Recovering Occlusion Boundaries from a Single Image

Derek Hoiem* Andrew N. Stein Alexei A. Efros Martial Hebert
Robotics Institute
Carnegie Mellon University
{dhoiem, anstein, efros, hebert}@cs.cmu.edu

Abstract

Occlusion reasoning, necessary for tasks such as navigation and object search, is an important aspect of everyday life and a fundamental problem in computer vision. We believe that the amazing ability of humans to reason about occlusions from one image is based on an intrinsically 3D interpretation. In this paper, our goal is to recover the occlusion boundaries and depth ordering of free-standing structures in the scene. Our approach is to learn to identify and label occlusion boundaries using the traditional edge and region cues together with 3D surface and depth cues. Since some of these cues require good spatial support (i.e., a segmentation), we gradually create larger regions and use them to improve inference over the boundaries. Our experiments demonstrate the power of a scene-based approach to occlusion reasoning.

1. Introduction

What makes scene understanding different from other image processing tasks, such as medical or aerial image analysis, is the notion that the image is not a direct representation, but merely a projection of the 3D scene. One major consequence of this projection is *occlusion* – the concept that two objects that are spatially separated in the 3D world might interfere with each other in the projected 2D image plane. Consider the scene in Figure 1: nearly every object is partially occluded by some other object, and each occludes part of the ground. Yet, despite their pervasiveness, occlusions have too often been ignored. In computer vision, the study of occlusion reasoning has been largely confined to the context of stereo, motion and other multiview problems (e.g., [3, 24, 28]). For single-view tasks, such as object recognition, occlusions are typically considered a nuisance requiring more robust algorithms.

In this paper, we argue that occlusion reasoning lies at the core of scene understanding and must be addressed explicitly. Our goal is to recover the boundaries and depth ordering of prominent objects in sufficient detail to provide an accurate sense of depth. Our 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 circular region above the shirt actually form a single object?

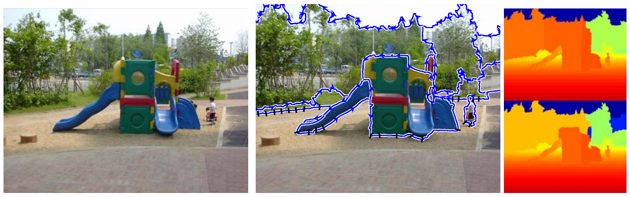


Figure 1. Given an image (left), we recover occlusion boundaries (center) and infer a range of possible depths (right) that are consistent with the occlusion relationships. In the center, blue lines denote occlusion boundary estimates, arrows indicate which region (left) is in front, and black hatch 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).

We believe that the perception of these structures as single objects arises from a physical 3D interpretation of the scene. Correspondingly, we consider a scene to consist of a ground plane, a set of free-standing structures (objects), and the sky. In our example, the entire boy is an object because his whole body is connected to the ground through his legs. Our goal is to determine which parts of the image correspond to ground, objects, and sky and to find the boundaries and a depth ordering of the objects. Then, with estimates of visible object-ground contact points, we can recover a *depth range*, up to a scale. Our approach is to learn models of occlusion based on both 2D perceptual cues and 3D surface and depth cues from a training set. We can then use those learned models to gradually infer the occlusion relationships, reasoning together about boundaries in the image and surfaces in the scene.

While the task we have set for ourselves is clearly very difficult, we believe that it is necessary to make progress in scene understanding. Indeed, the importance of occlusion boundaries for human scene perception has long been

*This research was funded in part by a Microsoft Graduate Fellowship and NSF CAREER award IIS-0546547.

