Closing the Loop in Scene Interpretation

Derek Hoiem
Beckman Institute
University of Illinois
dhoiem@uiuc.edu

Alexei A. Efros
Robotics Institute
Carnegie Mellon University
efros@cs.cmu.edu

Martial Hebert
Robotics Institute
Carnegie Mellon University
hebert@ri.cmu.edu

Abstract

Image understanding involves analyzing many different aspects of the scene. In this paper, we are concerned with how these tasks can be combined in a way that improves the performance of each of them. Inspired by Barrow and Tenenbaum, we present a flexible framework for interfacing scene analysis processes using intrinsic images. Each intrinsic image is a registered map describing one characteristic of the scene. We apply this framework to develop an integrated 3D scene understanding system with estimates of surface orientations, occlusion boundaries, objects, camera viewpoint, and relative depth. Our experiments on a set of 300 outdoor images demonstrate that these tasks reinforce each other, and we illustrate a coherent scene understanding with automatically reconstructed 3D models.

1. Introduction

Scene understanding requires the coordination of many different tasks – occlusion reasoning, surface orientation estimation, object recognition, and scene categorization, among others. How can we even begin to sort them out? Grappling with this issue, Marr proposed, in 1978, to organize the visual system as a sequential process, producing increasingly high-level descriptions of the scene: from a low-level primal sketch to a $2\frac{1}{2}$D sketch of surfaces to a full 3D model of the scene [14]. Unfortunately, with this model, a flaw in early processing can ruin the entire interpretation. Barrow and Tenenbaum [1] extended Marr's idea of geometric sketches to a general representation of the scene in terms of *intrinsic images*,[1] each a registered map describing one characteristic of the scene. But in contrast to Marr's feed-forward philosophy, Barrow and Tenenbaum proposed that the entire system should be organized around the recovery of these intrinsic images, so that they serve, not as a pre-process, but as an interface between the interdependent visual tasks. Their key idea is that the ambiguities of the scene can be resolved only when the many visual processes are working together.

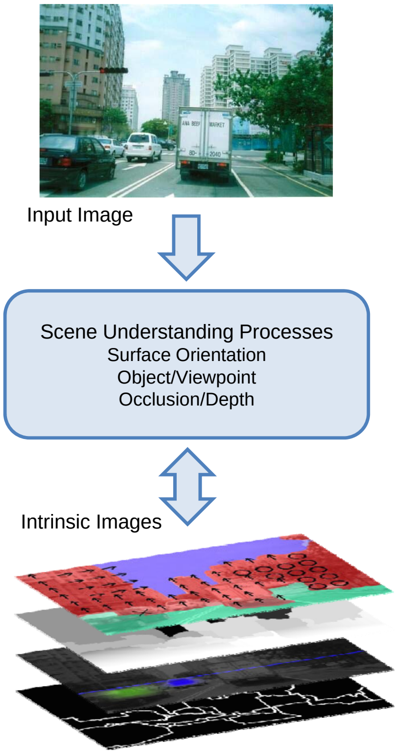
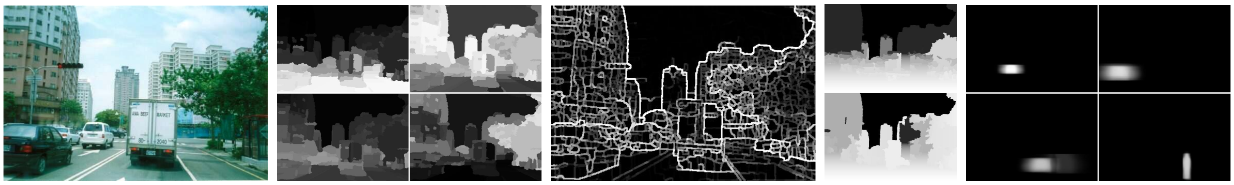


Figure 1: We propose a simple framework for integrating disparate scene understanding processes using maps of scene characteristics as an interface.

Recently, researchers have made much progress in recovering scene properties, such as primal sketch [6], surface orientations [10], depth [21], illumination [23], and occlusion boundaries [11, 19]. In the tradition of Marr, such advances have typically been applied as part of a feed-forward system. We believe that it is time to revisit the ideas of Barrow and Tenenbaum – to see how these tasks can work together to achieve something greater than the sum of their parts.

In this paper, we propose a simple and flexible frame-

[1] The popularity of the reflectance and illumination images from Barrows and Tenenbaum's 1978 work has led, over the years, to a redefinition of the intrinsic image problem as reflectance/illuminant factorization. However, in this paper, we will use the term in its original meaning [1].



