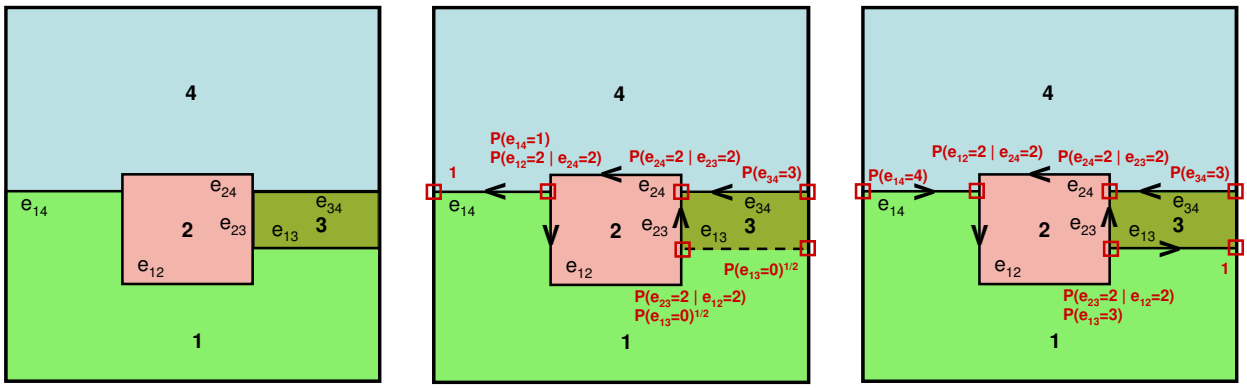


Figure 10. **Illustrating example of a simple scene with junction CRF factors** under two valid interpretations. *Left*, we show the scene and boundaries to label. *Center*, we show a labeling that implies: 1 and 3 are the same surface; 1+3 is in front of 4; and 2 is in front of 1+3. *Right*, we show a labeling that implies a front-to-back depth ordering of regions 2, 3, 4, 1. See Figure 9 for explanation of notation. Here, the data term is omitted to simplify presentation.

as suggested by Yuille [77] efficiently provides "soft-max" likelihood estimates.

6. Segmentation from Boundary Likelihoods

Given a soft boundary map, we can compute a hierarchical segmentation and threshold it to get the initial segmentation for the next iteration (see Figure 12). The hierarchical segmentation is computed by iteratively merging regions with the minimum boundary strength until no boundary is weaker than the given threshold. We define the boundary strength between two regions as the maximum of 1) the value of the strongest boundary between them ($1 - \mathrm{P}(e_{12} = 0|\mathbf{data})$); and 2) a re-estimate of boundary strength computed when new regions are formed. The first of these is the value from our CRF inference. The second is computed by estimating the boundary likelihood of newly formed regions based on quickly computable cues (S1-S4, C1, R1-R5 in Table 1). By incorporating this second estimate, we better handle cases in which two distant regions are clearly different objects but are separated by a set of weak boundaries (as in the case of a slowly varying gradient). Our definition of total boundary strength as the maximum of the two estimates ensures that our merging metric is an ultrametric [5], guaranteeing a true hierarchy. We threshold the hierarchy to provide our next initial segmentation.

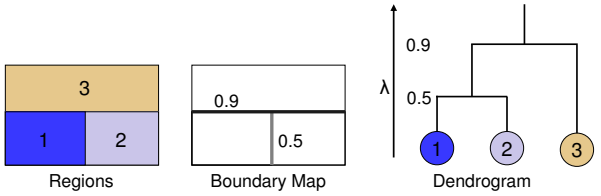


Figure 12. Illustration of hierarchical segmentation from soft boundary map. Here, for hierarchy threshold $\lambda > 0.5$ the two bottom regions are merged, and for $\lambda > 0.9$, all regions are merged.

7. Implementation Details

We train and test our method on our Geometric Context dataset [25], consisting of a wide variety of scenes including beaches, fields, forests, hills, suburbs, and urban streets.

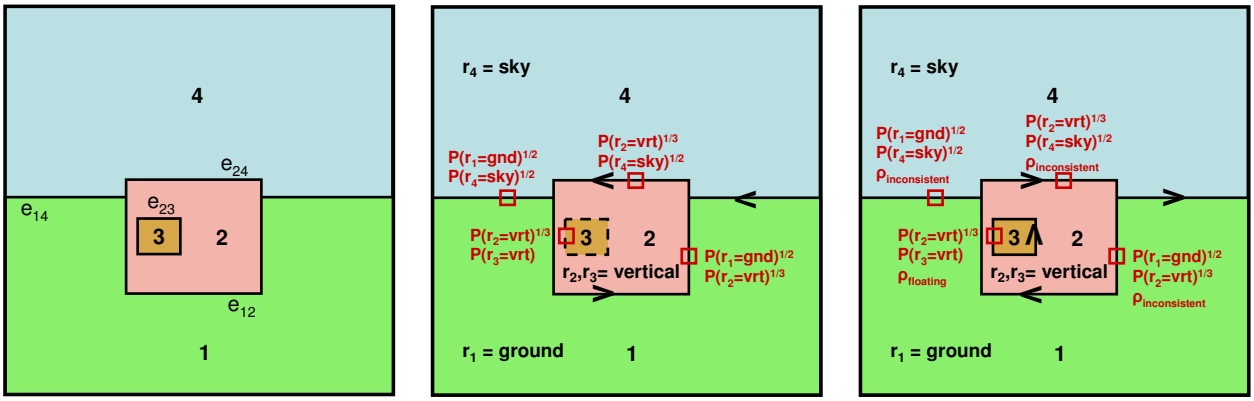


Figure 11. **Illustrating example of a simple scene with surface CRF factors** for two interpretations. A factor is defined on each boundary to penalize inconsistencies and incorporate region label likelihood. Note that if the individually most likely region labels are already consistent with the boundary labeling, these factors will have no effect on boundaries and will assign regions to the most likely labels.

7.1. Assigning Ground Truth

To assign ground truth, we segment each image into thousands of regions, using watershed with pB soft boundaries, and manually group them into object regions, which could be discontinuous due to occlusion. We then assign a depth ordering of objects and use it to assign figure/ground labels. We assigned ground truth to 100 images: 50 for training and 50 for testing. For training, we use the 50 images that were originally used to train the segmentation in the surface estimation algorithm. For quantitative evaluation, we use 50 of the test images from the dataset (specifically, the images from the first fold of the five-fold cross-validation). Examples of the ground truth can be seen in Figure 14. A medium-complexity image will typically contain 10-15 objects according to our ground truth labels.