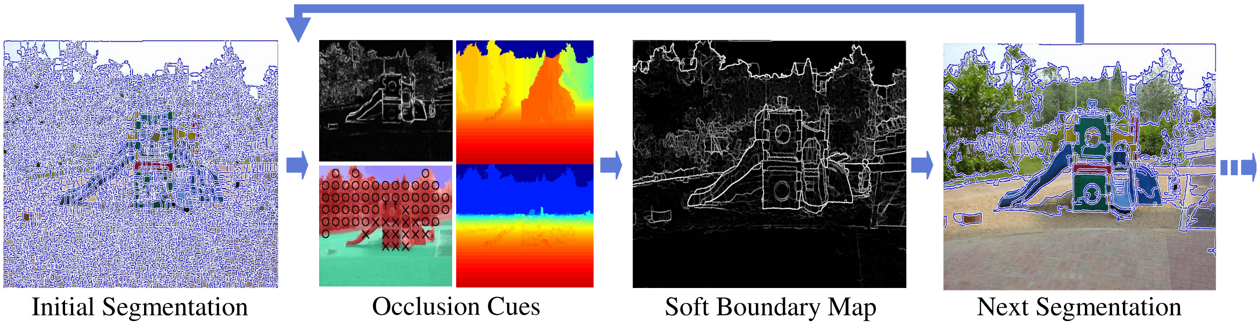


Figure 2. Illustration of our occlusion recovery algorithm. Beginning with an initial oversegmentation into thousands of regions, we gradually progress towards our final solution, iteratively computing cues over boundaries and regions in the current segmentation, estimating a soft boundary map by performing inference over our CRF model, and using the boundary map to create a new segmentation. At the end of this process, we achieve the result shown in Figure 1.

established by psychologists. Gibson argues that occlusion boundaries, together with surfaces, are the basis for the perception of the surface layout of a scene [7]. Biederman includes occlusion (or *interposition*) as one of the five relational rules for a well-formed scene [2]. Good progress has been made in operationalizing several of Biederman's rules, including *likelihood* [27], *support* [13], *size* [12], and *position* [25]. Occlusion reasoning is the natural next step.

1.1. Background

Early computer vision successes in image understanding, such as Roberts' blocks world [20], encouraged interest in occlusion reasoning as a key component for a complete scene analysis system. In 1968, Guzman proposed an elegant approach to interpret polyhedral line drawings: define a set of possible line labels and use constraint propagation to rule out globally-inconsistent geometric interpretations. This approach has been more fully developed by Waltz [30] and others (e.g., [4]), with extensions to handle curved objects [16] as well as algebraic [26] and MRF-based [22] reformulations. While these techniques have been mostly limited to line drawings, Ren et al. [19] have recently proposed a method for labeling occlusion boundaries in images of natural scenes. They take a two-stage approach of image segmentation, followed by figure/ground labeling of each boundary fragment according to local image evidence and a learned MRF model. Given a perfect segmentation, their method produces impressive results on difficult natural images. But performance drops dramatically without perfect segmentation, suggesting that the main difficulty is in finding occlusion boundaries, rather than labeling them.

However, most segmentation algorithms rely on 2D perceptual grouping cues, such as brightness, color, or texture similarity for region-based methods [1, 6, 18] or edge strength, continuity, and closure for contour-based methods (e.g., [14]). As a result, the boundaries of such segmentations could be due to reflectance, illumination, or material discontinuities as well as occlusions, and resulting regions need not correspond to actual objects (see BSDS [18]).

Our goal of recovering depth is similar to recently proposed methods by Hoiem et al. [11] and Saxena et al. [23] for single-view 3D reconstruction. 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.

1.2. Algorithm Overview

Our strategy is to *simultaneously reason about the regions and boundaries in the image and the 3D surfaces of the scene* using learned models. We learn to identify boundaries based on a wide variety of cues: color, position, and alignment of regions; strength and length of boundaries; 3D surface orientation estimates; and depth estimates. In a conditional random field (CRF) model, we also encode Gestalt cues, such as continuity and closure, and enforce consistency between our surface and boundary labels.

To provide an initial conservative hypothesis of the occlusion boundaries, we apply the watershed segmentation algorithm to the soft boundary map provided by the Pb algorithm of Martin et al. [17] (skipping the non-maxima suppression step, as suggested by Arbelaez [1]). This produces an oversegmentation into thousands of regions that preserves nearly all true boundaries. In training, we assign ground truth to this initial hypothesis. Given a new image, our task is to group the small initial regions into objects and assign figure/ground labels to the remaining boundaries.

To get a final solution, we could simply compute cues over each region and boundary and perform a single segmentation and labeling step. However, the small regions from the initial oversegmentation do not allow the more complicated cues, such as depth, to be reliable. Furthermore, global reasoning with these initial boundaries is ineffective because most of them are spurious texture edges.