(a) Input Image (b) Surfaces (c) Occlusion (d) Depth (e) Objects

Figure 2: Examples of intrinsic images estimated from the image (a) in the first iteration. In (b), we show four of the surface confidence maps (brighter is higher confidence); clockwise, from upper-left: "support", "vertical planar", "vertical porous", "vertical solid". In (c), we show the confidence map for occlusion boundaries (bright indicates occlusion likely). In (d), we show upper and lower estimates of depth (log scale, brighter is closer). In (e), we show four of the object intrinsic images. Each is a confidence map indicating the likelihood of each pixel belonging to an individual object (cars or pedestrians in this case).

work (Figure 1) for integrating various scene characteristics. We organize our system as a set of processes that each outputs a set of intrinsic images. Iteratively, each visual task interacts with the others through cues computed from the intrinsic images and outputs its own intrinsic images in return.

Much recent work in computer vision has focused on contextual relationships. For example, Oliva and Torralba [16, 17] propose a scene gist representation that can be used to coarsely estimate scene depth and camera viewpoint and to improve object recognition [15]. Sudderth et al. [22] infer the depth of points in the image using detected objects. Leibe et al. [13] and Ess et al. [4] model the relationship between objects and scene geometry. Several others model object-object interactions (Rabinovich et al. [18] is a recent example).

Our framework has several advantages. First, it is synergistic, allowing all processes to benefit from their combined interaction. By contrast, much existing work in context is feed-forward. For instance, Hoiem et al. [8, 11] use estimates of surface orientations to improve object detection and occlusion estimation, but the original surface estimates not are improved. Second, our framework is modular, allowing a new process to be inserted without redesigning the entire system. Systems that define contextual relationships symbolically and perform inference over graphical models (e.g., [9, 12]) usually cannot easily accommodate new types of information. Third, by allowing one process to influence another through cues, rather than hard constraints (as in the original Barrow and Tenenbaum paper), the framework is robust and not subject to researcher-designed assumptions.

We test our framework by integrating several scene analysis algorithms from Hoiem et al. that describe surfaces [10] ("surface layout"), detect objects and infer viewpoint [9] ("objects in perspective"), and recover object occlusion boundaries and estimate depth [11] ("occlusion"). We treat each of these algorithms as a component that takes as input the raw image and the intrinsic images from the other algorithms and outputs its own set of intrinsic images. Note that in this paper we do not attempt to improve the low-level cues or inference mechanisms of the individual algorithms. Instead, our goal is to provide a more accurate and coherent scene interpretation by closing the feedback loop among them.

We analyze the effectiveness of our framework and the contextual cues on the Geometric Context dataset [8] of 300 outdoor images from a wide variety of scenes. Our results demonstrate modest quantitative improvement in surface estimates and object detection, as well as substantial qualitative improvement in occlusion estimates. Finally, we demonstrate the scene understanding of our system with a new automatic single-view 3D reconstruction algorithm, which is the first to model foreground objects.

2. Intrinsic Image Representation

The intrinsic images, displayed in Figure 2, serve as an interface between the various scene understanding processes. As proposed by Barrow and Tenenbaum [1], each intrinsic image is an image-registered map of one scene characteristic. Our intrinsic images differ from those of Barrow and Tenenbaum in that they reflect the confidences of the estimates, either by representing the confidences directly, as with the surfaces, or by including several estimates, as with the depth.

Surfaces. The surface intrinsic images consist of seven confidence maps for "support" (e.g., the ground), vertical planar facing "left", "center", or "right" (e.g., building walls), vertical non-planar "porous" (e.g., tree leaves), vertical nonplanar "solid" (e.g., people), and "sky". Each image is a confidence map for one surface type indicating the likelihood that each pixel is of that type.

We compute the surface intrinsic images using the surface layout algorithm [10]. In this algorithm, the image is partitioned several times into multiple segmentations. Image cues are then computed over each segment, and a boosted decision tree classifier [2] estimates the likelihood that the segment is valid (does not contain several different labels) and the likelihood of each possible label. These likelihoods are then integrated pixel-wise over the segmentations to provide several confidence maps. We modify the original surface layout algorithm by storing the multiple segmentations and augmenting the cue set with the contextual cues from the other processes.

Occlusions and Depth. One intrinsic image is computed for occlusion boundaries, indicating the likelihood that a pixel lies on the occlusion boundary of a free-standing object (a physical structure that is entirely self-supporting). We compute the occlusion boundaries using the publicly available code from Hoiem et al. [11]. This algorithm uses a CRF model with unary potentials estimated by boosted decision tree classifiers to estimate the likelihood that the region on either side occludes for each boundary in a hypothesized segmentation. Then, regions that are unlikely to have an occluding boundary between them are merged, and the cues and boundary estimates are updated for the new segmentation. As this method is already iterative, we perform one iteration of the occlusion estimation each time the other intrinsic images are updated and augment the original cues with the object and viewpoint information.