7.2. Training

After defining the ground truth over our dataset, we train using the algorithm outlined in Figure 13. We estimate the unary ($\mathrm{P}(e_1|\mathrm{data})$) and conditional ($\mathrm{P}(e_1|e_2,\mathrm{data})$) boundary classifiers using a logistic regression version of Adaboost [12], with 20 16-node decision trees as weak learners. This classification method provides good feature selection and probabilistic outputs. The cues used in the unary classifier are described in Section 4. For pairwise cues, we simply concatenate the unary cues for both boundaries and add cues for continuity (relative angle of the two adjacent boundaries) and boundary length (in pixels). Since cues such as color histograms become more useful in the later iterations of our algorithm (with larger regions), we train separate classifiers for the initial segmentations (about 4,400 regions per image, on average) and for the segmentations obtained after the first and second iterations (300 and 100 regions per image, on average, respectively).

In the first two iterations, we set the threshold for the hierarchical segmentation to a conservative value corresponding to an "acceptable" level of pixel error in the training set (1.5%, 2%, respectively), as is typically done in cascade algorithms such as Viola and Jones object detection [74]. The threshold values are 0.105 and 0.25. For instance, in the first iteration, we merge two regions if we are less than 10.5% confident that there is an occlusion boundary between them. The threshold for the remaining iterations can be set to reflect the desired trade-off between the number of regions and how well the true object regions are preserved. In our experiments, we set this threshold to 0.6. To train the boundary classifiers after the first iteration, we transfer the ground truth from the initial oversegmentations to the current segmentations by automatically labeling each region as the object that occupies the largest percentage of its pixels.

In our experiments, we set the surface factor unary term by combining, in a linear logistic model, two likelihood estimates: 1) the multiple segmentation estimate from [25]; and 2) an estimate using the same cues as (1) but using the current segmentation from our occlusion algorithm. The logistic weights (1.34, 0.16) were learned to maximize likelihood of the training surface labels.

7.3. Inference

To evaluate a new image, we perform the algorithm described in the previous sections, initializing with an oversegmentation, and iteratively progressing toward our final solution. In each iteration, we compute cues over the current regions, compute boundary likelihoods in a CRF model based on those cues, and create a new segmentation by merging regions based on the boundary likelihoods. In the first iteration, we restrict our CRF model to the unary boundary likelihoods, since boundary and surface reasoning over the initial segmentation is ineffective and computationally expensive. In the second iteration, we expand our model to include the junction factor terms. In each addi-

TRAINING

Input:
- Training images
- Initial segmentations $\{\mathbf{s}^0\}$ (from watershed/pB)
- Ground truth object regions $\{\hat{\mathbf{s}}^0\}$
- Ground truth boundary labels $\{\hat{\mathbf{e}}^0\}$ (NoEdge/Side1Occludes/Side2Occludes)

For iteration t = 1..3:
1. For each image k: compute cues for segmentation ${\mathbf{s}_k}^{t-1}$ (Section 4)
2. Train boosted decision tree boundary classifiers to get $\mathrm{P}^t(e_i|\mathbf{data})$, $\mathrm{P}^t(e_i|e_j = \text{on}, \mathbf{data})$
3. For each image: compute soft boundary map by CRF inference (Section 5)
4. For each image: compute hierarchical segmentation (Section 6)
5. Set hierarchy threshold by training error
6. For each image k: get next segmentation ${\mathbf{s}_k}^t$ by thresholding hierarchy
7. Update ground truth $\{\hat{\mathbf{s}}^t\}$, $\{\hat{\mathbf{e}}^t\}$ as best fit from $\{\hat{\mathbf{s}}^0\}$, $\{\hat{\mathbf{e}}^0\}$ given $\{\mathbf{s}^t\}$

Output:
- Boundary classifiers for each iteration
- Thresholds for hierarchical segmentations for each iteration

Figure 13. **Procedure for training** our occlusion recovery algorithm. The main idea is that, because we have less information (smaller regions) at the start of the segmentation process, we should train and apply new classifiers as the segmentation progresses. This idea is validated in our experiments, which show that some features are helpful for only the later stages of segmentation. Though procedurally iterative, the algorithm may be viewed as updating classifiers during a single agglomerative merging process.

tional iteration, we perform inference over the full model. The algorithm terminates when no regions are merged in an iteration (typically after 4 or 5 iterations, in total). In our Matlab implementation, our algorithm takes about 460 seconds for a 600x800 image, running on a single thread of a 64-bit Intel core i7 2.93GHz, including about 215 seconds for Pb [49] without non-maximum suppression, 10 seconds for our surface estimation algorithm, and 235 seconds for the occlusion algorithm. By comparison, the full Pb algorithm takes about 660 seconds, and the Global Pb algorithm [46] takes 480 seconds per image on the same processor for the same images.

8. Experiments

Using the Geometric Context dataset, we quantitatively evaluate our method in terms of occlusion boundary classification and figure/ground classification on 50 test images (Section 8.1). We also provide several qualitative results and analyze the impact of our features, of retraining classifiers during agglomerative segmentation, and of the CRF model (Section 8.2). Finally, we compare our algorithm to Pb [51] and Global Pb [46] on several external datasets (Section 8.3) to evaluate our algorithm's usefulness in perceptual and object boundary prediction.

8.1. Results in Geometric Context Dataset

We provide several examples of results from the Geometric Context Dataset in Figure 14. For quantitative analysis of average precision in boundary prediction and accuracy of figure/ground labels, see Table 2. As shown in Figure 15, we can often give a reasonable sense of relative depth by combining simple scene models with predicted boundaries, figure/ground labels, and surface layout labels. In an earlier conference paper [27], we provide further analysis of segmentation accuracy and show that segmentations compare favorably to normalized cuts [13] or using the surface layout labels as a segmentation. In other work [26], we show that the occlusion boundaries are helpful for single-view 3D reconstruction. To avoid overwhelming the reader with experimental results, we suppress the details in this article.

8.2. Analysis of Features and Approach

We analyze several major design decisions of our approach: the choice of features, the retraining during agglomerative segmentation, and the CRF for imposing non-local priors and constraints. For these experiments, we use the Geometric Context dataset.