Our solution is to gradually evolve our segmentation by iteratively computing cues over the current segmentation and using them with our learned models to merge regions that are likely to be part of the same object. In each iteration, the growing regions provide better spatial support for complex cues and global reasoning, improving our ability

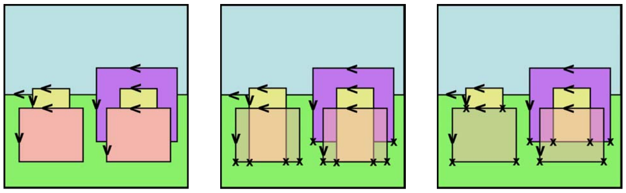
Occlusion Cue Descriptions	Num
Region	**18**
R1. Color: distance in L*a*b* space	1
R2. Color: entropy difference of L*a*b* histograms	1
R3. Area: area of region on each side	2
R4. Position: differences of bounding box coordinates	10
R5. Alignment: extent overlap (x,y, overall,at bndry)	4
Boundary	**7**
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
3D Cues	**34**
S1. GeomContext: average confidence, each side	10
S2. GeomContext: difference of S1 between sides	5
S3. GeomContext: sum absolute S2	1
S4. GeomContext: most likely main class, both sides	1
S5. GeomTJuncts: if event of vrt-gnd-vrt, vrt-sky-vrt	4
S6. GeomTJuncts: if S8 for both boundary endings	2
S7. Depth: three estimates, each side	6
S8. Depth: discontinuity along boundary	3
S9. Depth: minimum depth discontinuity (un/signed)	2

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.

to determine whether remaining boundaries are likely to be caused by occlusions. Each iteration (illustrated in Figure 2) consists of three steps based on the image and the current segmentation: (1) compute cues; (2) assign confidences to boundaries and regions; and (3) remove weak boundaries, forming larger regions for the next segmentation. We describe each of these steps in the following sections.[1]

2. Cues for Occlusion Reasoning

Traditional approaches to segmentation are based on region color and texture cues or boundary cues, such as gradient strength. These 2D cues are helpful for occlusion reasoning as well, but we can further benefit from 3D cues of surface orientations and depth because our segmentations are defined according to physical boundaries. Using all of these cues, we learn to detect whether a boundary is likely to be due to an occlusion and, if so, which side is likely to be in front. Our occlusion cues are listed in Table 1 and described below.



(a) Image (b) Min Depth (c) Max Depth

Figure 3. Illustration of minimum and maximum depth estimates. In (a), we show a segmentation with figure/ground labels (left of arrow is foreground). In (b) and (c), we show the ground-contact points ('X') corresponding to minimum and maximum estimates of depth. If the contact points of a region are visible, the depth of the region is known (up to a scale). Otherwise, we estimate a depth range, based on the visible portion of the region and its occlusion relationships to objects with known depth.

Region Cues. Adjacent regions are more likely to be separate objects with an occlusion boundary between them if they have different colors or textures or are misaligned. Further, as lower regions tend to be closer, image position is a valuable cue for figure/ground labeling. 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 and center coordinates (R4), and the alignment of the regions (R5), measured by overlap of minimum to maximum position along each axis.

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 [17] (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. We measure smoothness (B3) as the ratio of boundary length to Euclidean distance between endpoints, orientation (B4) as the angle between endpoints, and continuity (B5) as the difference between the orientations of adjacent boundaries. Finally, we apply a simple chaining algorithm to chain approximately (within 45 degrees) continuous boundaries together (B6).

3D Surface Cues. Many occlusion boundaries can be found by determining where two adjacent regions have different 3D surface characteristics. For instance, 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 the work of Hoiem et al. [13] 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.

T-junctions, which occur when one boundary ends on another boundary, have long been used as evidence for an occlusion event [8]. Such junctions, however, are only reliable

[1] See Chapter 4 of the dissertation of Hoiem [10] for further details. Data and code is publicly available.