The original occlusion algorithm also outputs three estimates of depth (with a fixed guess of intrinsic parameters that determine scale), which we use as intrinsic images. Each is a separate estimate of depth in log scale computed directly from the current surface and boundary estimates based on assumptions that ground is horizontal and other objects are vertical. See [11] for details.

Objects and Camera Viewpoint. Each hypothesized object is represented by a confidence map, indicating the likelihood that a pixel is part of the object times the likelihood that the object exists.

The objects in perspective algorithm [9] outputs a set of hypothesized objects, along with a probability distribution over potential bounding boxes. We use this distribution (using the Dalal-Triggs local detector [3]), along with an expected mask of the object class, to compute the likelihood that each pixel is part of an object instance. We compute the expected mask of an object by averaging the masks of manually segmented objects in LabelMe [20]. The result is an "object map" for each object that is detected with confidence greater than some threshold. Objects at lower image positions are assumed to be closer to the viewer, so if two objects overlap, the confidences of the further object pixels are multiplied by one minus the confidences of the closer object pixels (loosely modeling object occlusion). The sum of the object maps over a pixel yields the likelihood that the pixel is generated by *any* detectable object. Intrinsic images are included for each hypothesis that passes a confidence threshold (5% in our experiments).

The camera viewpoint is a two-parameter fit to the ground plane (the plane that supports most objects of interest) with respect to the camera, corresponding loosely to the horizon position and the camera height. The objects in perspective algorithm captures the relationship between the image size of grounded objects and the camera viewpoint and directly outputs the most likely camera viewpoint.

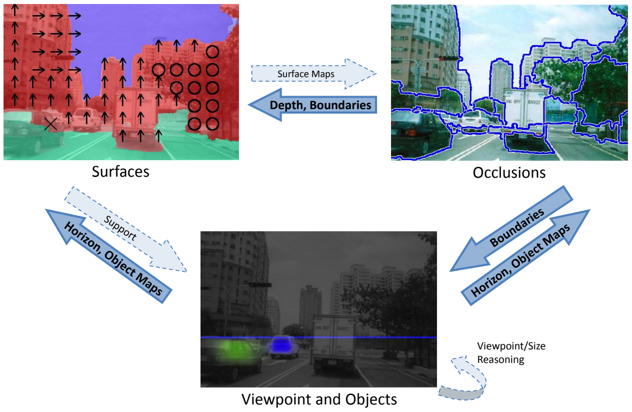


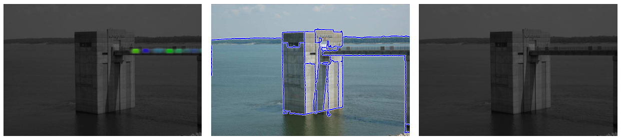
Figure 3: Contextual symbiosis. We show our final estimates for surfaces, occlusion boundaries, viewpoint, and objects and illustrate the interplay among them. The dotted arrows contain contextual relationships modeled by the previous work of Hoiem et al. [9, 11], while the solid arrows denote new cues proposed in this paper. For surfaces: green=support, red=vertical, blue=sky; arrows=planar orientation, X=solid, O=porous. Occlusion boundaries are denoted with blue/white lines. Objects are shown as overlaid individually colored confidence maps of object extent, with the blue line denoting the estimated horizon.

3. Contextual Interactions

Here, we describe how the processes can interact using the intrinsic images as an interface between them. The original Hoiem et al. algorithms [9, 11] incorporate interactions from surfaces to objects and from surfaces to occlusions. We summarize those and propose several new contextual interactions to close the feedback loop. See Figure 3 for an illustration.

Surfaces and Objects. An object tends to correspond to a certain type of surface. For instance, the road is a supporting surface, and a pedestrian is a vertical, non-planar solid surface. In addition, many objects, such as cars and pedestrians tend to rest on the ground, so a visible supporting surface lends evidence to a hypothesized object. As Hoiem et al. [9] showed, these relationships between objects and surfaces can be used to improve object recognition.

In this paper, we also allow object detections to improve surface estimation. We provide object-based cues for the surface classifier by summing pixel confidences for each class of object (cars and pedestrians in our experiments) and computing the mean over each surface region. The object recognition algorithm also outputs a viewpoint estimate which can be used to further improve surface estimation by representing the differences between the horizon position and the top and bottom (10th and 90th percentile of row-position) of the surface region.