Feature Importance. We use a comprehensive set of features describing: the length, strength, and continuity of boundaries; area, position, and color similarity of neighboring regions; predicted surface layout confidences; and depth and t-junction cues based on simplified scene models. We would like to know how these features impact performance. To get a rough sense, we evaluated performance of boundary and figure/ground classifiers using different subsets of features, as shown in Table 2. The boundary classifier labels a potential boundary as occlusion or not, while the fig-

Figure 14. **Qualitative test results on the Geometric Context dataset.** Occlusion boundaries are shown in blue and white; the region on the left side of the arrow is thought to be in front. The algorithm tends to do well in separating objects from ground and sky. When nearby objects have similar color or depth, the algorithm has difficulty separating them. Decapitation of humans is also common (row 4, left), encouraged by continuity through the shoulders and color differences of skin and garments. These difficulties and successes can be partly explained as a heavy reliance on surface layout features, though we show in earlier work [27] that segmentation purely based on surface layout performs relatively poorly. Boundaries for detectable objects can be improved with simple mechanisms [26].

ure/ground classifier estimates which side occludes, if any. Because we retrain classifiers at different levels of the agglomerative segmentation, we show results for each classifier. We can see that for both classifiers, the region and boundary features lead to a better classifier than using only Pb estimates [51], and including features based on surface layout predictions gives a substantial improvement. However, the remaining "geometric T-junction" and depth-based cues do not seem to help.

In Figure 16, we also describe a more detailed analysis using L1 regularized logistic regression [32] on standardized (zero mean, unit norm) features. This is a linear classi-

	iter 1	iter 2	iter 3
Pb only	0.332	0.494	0.611
Region+Boundary Cues	0.410	0.603	0.734
R+B+GeomContext Cues	0.577	0.723	0.780
All Cues	0.584	0.702	0.779
All Cues, No Retrain	—	0.698	0.759
All Cues with CRF	—	**0.723**	**0.782**

(a) Occlusion/Non-occlusion Average Precision

	iter 1	iter 2	iter 3
Region+Boundary Cues	58.7%	65.4%	68.2%
R+B+GeomContext Cues	73.2%	77.1%	77.0%
All Cues	71.7%	75.6%	77.1%
All Cues, No Retrain	—	71.9%	74.0%
All Cues with CRF	—	**77.3%**	**79.9%**

(b) Figure/Ground Accuracy

Table 2. **Quantitative test results on the Geometric Context dataset.** Left: Average precision of boundary occlusion/non-occlusion classification, computed for each image and averaged over all test images. Right: figure/ground labeling accuracy, averaged over all boundary pixels. Shown are results using Pb [49], boundary/region cues only (R1-R5,B1-B6 in Figure 1), with additional surface layout cues (S1-S4), with all cues and retraining after each iteration, with all cues and no retraining, and using all cues and retraining after performing inference with our CRF model (only unary likelihoods were used in the first iteration). Using the surface layout cues, retraining, and the CRF model all provide significant gains, while the more complex junction and depth cues (S5-S9) do not help and sometimes degrade performance, possibly due to overfitting. Results are not directly comparable across iterations because there are fewer boundaries in each successive iteration.

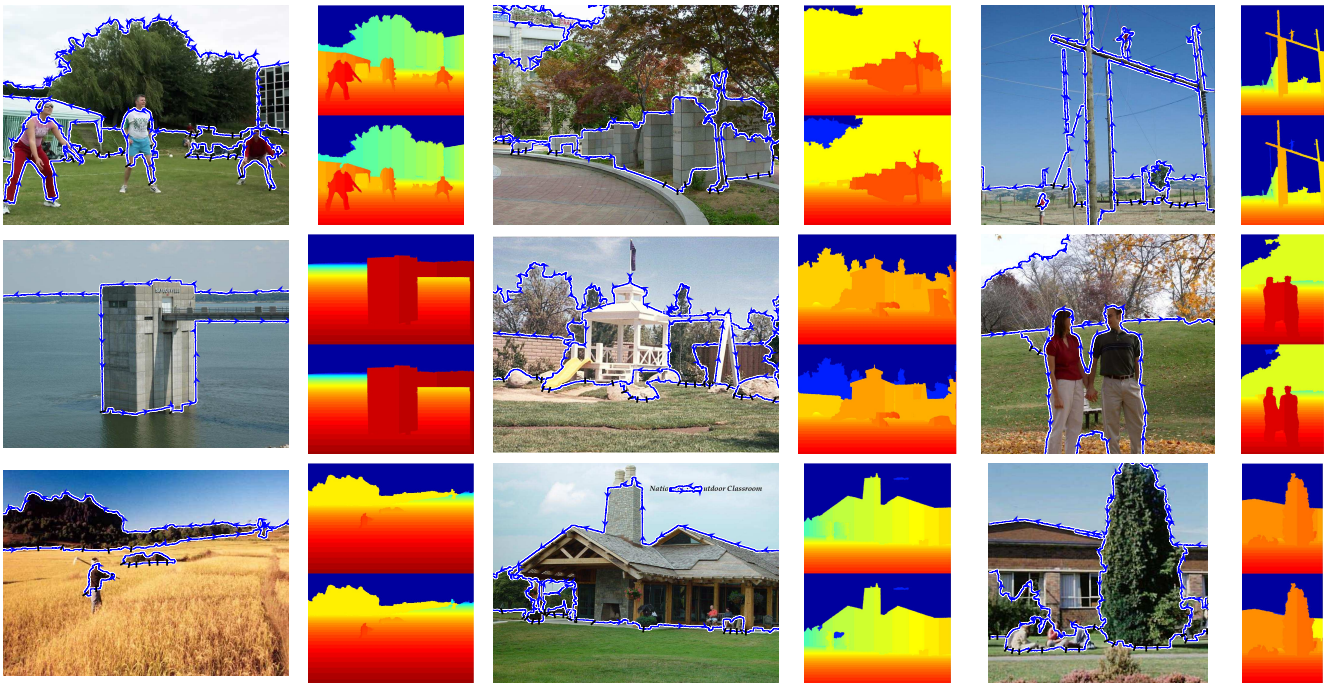
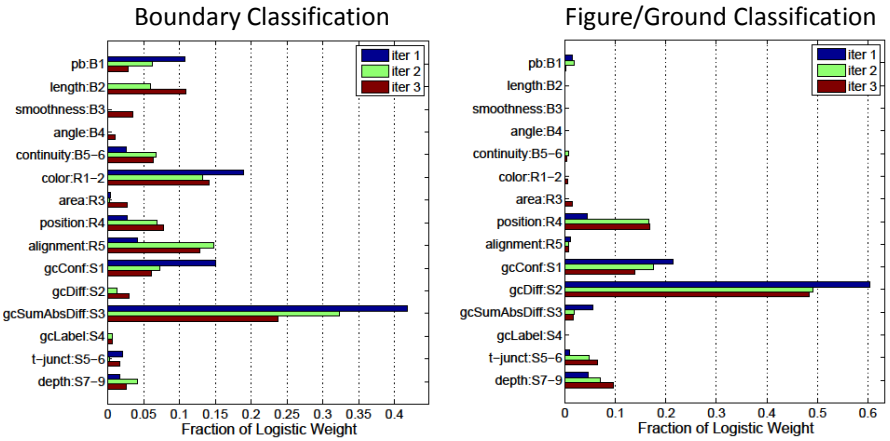