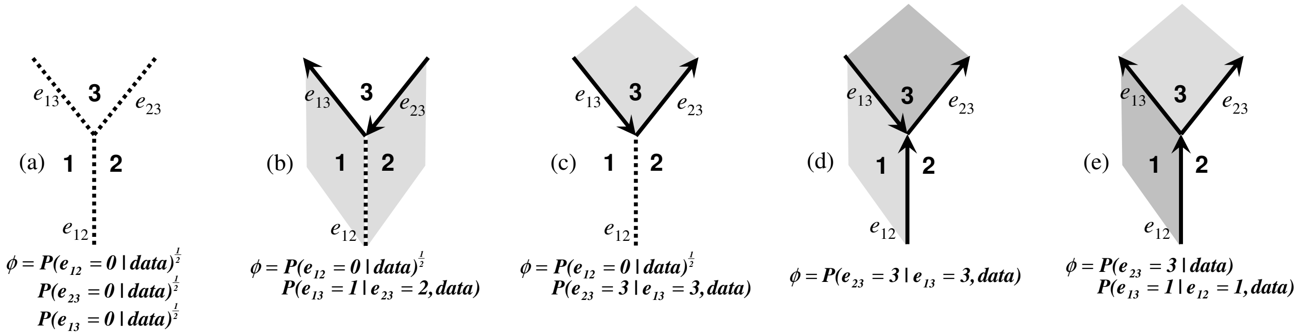

(a)
(b)
(c)
(d)
(e)
1
2
3
e_{12}
e_{13}
e_{23}
$\phi = P(e_{12} = 0 \mid data)^{\frac{1}{2}}$ $P(e_{23} = 0 \mid data)^{\frac{1}{2}}$ $P(e_{13} = 0 \mid data)^{\frac{1}{2}}$
$\phi = P(e_{12} = 0 \mid data)^{\frac{1}{2}}$ $P(e_{13} = 1 \mid e_{23} = 2, data)$
$\phi = P(e_{12} = 0 \mid data)^{\frac{1}{2}}$ $P(e_{23} = 3 \mid e_{13} = 3, data)$
$\phi = P(e_{23} = 3 \mid e_{13} = 3, data)$
$\phi = P(e_{23} = 3 \mid data)$ $P(e_{13} = 1 \mid e_{12} = 1, data)$

Figure 4. The five types of valid junctions with the expressions for their corresponding potentials. By convention the foreground region (shaded) is to the left of the directed edge. Dotted lines indicate non-occlusion boundaries.

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).

3D Depth Cues. If we can determine that there is a large depth discontinuity between adjacent regions, the boundary is likely to be an occlusion boundary. Though we cannot calculate the absolute depth of a region without knowing the camera parameters, we can estimate the relative depth between two regions if we can see where each region contacts the ground. When the ground contact point of a region is occluded, we can estimate a possible depth range for that region (see Figure 3).

More formally, under assumptions of no camera roll, unit aspect ratio, zero skew, and an approximately level camera, 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 depth log difference of two such ground points is $\log(z_2) - \log(z_1) = -\log(v_2 - v_0) + \log(v_1 - v_0)$.

We estimate the horizon to be below the lowest sky pixel, above the highest 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. [15]. 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.

3. 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 (denoted 0 for no boundary or the region number of the occluding side) and of the surface labels (into "ground", "planar", "porous", "solid", and "sky"). Our CRF model allows joint inference over both boundary and surface labels, modeling boundary strength and continuity and enforcing closure and surface/boundary consistency.

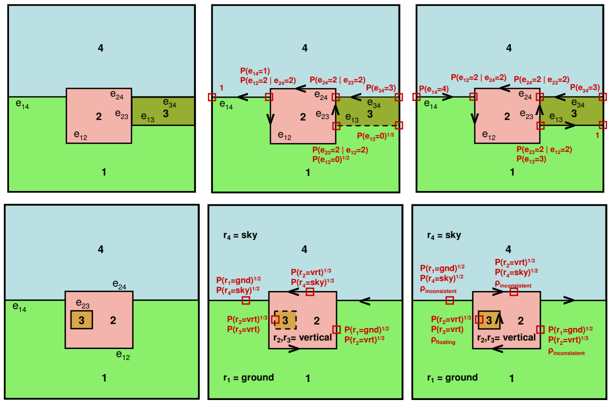
We represent the model with a factor graph, in which the probability of the boundaries and surfaces is given by

$$\mathrm{P}(\mathbf{labels}|\mathbf{data}) = \frac{1}{Z}\prod_{j}^{N_j}\phi_j\prod_{e}^{N_e}\gamma_e \tag{1}$$

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.