Objects (2nd) Occlusion (2nd) Objects (Final)

Figure 4: Example of the influence of occlusion on object estimates. Before considering occlusion information in the second iteration, pieces of the crosswalk are mistaken for cars (left). During the occlusion reasoning, however, it is determined that the crosswalk is a single structure, and the false detections are discarded.

Surfaces and Occlusions. Occlusion boundaries often occur at the boundary between neighboring surface regions of different types. Further, the surface types are often a good indicator of the figure/ground label (e.g., the sky is always occluded). For this reason, cues based on surface estimates can greatly aid occlusion estimation. Additionally, the surface images, together with occlusion boundaries and camera viewpoint, are used to estimate depth, as in the original occlusion algorithm [11].

In this paper, we also model how the boundary and depth estimates can benefit surface estimation. The boundary estimates allow better spatial support. In each segment produced by the multiple segmentation algorithm, we look up the confidence of the most likely internal boundary which helps determine whether a segment is likely to correspond to a single label. Also, the average depth and a measure of the slope in depth from left to right is computed each segment (for all three depth estimates). The average depth may help determine the label of the segment since appearance characteristics vary with distance (e.g., the texture in foliage is lost at a distance). The slope in depth may help determine whether a planar segment faces the left, center, or right of the viewer.

Objects and Occlusions. Object detections can aid occlusion reasoning by helping to determine whether neighboring regions are part of the same individual object. To represent this, we first compute the mean and max confidences for each individual object for each region. As cues, we then compute: the sum (over objects) of the mean confidences (giving total object likelihood); the sum of the absolute difference of the mean confidences between two regions (an overall measure of the object confidence difference); the confidence of the most likely individual object within each region; and the maximum absolute difference between individual object confidences.

Likewise, occlusion reasoning can help remove false object detections by showing them to be part of a larger structure. For example, pieces of the crosswalk in Figure 4 individually appear to be cars (and are consistent in viewpoint) but are discarded when found to be part of the larger cement structure. In our implementation, we simply remove object hypotheses if its soft mask is inconsistent with the occlusion estimates. More specifically, regions from the occlusion result are assigned to an object if they overlap at least 50% with the object bounding box; if the area of the regions assigned to an object is less than 25% of the expected object area, the object candidate is removed. We note that this deviates from our general principal of "soft" consistency and sometimes causes small pedestrians to be incorrectly discarded.

```
TRAINING
Initialize:
  • Get multiple segmentations for each training image
  • Estimate horizon
  • Perform local object detection
For iteration $t = 1..N_t$:
  1. Train and apply surface estimation
     (a) Compute features for each segment (using results of (2),(3) from iterations $1 \ldots t-1$)
     (b) Train surface classifiers and compute surface confidences with cross-validation
  2. Apply object/viewpoint/surface inference (using result of (1) from iteration $t$ and (3) from $t-1$)
  3. Train and apply occlusion reasoning algorithm (using results of (1), (2) from iteration $t$)
     (a) Train on hold-out set
     (b) Perform occlusion reasoning on cross-validation images
```

Figure 5: Iterative training algorithm for combining surface, occlusion, viewpoint, and object information. Training and testing is performed on the Geometric Context dataset. The holdout set of 50 images used to train the surface segmentation algorithm is used to train the occlusion reasoning, and the remaining 250 images are used for testing (using five-fold cross-validation for the surface estimation). N_t=3 in our experiments.

4. Training and Inference

Training and inference are performed in a simple iterative manner, cycling through the surface estimation, object detection and viewpoint recovery, and occlusion reasoning. We outline our training and inference algorithm in Figure 5. Each of our algorithms are evaluated using the Geometric Context dataset. The first fifty images are used for training the surface segmentation and occlusion reasoning. The remaining 250 are used to test the surface, object, viewpoint, and occlusion estimators. The surface classifiers are trained and tested using five-fold cross-validation.

In training and testing the surface classifiers, the multiple segmentations are computed once. In each iteration after the first, the cues for each segment are updated with information gleaned from the latest object, viewpoint, and occlusion estimates. The object/viewpoint inference uses the surface estimates from the current iteration and the occlusion information from the previous iteration (starting in the second iteration). The occlusion algorithm uses the latest surface, object, and viewpoint estimates. The first two iterations of the occlusion algorithm correspond to the first two iterations of the original algorithm, with additional cues from the latest surface, object, and viewpoint estimates. In the third iteration, the occlusion algorithm re-iterates until convergence. In all other respects, the training and testing for the three algorithms is implemented as described in the work of Hoiem et al. [8, 9, 11].