Figure 15. For each image, we show the recovered object boundaries with arrows indicating the foreground object (left side) and black straight lines indicating ground contact points. To the right of each image, we show an estimated depth range, with the minimum depth on top, and the maximum depth underneath. Depth ranges are computed based on the object boundaries, the surface geometry, the ground contact points, and the depth ordering.

fier that can be used for feature selection [53] or learning Markov network structures [39]. Because the regularization encourages sparsity, important features are assigned weights with high magnitudes, and features that add little predictive power are assigned zero weight. As a measure of feature importance, we report weight magnitudes that are normalized to sum to one. Overall, we can see that different features are important for boundary classification and figure/ground classification, but nearly all features carry some weight. Some of these features, such as boundary length and continuity and region position and alignment are not found to be important until later stages, when the regions are larger. For boundary classification, the most important features are the absolute difference of surface confidences, the entropy-based region color feature, Pb confidence, boundary length, and x-alignment of region bounding boxes. For figure/ground classification, the most important features are those based on differences in surface confidences and relative position of neighboring regions.

Retraining Classifiers. Though hierarchical segmentation is a well-worn technique, our approach adds a twist by retraining classifiers each time the merging cost reaches a cer-



Most Important Boundary Features

	Iter1	Iter2	Iter3
S3 (GC Abs Diff)	42%	32%	24%
R2 (Color Ent.)	15%	10%	10%
B1 (Pb)	11%	6%	3%
B2 (Length)	0%	6%	11%
R5 (Bbox XOv)	4%	7%	6%

Most Important Figure/Ground Features

	Iter1	Iter2	Iter3
S2 (GC Sky Diff)	28%	19%	15%
S2 (GC Sol Diff)	17%	17%	17%
S2 (GC Gnd Diff)	14%	9%	10%
S1 (GC Sky Conf)	15%	9%	9%
R4 (Bottom Diff)	1%	7%	7%

Figure 16. **Feature importance**, measured by percentage of weight magnitude assigned to each feature in a L1-regularized linear logistic regression classifier. Different features are important for boundary classification (occlusion boundary vs. no boundary) and figure/ground classification. Some features are important only in later stages, when larger regions are available. Overall, features based on surface layout have the most influence, but nearly all features have some weight. On the right, we list the individual features with the highest magnitude weight, given as a percentage of the total magnitude of the weight vector.

tain threshold. This results in three sets of classifiers. In Table 2, we show that retraining gives an moderate improvement of between 0.02 and 0.04 AP, likely because some features are only effective for larger, more confident regions, as shown in Figure 16. These experiments provide some justification for adjusting classifiers as the segmentation proceeds.

CRF. Our CRF model encodes continuity and imposes a penalty for physically implausible combinations of occlusion and surface labels. The inference is approximate and computationally costly, and we use it only in the later stages of segmentation. In Table 2, we show that the CRF model does give a modest to moderate gain in average precision of boundary prediction.

8.3. Predicting Perceptual and Object Boundaries

Often, boundary predictors are used, not as a result, but as a feature for object recognition or segmentation. In these experiments, we compare our occlusion-based boundary predictor to the well-regarded boundary predictors Pb and GlobalPb, which are trained on 200 images from the Berkeley Segmentation Dataset (BSDS), annotated with perceptual boundaries. Without retraining either algorithm, we compare on three datasets: BSDS test set, LabelMe [60], and the PASCAL VOC 2008 segmentation set [16]. On the perceptual boundaries task (BSDS), the Global Pb classifier performs best, while on the object boundaries tasks (LabelMe and PASCAL VOC), our occlusion algorithm performs best. Further, our occlusion algorithm provides figure/ground confidences, which could be useful for object recognition.

Boundary Prediction. Until now, the final output of our occlusion algorithm has been one segmentation with figure/ground labels over the boundaries. To use boundary prediction as a feature, we would like confidences for each possible boundary pixel, not just those that survive to reach the final segmentation. To get a per-pixel probability of boundary map, we compute the mean of the soft maps produced in the first, second, third, and final iterations. This has the desired effect of down-weighting boundaries that are removed in early stages, without assigning them a zero likelihood. We also experimented with logistic regression weightings, but, overall, it performed similarly to the simple averaging method. To estimate figure/ground likelihood, we compute an average of the figure/ground predictions from each iteration, weighted by the per-iteration occlusion likelihoods.

For evaluating boundary prediction, we use the precision-recall software provided with the BSDS dataset. As a quantitative measure, we compute the mean of the 101-point interpolated average precision for each test image. In LabelMe, the provided test set of 1133 fully labeled images was used, and ground truth was obtained by merging parts into objects, layering the object regions according to estimated depth, and flattening the layers so that each pixel label corresponds to the foremost object. Images were downsampled to a maximum height or width of 800 pixels. Predicted boundaries are considered correct if within 1% of the image diagonal of any object boundary. While the ground truth is not perfect, it is sufficiently accurate for useful evaluation. Our results are shown in Table 3, and examples for LabelMe test images are shown in Figure 17. We show additional results with color-coded figure/ground predictions in Figure 18.

	LabelMe(AP)	LabelMe (Rank1)	BSDS (AP)	BSDS (F)
Pb	0.388	3.7%	0.689	0.656
Global Pb	0.441	4.7%	0.744	0.697
Global Pb + UCM	0.474	22.5%	**0.765**	**0.702**
Occlusion iter 1	0.504	20.6%	0.708	0.656
Occlusion final	**0.522**	**48.6%**	0.698	0.653

Table 3. **Test results for boundary prediction on the LabelMe and BSDS datasets.** Comparison algorithms are Pb [49], Global Pb [46], and an extension of global Pb to an ultrametric contour map (UCM) [6], with each trained on BSDS. We also show results of our algorithm, trained on the Geometric Context dataset, after one iteration (iter 1) and after combining outputs from each iteration (final). Column 1: average precision on LabelMe (AP computed for each image and averaged over all images). Column 2: percentage of images for which each algorithm achieved the highest AP score. Column 3: average precision on BSDS. Column 4: F-measure on BSDS (the standard measure for that dataset). Global Pb with UCM achieves the best results for perceptual boundaries in BSDS, while our algorithm provides the best performance on object boundaries in LabelMe.