The junction factors encode the likelihood of the label of 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. We give the terms for these five types in Figure 4. Four-junctions are handled in a similar manner. A prohibitively high penalty is set for invalid junctions, such as one edge leading into the junction and none leading out.

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 impose a strong penalty ($\rho_{inconsistent} = e^{-1}$) 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 ($\rho_{floating} = e^{-0.25}$) for a vertical region that is entirely surrounded by another vertical region or by sky (which would imply that the former is floating). The unary region likelihood is computed as the mean geometric class confidence over the region (S1 in Table 1). To write one surface fac-



Segmentation Interpretation 1 Interpretation 2

Figure 5. Examples of CRF junction factors (top row), defined over each junction, and surface factors (bottom row), defined over each boundary, for two different figure/ground labelings of a segmentation. Left side of arrows indicates foreground, and dashed lines indicate non-occlusion boundaries. To improve clarity, we omit the data term in the likelihood expressions here.

tor 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 the region with surface label r_i.

In Figure 5, we show the factor terms for example segmentation and line labelings. Note that the factor graph, when given a valid labeling and excluding the surface terms, decomposes 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 5 for implementation details).

Even approximate max-product inference over our model is intractable due to the high closure penalties, but we can efficiently obtain "soft-max" likelihood estimates. To do this, we combine the sum-product algorithm of Heskes et al. [9] (based on the Kikuchi free energy) with the mean field approximation suggested by Yuille [31] (raise each factor to the $\frac{1}{T}$, with $T = 0.5$ in our experiments).

4. 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. 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 [1], guaranteeing a true hierarchy. We threshold the hierarchy to provide our next initial segmentation.

5. Implementation Details

We train and test our method on the Geometric Context dataset [13], consisting of a wide variety of scenes including beaches, fields, forests, hills, suburbs, and urban streets.

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 label the occlusion relationships of adjacent regions. We assigned ground truth to 100 images: 50 for training and 50 for testing. Examples of the ground truth can be seen in Figure 8. A medium-complexity image will typically contain 10-15 objects according to our ground truth labels.

Training. 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 [5], with 20 16-node decision trees as weak learners. This classification method provides good feature selection and probabilistic outputs. For pairwise cues, we simply concatenate the unary cues (Table 1) for the two boundaries and add cues for continuity (relative angle of the two adjacent boundaries) and boundary length (in pixels). Since cues such as depth and 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 (an average of roughly 300 and 100 regions per image, 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 object detection cascade algorithms [29]. 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 tradeoff between the number of regions and how well the true object regions are preserved (0.6 in our experiments). To train the boundary classifiers after the first iteration, we transfer the ground truth from the initial oversegmentations to the current segmentations by labeling each region as the object that occupies the largest percentage of its pixels.

In our experiments, we set the surface factor unary term

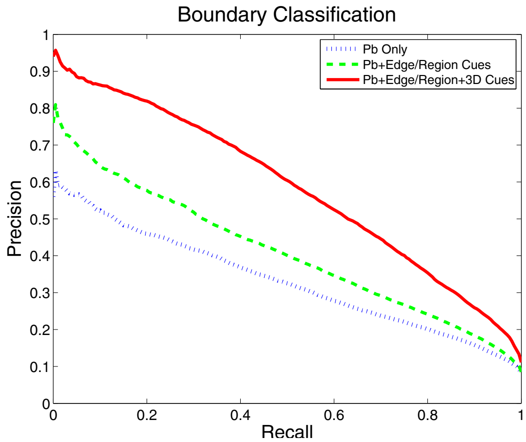


Figure 6. Precision-recall curve for classifying whether a boundary is an occlusion boundary in the first iteration. These results show that 3D cues are important for occlusion reasoning.

by combining, in a linear logistic model, two likelihood estimates: (1) the multiple segmentation estimate from Hoiem et al. [13]; 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.

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 full junction factor terms. In each additional iteration, we perform inference over the full model. The algorithm terminates when no regions are merged in an iteration (typically after a total of 4 or 5 iterations). In our Matlab implementation, our algorithm takes about four minutes for a 600x800 image on a 64-bit 2.6GHz Athalon running Linux, including about 70 seconds for Pb [17] and about 25 seconds for the surface estimation algorithm [13].