5. Experiments

We applied the training and inference procedure described in the previous section to the Geometric Context dataset. We show examples of final results in Figure 6. In the object results here and elsewhere, only objects that pass a preset confidence threshold are shown (threshold corresponds to 0.5 false positives per image). In this section, we discuss the improvement in each type of estimation. Our analysis includes qualitative assessment, inspection of the decision tree classifiers learned in the surface and occlusion estimation, and quantitative performance comparison. In the decision tree learning, early and frequent selection of a cue indicates that the cue is valuable but does not necessarily imply great value beyond the other cues, as there may be redundant information.

Surfaces. The decision tree learning indicates that the boundary likelihood from occlusion reasoning is the most powerful cue for the segmentation classifier, as it is the first and most frequently selected. For the "solid" classifier, the pedestrian confidence value from the object detection is the first feature selected. The learning algorithm determines that regions with high pedestrian confidence are very likely to be solid, but regions without pedestrian confidence are not much less likely (i.e., all pedestrians are solid, but not all solids are pedestrians). Our measure of depth slope from the occlusion reasoning is used frequently by the planar "left" and "right" classifiers.

We also find that the position cues relative to the estimated horizon position are used overall twice as frequently as the absolute position cues. In a separate experiment, in which we train and test surface classification with manually assigned (ground truth) horizon estimates, we find that the main (support, vertical, sky) classification and subclassification of vertical (left, center, right, porous, solid) each improve by 2% (from 87% to 89% and 61% to 63%, respectively). Thus, knowledge of the horizon is an important cue, but the potential improvement in surface estimation by improving the horizon estimate is limited.

We find a modest quantitative improvement in surface classification with the inclusion of object and occlusion information. Using the surface layout algorithm [7] by itself, the main classification accuracy is 86.8% and subclassification accuracy is 60.9%. The accuracy improves by roughly 1% to 87.6% and 61.8% once the contextual information from the objects, occlusions, and depth is considered. For the main classification ("support" vs. "vertical" vs. "sky"), this difference is statistically significant ($p < 0.05$), but for the subclassification (subclasses within "vertical") it is not ($p > 0.05$). With closer inspection on an image-by-image basis, of the 18% of images that change in main class pixel error by more than 5%, 74% improve. Thus, while large changes are rare, changes made after considering the new object and occlusion information are much more likely to improve results than not.

	Car	Ped
Dalal-Triggs 2005 (Initial)	41%	54%
Hoiem et al. 2006 (Iter 1)	41%	66%
This paper (Final)	51%	61%

Table 1: Detection rates at 1 false positive per image for cars and pedestrians on the Geometric Context dataset. We compare results from the local detector [3] (top row), the putting objects in perspective algorithm [9], and our iterative algorithm.

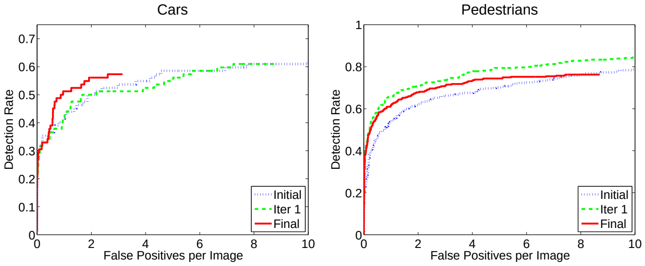


Figure 7: ROC curves for initial Dalal-Triggs classification, after one iteration (equivalent to the "Putting Objects in Perspective" algorithm [9]), and the final result.

Objects. As in [9], we use the Dalal-Triggs [3] object detector to supply object hypotheses and confidences based on local image information. In Table 1, we report the detection rate at 1 false positive per image for cars and pedestrians in the Geometric Context dataset, ignoring cars smaller than 24 pixels tall and pedestrians smaller than 48 pixels tall (these are the minimum sizes of the respective detector windows). In total, the test portion (excluding the occlusion training set portion) of the dataset contains 82 cars and 379 pedestrians.

When viewpoint and surfaces are considered (result of the first iteration, equivalent to the objects in perspective algorithm [9]), pedestrian detection improves considerably. When occlusion information is considered (later iterations) car detection improves but pedestrian detection drops slightly, likely due to the difficulty of maintaining occlusion boundaries for distant pedestrians (see Figure 7). Along with many false positives, 11% of true pedestrian and 8% of true car detections are discarded by the occlusion-based filtering. Overall, the car detection improves by 10% and the pedestrian detection by 7% when considering surface,