Region Extraction. Besides directly evaluating the boundaries, it can also be helpful to evaluate the regions that they produce in a hierarchical segmentation. In doing so, we follow the methodology of Arbelaez et al. [6]. To generate regions with the occlusion algorithm, we compute hierarchical segmentations from the boundary map described above. Likewise, we generate regions for Pb and Global Pb using the hierarchical segmentation with oriented watershed described in [6]. The objective is to produce a set of regions such that at least one region overlaps perfectly with each object region from the ground truth segmentation. The area-weighted average overlap score for a given image is computed as

$$\mathrm{Score}_{\mathrm{area}} = \sum_{\mathrm{i}} \frac{1}{|\mathrm{R_i}|} \sum_{\mathrm{i}} |\mathrm{R_i}| \max_{\mathrm{j}} \mathrm{Overlap}(\mathrm{R_i}, \mathrm{S_j}) \quad (2)$$

and the unweighted score as

$$\mathrm{Score}_{\mathrm{unweighted}} = \sum_{\mathrm{i}} \frac{1}{\mathrm{N_r}} \max_{\mathrm{k}} \mathrm{Overlap}(\mathrm{R_i}, \mathrm{S_j}) \quad (3)$$

where R_i is the i^{th} ground truth object region, $|R_i|$ is the pixel area of R_i, S_j is the j^{th} region from the generated hierarchical segmentation, N_r is the number of ground truth regions, and $\mathrm{Overlap}(\mathrm{R_i}, \mathrm{S_j}) = \frac{\mathrm{R_i} \cap \mathrm{S_j}}{\mathrm{R_i} \cup \mathrm{S_j}}$. We report the mean score, averaged over images, in Table 4. When computing the scores for PASCAL VOC2008, we use the 1023 images in the trainval set with ground truth object segmentations and ignore the "void" regions. Our results, shown in Table 4 and Figure 19, indicate that our algorithm outperforms Global Pb in both datasets, which was previously reported to be the best algorithm for this task [6]. The per-image results of our algorithm in region overlap are highly correlated with those of Global Pb: linear correlation of 0.79 for LabelMe and 0.60 for VOC2008. As we might expect, the correlation between "iter 1" and our final result is even stronger: 0.92 and 0.79 for LabelMe and VOC, respectively.

9. Discussion

Summary of Findings. We have proposed an algorithm for finding major occlusion boundaries in an image and assigning figure/ground labels to them. In addition to the usual segmentation and figure/ground cues, we incorporate features based on surface layout estimates, which experiments show to be very important (Figure 16, Table 2). In generating the boundaries, our algorithm starts with an oversegmentation and gradually removes boundaries until all remaining boundaries are confidently due to occlusion. One of our innovations is to train a set of classifiers that operate at different stages of the agglomerative segmentation process. The motivation is some features become more important as the regions evolve, so that a one-size-fits-all classifier is not appropriate. Our experiments validate this hypothesis, showing that some features are important only in later stages (Figure 16) and that the retraining and refined predictions help (Tables 2 and 3). Our CRF model also improves occlusion and figure/ground classification performance, compared to independent classification (Table 2). Qualitatively, the occlusion boundary results (Figure 14) and relative depth estimates (Figure 15) look promising, and the results on several external datasets (Figures 17 and 18, Tables 3 and 4) indicate that the boundaries may be more generally useful as a pre-process for recognition and other scene analysis tasks.

Limitations, Extensions, and Revisions. Currently, our algorithm does not incorporate object-specific knowledge, which makes it difficult to separate similar objects that are grouped together in the image. The algorithm could be improved by incorporating object detectors, as in [26], or, more interestingly, developing a object localization and segmentation algorithms that apply within some broader domain [17], such as animals or vehicles. Recent work in generic object detection [42, 2, 4] may also be useful for finding boundaries of individual objects.

Currently, we train with only 50 training images. Each annotation takes 5 to 15 minutes of careful work, but more