6. Experiments

We quantitatively evaluate our method in terms of boundary classification, figure/ground classification, and overall segmentation accuracy on 50 test images. We also provide several qualitative results, showing the recovered object boundaries and estimated depth maps.

Boundary Classification. In Figure 6, we show the precision-recall curve for detecting whether a boundary in the initial oversegmentation is an occlusion boundary using only Pb [17], after adding region and boundary cues, and using all cues. Our results show that the 3D cues are valuable for occlusion reasoning. In computing the precision and recall, boundaries are weighted by length in pixels. For the Pb result, the precision-recall curve is generated by ranking boundaries according to Pb confidence.

	Edge/Region Cues	+ 3D Cues	with CRF
Iter 1	58.7%	71.7%	–
Iter 2	65.4%	75.6%	77.3%
Final	68.2%	77.1%	**79.9%**

Table 2. Figure/ground labeling accuracy results for using edge/region cues only, all cues (including 3D cues), and after performing inference using our CRF model (only unary likelihoods were used in the first iteration).

Figure/Ground Classification. In Table 2, we report the figure/ground classification accuracy. Accuracy is computed over all true occlusion boundaries, including those which are incorrectly classified as non-boundaries in testing. Our accuracy improves in each iteration, as increasingly refined segmentations offer better spatial support for occlusion reasoning. Our final accuracy of 79.9% is noteworthy, considering that on the BSDS dataset [18] the algorithm of Ren et al. [19] achieves figure/ground accuracy of 78.3% using *manual* segmentations or 68.9% with automatically computed boundaries. Our high accuracy is due to our integrated reasoning over boundaries and 3D surfaces.

Overall Segmentation Accuracy. We measure the accuracy of a segmentation in terms of its "conservation" and "efficiency". We report conservation as the pixel error according to the ground truth segmentation, and the efficiency as $\log_2 \frac{N_{objects} - N_{missed}}{N_{regions}}$. Here, $N_{objects}$ is the total number of objects, and N_{missed} is the number of objects that cannot be recovered from the current segmentation. The difference is taken so that efficiency does not increase when two ground truth objects are merged. We show a scatter plot of the iteration 1, iteration 2, and final quantitative results in Figure 7. Examples of ground truth and results, annotated with conservation and efficiency scores, are shown in Figure 8. In Figure 9, we show results for a variety of other images, together with estimated depth maps. See Section 2 for details on how we compute the depth maps from occlusion and surface estimates. Our depth maps are not quantitatively accurate, since camera viewpoint and focal length are unknown, but they give a good qualitative sense of the depth of the scene.

To provide further validation, we compare segmentations to two baseline methods: (1) connected components on the most likely surface labels from Hoiem et al. [13]; and (2) a recent version of the NCuts segmentation algorithm [6]. In Table 3, we show that our algorithm achieves greater mean conservation and efficiency than both of the others. Also, only our algorithm provides figure/ground labels.

Object Pop Out. In Figure 10, we show a few examples of regions automatically found by our system that could serve as an initial stage for unsupervised object discovery [21].

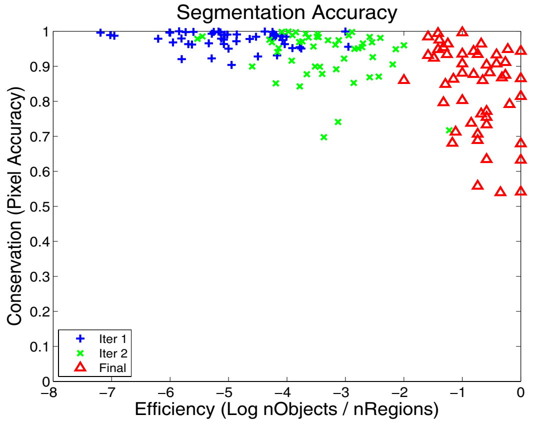


Figure 7. Scatter plot of "efficiency" vs. "conservation" for each test image as the segmentations become increasingly coarse. A perfect segmentation would have $\log_2$ efficiency of 0 and conservation of 1.

	Conservation	$\log_2$ Efficiency
Our Algorithm	**83.7%**	**-0.8**
Surface-Based	82.4%	-1.4
Ncuts	81.7%	-1.2

Table 3. We outperform segmentations using only surface labels and an image-based normalized cuts algorithm [6] by using both surface and image cues together with boundary reasoning.