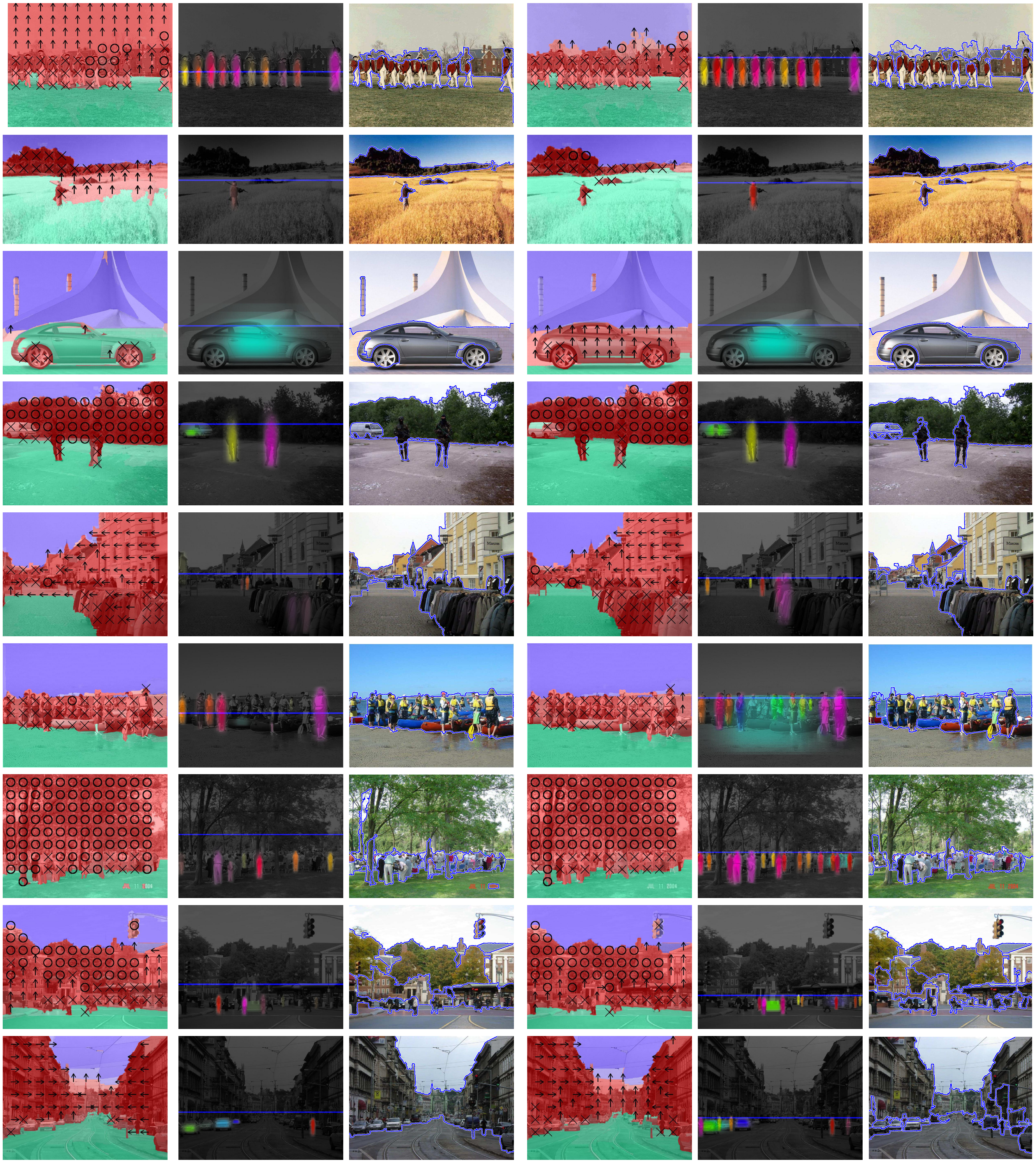


Figure 6: In each row, we show the results of the three original algorithms and the results when they are integrated with our framework. Each process achieves higher performance when they work together. Rows 1 and 2: the occlusion estimates help improve mistakes in the surface labels. Row 3: the detected car allows the surface labels and occlusion estimate to improve. Row 4: the soldiers need better camouflage as the algorithm is able to identify and segment them away from the background. Row 5: false detections on the coat rack are eliminated due to the occlusion information. Row 6: The contextual inference results in more pedestrians being detected (with a false positive car). Rows 7-9: A reasonably good job is done in complicated scenes.