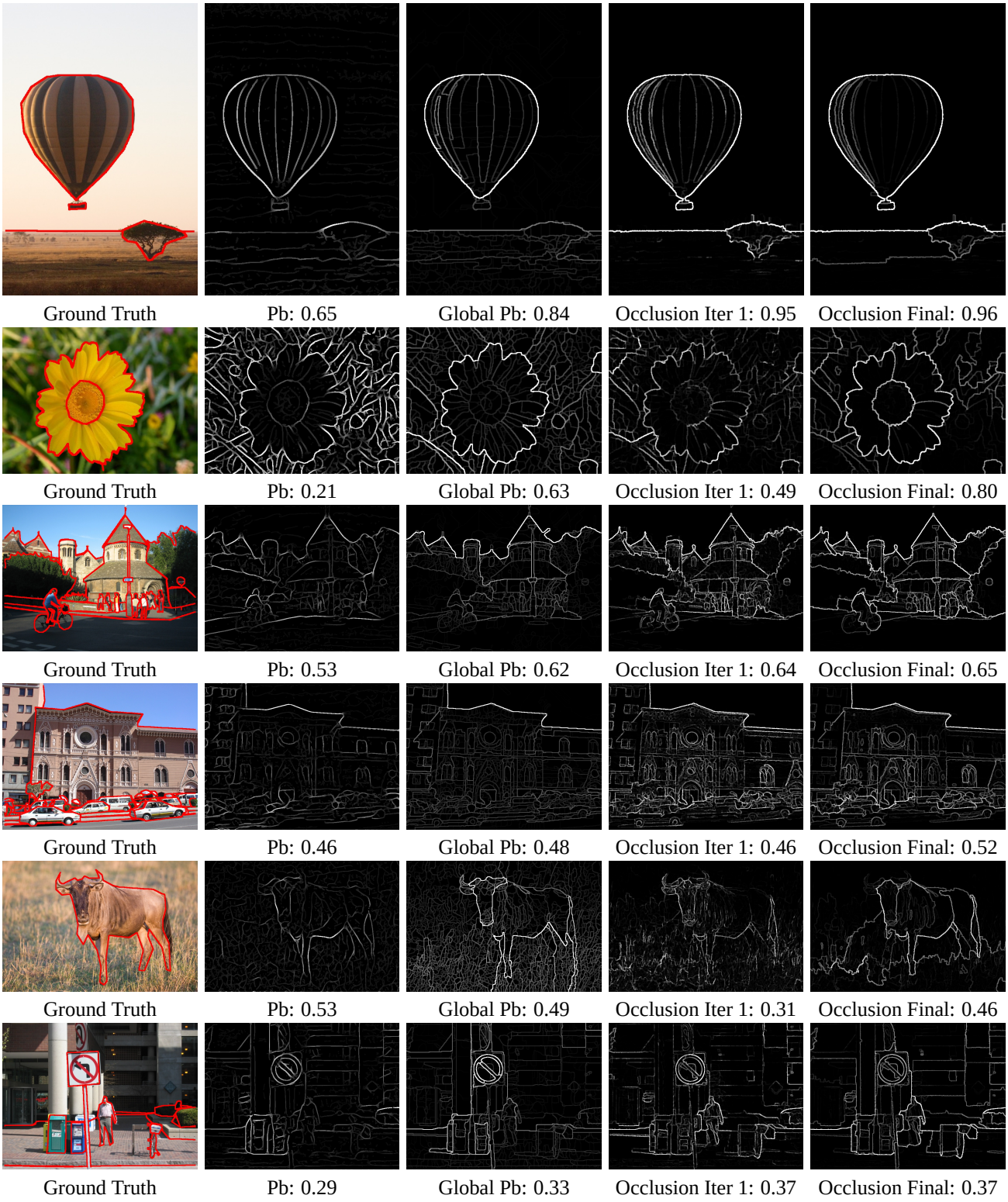


Figure 17. **Representative LabelMe test results for boundary prediction.** Numbers indicate average precision for each example. Red lines denote ground truth boundaries. Pixel brightness indicates predicted boundary strength.

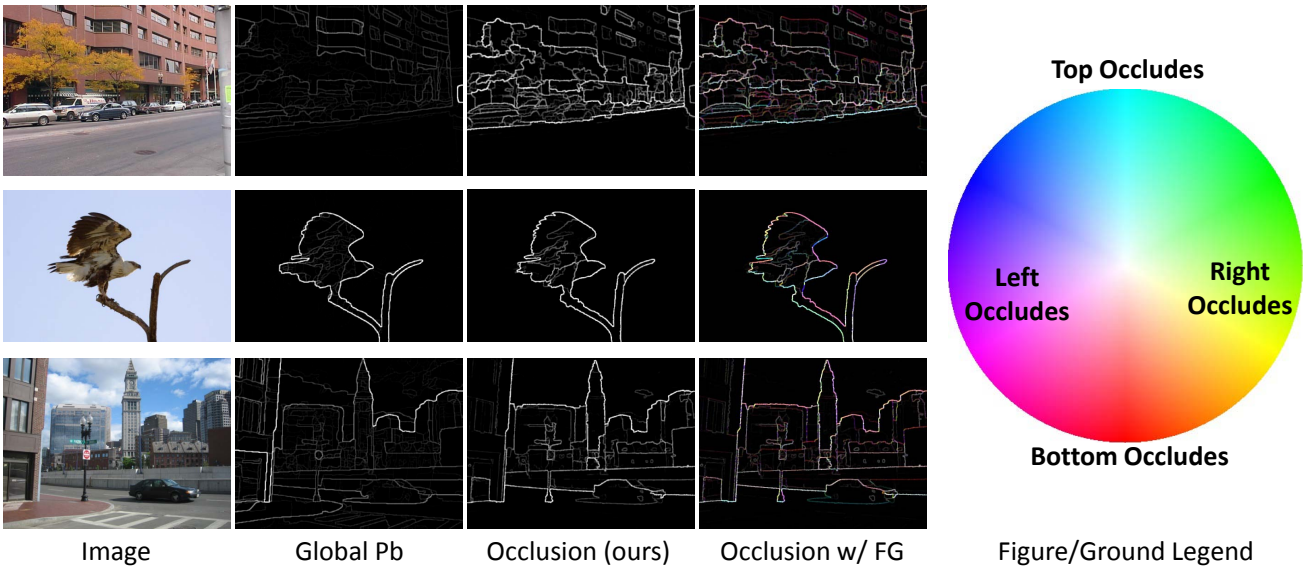


Figure 18. **More boundary prediction results on LabelMe.** In the fourth column, we display occlusion and figure/ground confidence in a single colored image. In HSV coordinates: occlusion confidence is intensity value, figure/ground confidence is saturation, and directed orientation is hue. The legend on the right provides the hue and saturation interpretation (center is 50% likelihood of figure/ground label, outer ring is 100% likelihood; direction to occluding region is hue). For example, a red boundary means that the region beneath is the occluding region, while a green boundary means that the region to the right occludes.

	LM $\text{Score}_{\text{area}}$	LM $\text{Score}_{\text{unweighted}}$	VOC2008 $\text{Score}_{\text{area}}$	VOC2008 $\text{Score}_{\text{unweighted}}$
Pb + UCM	0.602	0.426	–	–
Global Pb + UCM	0.712	0.588	0.620	0.616
Occlusion iter 1	0.753	0.593	0.635	0.629
Occlusion final	**0.769**	**0.613**	**0.671**	**0.665**

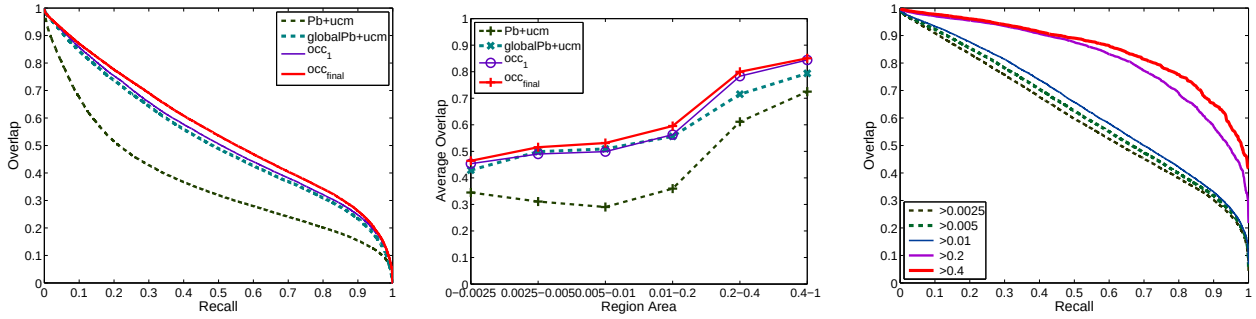
Table 4. Results for region coverage on the LabelMe and PASCAL VOC2008 segmentation datasets. Numbers indicate average overlap (intersection area divided by union area) of each ground truth object region with the best matching region in the hierarchical segmentation, averaged over all test images. Our occlusion algorithm outperforms Global Pb [46, 6] in this task. Our algorithm is trained on 50 images from the Geometric Context dataset, while the Pb algorithms are trained on 200 images from the BSDS dataset.