7. Conclusions

We believe that we have made much progress on an extremely difficult problem that is crucial to scene understanding. The key is to reason together about the segmentations and figure/ground relationships, taking advantage of both 2D and inferred 3D cues in the image. Further progress can be made by including object-specific information or by extending the current color and texture similarity measures to more general measures of co-occurrence in natural scenes. Acquisition of large training sets, ideally by automatic assignment of ground truth using stereo or video cues, would allow large improvements through more effective learning. We hope that our work inspires others to perform further research on single-image occlusion reasoning and the broader 3D scene understanding problem.

References

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

[2] I. Biederman. On the semantics of a glance at a scene. In M. Kubovy and J. R. Pomerantz, editors, *Perceptual Organization*, chapter 8. 1981.

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

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

[5] M. Collins, R. Schapire, and Y. Singer. Logistic regression, adaboost and Bregman distances. *Mach. Learn.*, 48(1), 2002.

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

[7] J. Gibson. The perception of surface layout: A classification of types. Unpublished "Purple Perils" essay, Nov 1968.

[8] A. Guzman. Computer recognition of three-dimensional objects in a visual scene. Tech. Rep. MAC-TR-59, MIT, 1968.

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

[10] D. Hoiem. *Seeing the World Behind the Image: Spatial Layout for 3D Scene Understanding*. PhD thesis, Robotics Institute, Carnegie Mellon University, August 2007.

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

[12] D. Hoiem, A. A. Efros, and M. Hebert. Putting objects in perspective. In *CVPR*, 2006.

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

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

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

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

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

[18] 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.

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

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

[21] 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.

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

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

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

[25] E. Sudderth, A. Torralba, W. T. Freeman, and A. Wilsky. Depth from familiar objects: A hierarchical model for 3D scenes. In *CVPR*, 2006.

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

[27] A. Torralba. Contextual priming for object detection. *IJCV*, 53(2), 2003.

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

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

[30] 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.

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

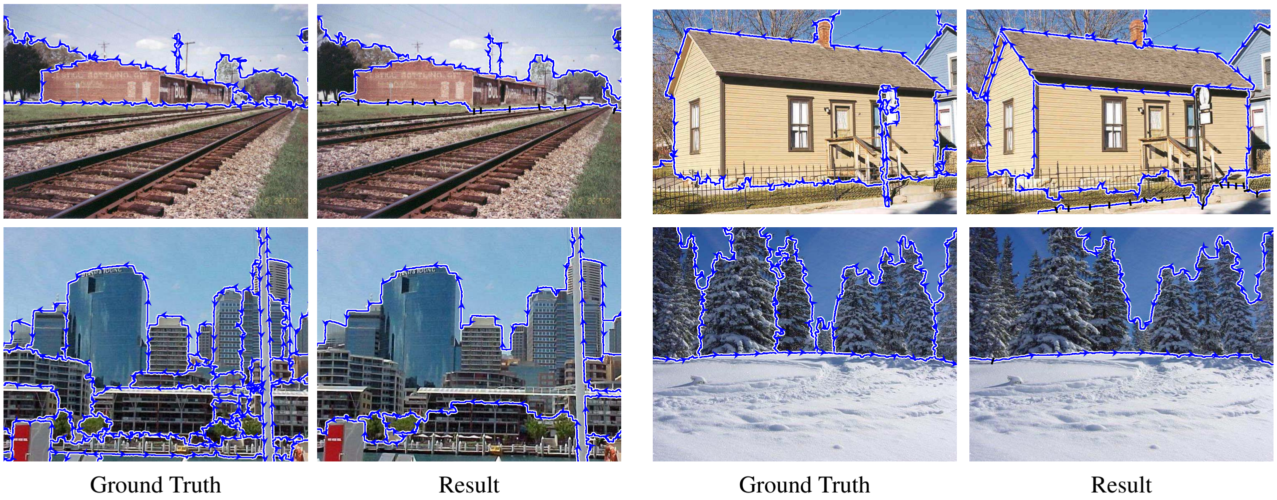


Figure 8. Ground truth and final occlusion boundary results (see Figure 9 for legend). From top-left clockwise, the efficiency and conservation values are (-0.22, 0.95), (-1.59, 0.93), (0, 0.68), (-0.35, 0.54). The lower-left image is the result with the lowest pixel accuracy out of the test images with ground truth.