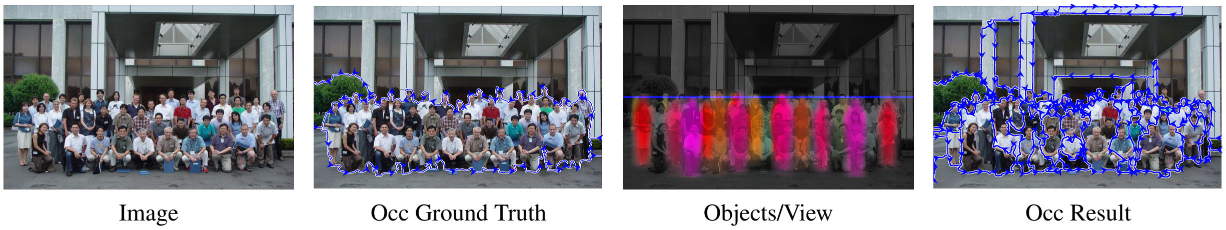
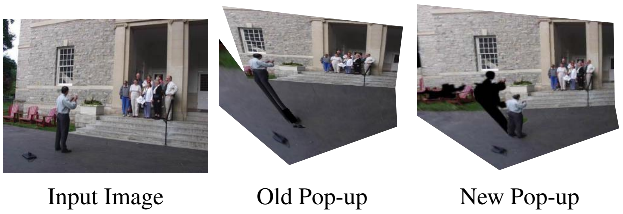


Image Occ Ground Truth Objects/View Occ Result

Figure 8: By reasoning together about objects and occlusions, we are sometimes able to find the occlusion boundaries of tightly crowded individuals. In this case, our occlusion estimates are more precise than the ground truth, which leads to an artificially low quantitative measure of improvement.



Input Image Old Pop-up New Pop-up

Figure 9: We show results for the original Hoiem et al. [7] algorithm and our extension of that algorithm to handle foreground objects. In the original algorithm the man is pressed into the ground and the building in back of him. In our new algorithm, the man is correctly pulled away from the background using the occlusion estimates.

viewpoint, and occlusions.

Viewpoint. We evaluate viewpoint estimation based on the horizon position, since it is difficult to obtain ground truth for camera height. Using the mean horizon position as a constant estimate yields an error of 12.8% (percentage of image height difference between the manually labeled horizon and the estimated horizon). This error drops to 10.4% when using a data-driven horizon estimate with the LabelMe training set. During the first iteration, which is equivalent to the objects in perspective algorithm [9], this drops further to 8.5%. Further iterations do not produce a statistically significant change ($p > 0.05$).

Occlusion. A subjective comparison reveals that individual cars and people are correctly delineated much more frequently when object information is considered. Figure 6 contains many such examples. However, we are not able to measure an overall quantitative improvement, due to ground truth labeling of a crowd of people or row of cars as a single object, as shown in Figure 8.

6. Automatic Photo Pop-up with Occlusions

We demonstrate the full scene understanding ability by providing a simple but effective extension of the Hoiem et al. [7] Automatic Photo Pop-up algorithm to handle occluding foreground objects. In the original algorithm, only surface estimates and a horizon position are used to create a simple piecewise planar 3D model of the scene. As a consequence, this method fares poorly in cluttered scenes with foreground objects. The more recent work of Saxena et al. [21] can better handle cluttered scenes, since it is not based on a segmentation into ground/vertical/sky surfaces, but it cannot handle foreground objects because it assumes that the entire visible scene is a continuous surface.

In this section we incorporate occlusion, viewpoint, and object information into the reconstruction. The object detections provide better occlusion boundaries for cars and pedestrians. The occlusion boundaries allow foreground objects to be separated from the background and modeled as individual “billboards”. The viewpoint estimates provide the correct perspective. Overall, we are better able to handle complex scenes.

When the ground-contact points of a region are visible, we can fit a ground-vertical boundary polyline to those points (as in the original photo pop-up algorithm) to model the region with a series of planes. When the ground-contact points are obscured, however, we cannot measure the depth or orientation of the region directly. For the purpose of creating a graphically appealing model, it is better to group such regions with other regions of known depth, rather than to try to model them as separate surfaces. Our solution is to iteratively remove occlusion boundaries between regions of unknown depth and regions of known depth, removing the weakest boundaries first. The strength of a boundary is given by the occlusion intrinsic image. To determine the “support”, “vertical”, “sky” labels needed to create the model, we average surface confidences provided by the surface estimation process and the occlusion process (this latter estimate is not used elsewhere in our intrinsic image framework). In other respects, the 3D scene reconstruction is performed as originally described by Hoiem et al. [7].

We show a comparison of the original Photo Pop-up algorithm and our new algorithm in Figure 9. We display three additional results in Figure 10. The 3D models demonstrate a good understanding of the spatial layout of the scene. Overall, in the 250 images we processed (from the Geometric Context dataset), about 40% of the models were “pretty good” (based on a subjective assessment of whether there are major modeling errors). Hoiem et al. [7] report a success rate of 30% on a simpler subset of the data.