annotations could be obtained, likely with small to moderate improvements in performance. Alternatively, other modalities, such as video or depth cameras, in which occlusion boundaries can be automatically detected more reliably, could be used to supervise the single-image algorithm. The single image case is important, because we often have access to only one image or a static camera.

Although the code is now available online for download, the time to run the algorithm (a few minutes per image) may be prohibitive to some. The speed of the algorithm might be improved by: using a simpler initial boundary detector (e.g., pre-thresholded Canny [9], rather than Pb [51]); by dropping some of the more computationally expensive and marginally useful features, such as the depth and geometric T-junction cues; finding a faster alternative to the CRF model for incorporating long-range consistency; or beginning with fewer initial regions. Some of these modifications may slightly degrade results, but would dramatically increase speed.

Conclusion. We have proposed a method to recover occlusion boundaries from one image and provided a thorough experimental analysis. The resulting segmentations or depth maps of our algorithm could be directly useful for applications such as image editing or viewing on 3D displays. We also believe that the soft boundary maps with figure/ground labels could be valuable cues for object recognition that are not well-captured by currently popular gradient features. Although this work has made good progress on a difficult problem, we have suggested several paths for improvement, including unsupervised learning from video and incorporation of generic or category-specific object detection models.

Acknowledgements. This material is based upon work supported by the NSF under award IIS-0904209 (DH) and CAREER award IIS-0546547 (AE), as well as a Microsoft Graduate Fellowship (DH). We are grateful to Jenna Hebert for her immensely valuable efforts in ground truth labeling. We thank the Berkeley group for making code available.



(a) Algorithm Comparison (b) Effect of Region Area (c) Overlap Curves by Area

Figure 19. **Comparison of region overlap in LabelMe.** In (a), we show the cumulative distribution (fractional recall) of best-matching regions that have at least the given overlap with ground truth. In (b), we show the average region overlap as a function of ground truth region area. In (c), we show the overlap-recall curves for regions generated by the occlusion algorithm when matching ground truth regions with the given minimum area. For instance, for 70% of the object regions with size of at least 1% of image area, the hierarchical segmentation generated by the occlusion boundaries includes a region that has at least 50% overlap. All algorithms are more effective for larger objects; our algorithm consistently outperforms Global Pb for this task.

References

[1] N. Ahuja. A transform for multiscale image segmentation by integrated edge and region detection. *PAMI*, 18(12), 1996.

[2] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *CVPR 2010*, 2010.

[3] A. Amir and M. Lindenbaum. A generic grouping algorithm and its quantitative analysis. *PAMI*, 20(2), 1998.

[4] anon placeholder. submitted. In *ECCV*, 2010.

[5] P. Arbelaez. Boundary extraction in natural images using ultrametric contour maps. In *Proc. POCV*, 2006.

[6] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. From contours to regions: An empirical evaluation. In *CVPR*, 2009.

[7] J. S. Bakin, K. Nakayama, and C. D. Gilbert. Visual responses in monkey areas v1 and v2 to three-dimensional surface configurations. *The Journal of Neuroscience*, Nov.

[8] M. J. Black and D. J. Fleet. Probabilistic detection and tracking of motion discontinuities. *IJCV*, 38(3):231–245, 2000.

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

[10] L. Cao, J. Liu, and X. Tang. 3D object reconstruction from a single 2D line drawing without hidden lines. In *ICCV*, 2005.

[11] M. Clowes. On seeing things. *Artificial Intelligence*, 2(1):79–116, 1971.

[12] M. Collins, R. Schapire, and Y. Singer. Logistic regression, Adaboost and Bregman distances. *Machine Learning*, 48(1–3), 2002.

[13] T. Cour, F. Benezit, and J. Shi. Spectral segmentation with multiscale graph decomposition. In *CVPR*, 2005.

[14] S. Draper. The use of gradient and dual space in line-drawing interpretation. *Artificial Intelligence*, 17:461–508, 1981.

[15] J. Elder and S. Zucker. Computing contour closure. In *ECCV*, 1996.

[16] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results. http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html.

[17] A. Farhadi, I. Endres, and D. Hoiem. Attribute-centric recognition for cross-category generalization. In *CVPR*, 2010.

[18] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2), 2004.

[19] J. Gibson. *The Perception of the Visual World*. Houghton Mifflin, Boston, USA, 1950.

[20] S. Gould, T. Gao, and D. Koller. Region-based segmentation and object detection. In *NIPS*, 2009.

[21] A. Guzman. Computer recognition of three-dimensional objects in a visual scene. Technical Report MAC-TR-59, MIT, 1968.

[22] L. Herault and R. Horaud. Figure-ground discrimination: A combinatorial optimization approach. *PAMI*, 15, 1993.

[23] T. Heskes, K. Albers, and B. Kappen. Approximate inference and constrained optimization. In *Proc. UAI*, 2003.

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

[25] D. Hoiem, A. A. Efros, and M. Hebert. Recovering surface layout from an image. *IJCV*, 75(1):151–172, 2007.

[26] D. Hoiem, A. A. Efros, and M. Hebert. Closing the loop on scene interpretation. In *CVPR*, 2008.

[27] D. Hoiem, A. N. Stein, A. A. Efros, and M. Hebert. Recovering occlusion boundaries from an image. In *ICCV*, 2007.

[28] D. Huffman. Impossible objects as nonsense sentences. *Machine Intelligence*, 6:295–323, 1971.

[29] D. Huffman. Realizable configurations of lines in pictures of polyhedra. *Machine Intelligence*, 8:493–509, 1977.

[30] D. Jacobs. Robust and efficient detection of convex groups. In *CVPR*, 1993.

[31] R. Jain and J. Aggarwal. Computer analysis of scenes with curved objects. *Proc. of the IEEE*, 67(5):805–812, May 1979.