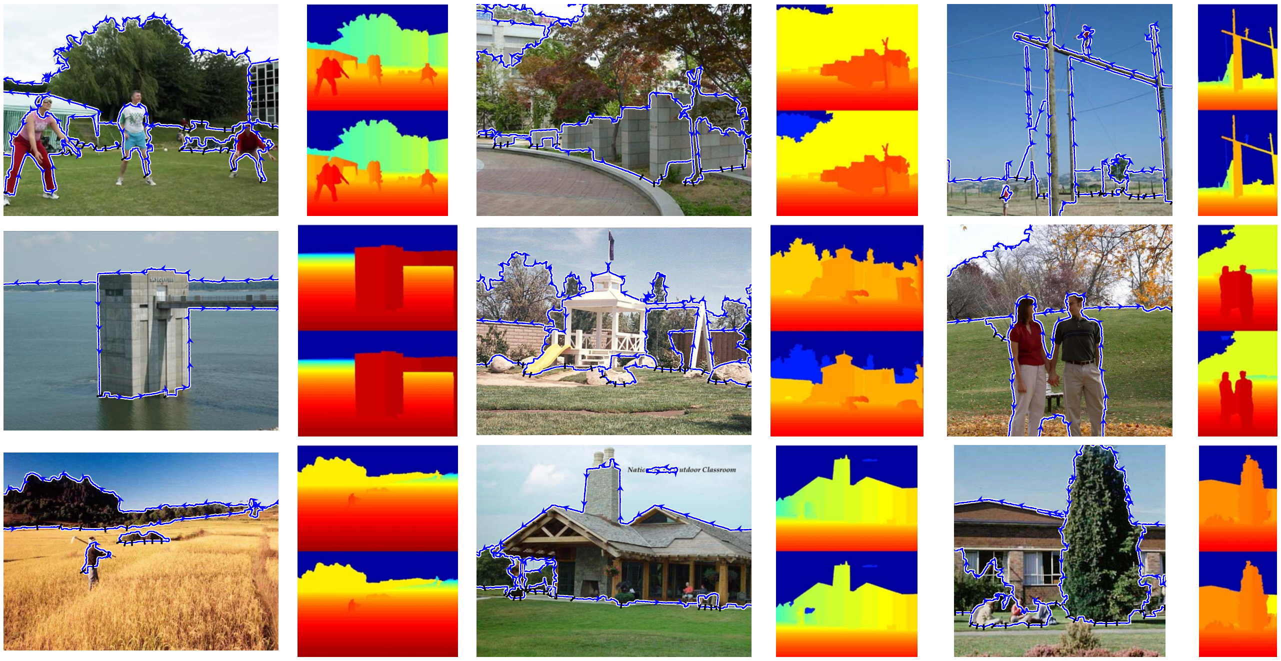


Figure 9. Examples of boundary and depth map results. Blue lines denote occlusion boundary estimates, arrows indicate which region (left) is in front, and black hatch marks show where an object is thought to contact the ground. On the right, we display the minimum and maximum depth estimates (red = close, blue = far).

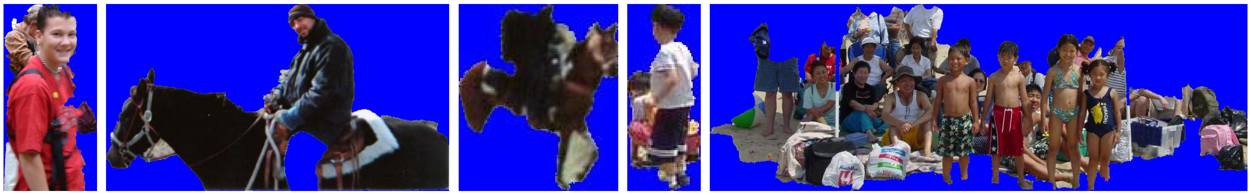


Figure 10. Object popout. We show five out of the fifteen most "solid" regions in the Geometric Context dataset. Our algorithm often finds foreground objects, which would be helpful for unsupervised object discovery [21].