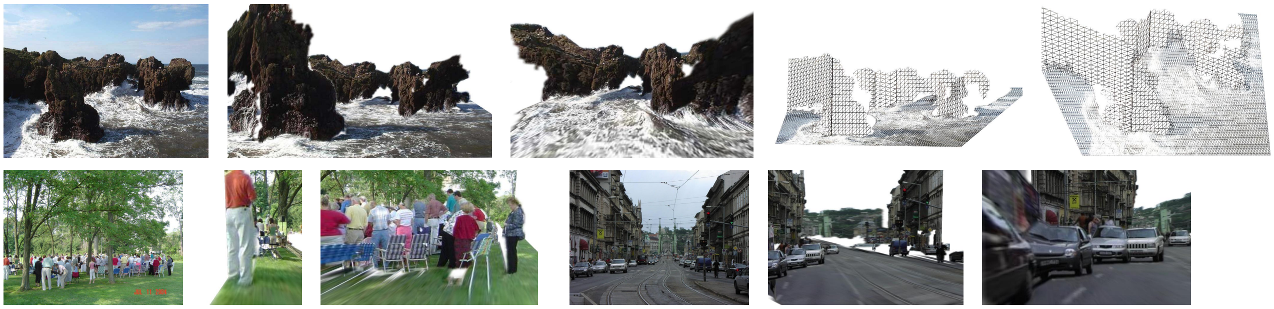


Figure 10: Input images and novel views taken from automatically generated 3D models, using estimated occlusion boundaries to separate foreground objects. In row 1, we show two novel views from the textured 3D model and from similar views in a wireframe model (wires are colored from image). Note how, in the scenes of row 2, pedestrians and cars are correctly segmented from the background.

7. Discussion

Our framework allows many different visual tasks to work together by iteratively relating them through their intrinsic images. Our experiments demonstrate the effectiveness of our approach, indicating that inference over explicit symbolic representations (e.g., graphical models) may not be required. This is important because graphical models, though effective in constrained domains such as object-viewpoint inference [9], often cannot be extended to tractably handle a large number of visual tasks.

With suitable choices of learning algorithms in each of the processes, our framework could provide additional advantages. If the algorithms are able to share features (e.g., [24]), one process could benefit from the structural knowledge of the input space that is learned by other processes. If a linear logistic regression algorithm is used (e.g., Adaboost [5]) and sufficient statistics of the training data are stored, a new visual task (e.g., building detection) could be inserted and used by the existing processes without completely retraining them.

Acknowledgements. We thank Takeo Kanade, Rahul Sukthankar, and Bill Freeman for discussions. Partial funding was provided by Microsoft Graduate and Beckman Postgraduate Fellowships (DH), NSF CAREER award IIS0546547 (AE), and NSF Grant IIS0745636 (MH).

References

[1] H. Barrow and J. Tenenbaum. Recovering intrinsic scene characteristics from images. In *Comp. Vision Systems*, 1978.

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

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[4] A. Ess, B. Leibe, and L. Van Gool. Depth and appearance for mobile scene analysis. In *ICCV*, 2007.

[5] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28(2), 2000.

[6] C. Guo, S. Zhu, and Y. Wu. Primal sketch: Integrating texture and structure. *Computer Vision and Image Understanding*, 106(1):5–19, April 2007.

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

[8] D. Hoiem, A. A. Efros, and M. Hebert. Geometric context from a single image. In *ICCV*, 2005.

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

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

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

[12] S. Kumar and M. Hebert. A hierarchical field framework for unified context-based classification. In *ICCV*, 2005.

[13] B. Leibe, N. Cornelis, K. Cornelis, and L. Van Gool. Dynamic 3d scene analysis from a moving vehicle. In *CVPR*, 2007.

[14] D. Marr. Representing visual information. In *Computer Vision Systems*, 1978.

[15] K. Murphy, A. Torralba, and W. T. Freeman. Graphical model for recognizing scenes and objects. In *NIPS*. 2003.

[16] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001.

[17] A. Oliva and A. Torralba. Building the gist of a scene: The role of global image features in recognition. *Progress in Brain Research*, 155, 2006.

[18] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. In *ICCV*, 2007.

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

[20] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. LabelMe: a database and web-based tool for image annotation. Technical report, MIT, 2005.

[21] A. Saxena, M. Sun, and A. Y. Ng. Learning 3-d scene structure from a single still image. In *ICCV 3dRR-07*, 2007.

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

[23] M. F. Tappen, W. T. Freeman, and E. H. Adelson. Recovering intrinsic images from a single image. *PAMI*, 27(9):1459–1472, Sept 2005.

[24] A. Torralba, K. P. Murphy, and W. T. Freeman. Contextual models for object detection using boosted random fields. In *NIPS*. 2004.