[32] S. jean Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. An interior-point method for large-scale l1-regularized logistic regression. *Journal of Machine Learning Research*, 8:1519–1555, July 2007.

[33] I. Jermyn and H. Ishikawa. Globally optimal regions and boundaries as minimum ratio weight cycles. *PAMI*, 23(10):1075–1088, 2001.

[34] T. Kanade. A theory of the Origami world. *Artificial Intelligence*, 13:279–311, 1980.

[35] I. Kovacs and B. Julesz. A closed curve is much more than an incomplete one: Effect of closure in figure-ground discrimination. *Proc. Nat'l Academy of Science USA*, 90, 1993.

[36] M. P. Kumar, P. Torr, and A. Zisserman. Objcut: Efficient segmentation using top-down and bottom-up cues. *PAMI*, 32:530–545, 2010.

[37] J.-F. Lalonde, D. Hoiem, A. A. Efros, C. Rother, J. Winn, and A. Criminisi. Photo clip art. In *ACM SIGGRAPH 2007*.

[38] Y. Leclerc and M. Fischler. An optimization-based approach to the interpretation of single line drawings as 3D wire frames. *IJCV*, 9(2), 1992.

[39] S.-I. Lee, V. Ganapathi, and D. Koller. Efficient structure learning of markov networks using L_1-regularization. In *NIPS*. 2007.

[40] I. Leichter and M. Lindenbaum. Boundary ownership by lifting to 2.1D. In *NIPS*, 2009.

[41] T. Leung and J. Malik. Contour continuity in region based image segmentation. In *ECCV*, 1998.

[42] F. Li, J. Carreira, and C. Sminchisescu. Object Recognition as Ranking Holistic Figure-Ground Hypotheses. In *CVPR*, 2010.

[43] H. Lipson and M. Shpitalni. Optimization-based reconstruction of a 3D object from a single freehand line drawing. *Computer-Aided Design*, 28(8), 1996.

[44] D. Lowe. *Perceptual Organization and Visual Recognition*. Kluwer, 1985.

[45] S. Mahamud, L. R. Williams, K. K. Thornber, and K. Xu. Segmentation of multiple salient closed contours from real images. *PAMI*, 25(4), April 2003.

[46] M. Maire, P. Arbelaez, C. Fowlkes, and J. Malik. Using contours to detect and localize junctions in natural images. In *CVPR*, 2008.

[47] J. Malik. Interpreting line drawings of curved objects. *IJCV*, 1(1):73–103, 1987.

[48] T. Marill. Emulating the human interpretation of line-drawings as three-dimensional objects. *IJCV*, 6(2), 1991.

[49] D. Martin, C. Fowlkes, and J. Malik. Learning to find brightness and texture boundaries in natural images. *NIPS*, 2002.

[50] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, 2001.

[51] D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *PAMI*, 26(5):530–549, 2004.

[52] J. McDermott. Psychophysics with junctions in real images. *Perception*, 33(9):1101–1127, 2004.

[53] A. Y. Ng. Feature selection, L_1 vs. L_2 regularization, and rotational invariance. In *ICML*, 2004.

[54] M. Nitzberg and D. Mumford. The 2.1-D sketch. In *ICCV*. 1990.

[55] P. Perona and W. Freeman. A factorization approach to grouping. In *ECCV*, 1998.

[56] M. Prasad, A. Zisserman, A. Fitzgibbon, M. Kumar, and P. Torr. Learning class-specific edges for object detection and segmentation. In *ICCV*, 2006.

[57] X. Ren, C. C. Fowlkes, and J. Malik. Figure/ground assignment in natural images. In *ECCV*, 2006.

[58] X. Ren and J. Malik. Learning a classification model for segmentation. In *ICCV*, 2003.

[59] L. Roberts. Machine perception of 3-D solids. In *OEOIP*, pages 159–197, 1965.

[60] B. C. Russell, A. A. Efros, J. Sivic, W. T. Freeman, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *CVPR*, 2006.

[61] E. Saund. Logic and MRF circuitry for labeling occluding and thinline visual contours. In *NIPS*. 2006.

[62] A. Saxena, S. Chung, and A. Y. Ng. Learning depth from single monocular images. In *NIPS*, 2005.

[63] A. Saxena, S. H. Chung, and A. Y. Ng. 3-d depth reconstruction from a single still image. *IJCV*, 76, 2007.

[64] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. PAMI*, 22(8), August 2000.

[65] K. Shoji, K. Kato, and F. Toyama. 3-D interpretation of single line drawings based on entropy minimization principle. In *ICCV*, 2001.

[66] P. Smith, T. Drummond, and R. Cipolla. Layered motion segmentation and depth ordering by tracking edges. *PAMI*, 26(4):479–494, April 2004.

[67] S.Sarkar and P. Soundararajan. Supervised learning of large perceptual organization: graph spectral partitioning and learning automata. *PAMI*, 22(5), 2000.

[68] A. N. Stein and M. Hebert. Local detection of occlusion boundaries in video. In *BMVC*, 2006.

[69] A. N. Stein and M. Hebert. Using spatio-temporal patches for simultaneous estimation of edge strength, orientation, and motion. In *Beyond Patches Workshop at CVPR*, 2006.

[70] A. N. Stein, D. Hoiem, and M. Hebert. Learning to find object boundaries using motion cues. In *ICCV*, 2007.

[71] K. Sugihara. An algebraic approach to the shape-from-image-problem. *Artificial Intelligence*, 23:59–95, 1984.

[72] K. Sugihara. A necessary and sufficient condition for a picture to represent a polyhedral scene. *PAMI*, 6(5):578–586, September 1984.

[73] R. Vaillant and O. Faugeras. Using extremal boundaries for 3D object modeling. *PAMI*, 14(2):157–173, February 1992.

[74] P. Viola and M. J. Jones. Robust real-time face detection. *IJCV*, 57(2), 2004.

[75] D. L. Waltz. Understanding line drawings of scenes with shadows. In P. Winston, editor, *The Psychology of Computer Vision*, pages 19–91. McGraw-Hill, New York, 1975.

[76] M. Wertheimer. Laws of organization in perceptual forms. In W. D. Ellis, editor, *A Sourcebook of Gestalt Psychology*. Routledge and Kegan Paul, 1938.

[77] A. L. Yuille. CCCP algorithms to minimize the Bethe and Kikuchi free energies: convergent alternatives to belief propagation. *Neural Comp.*, 14(7), 2002.