Beyond the line of sight: labeling the underlying surfaces

Ruiqi Guo and Derek Hoiem

Department of Computer Science
University of Illinois at Urbana-Champaign
guo29@illinois.edu,dhoiem@uiuc.edu

Abstract. Scene understanding requires reasoning about both what we can see and what is occluded. We offer a simple and general approach to infer labels of occluded background regions. Our approach incorporates estimates of visible surrounding background, detected objects, and shape priors from transferred training regions. We demonstrate the ability to infer the labels of occluded background regions in both the outdoor StreetScenes dataset and an indoor scene dataset using the same approach. Our experiments show that our method outperforms competent baselines.

1 Introduction

Semantic scene labeling is most often viewed as the problem of labeling pixels according to the depicted object category. If done accurately, such representations provide a good sense of what objects and surfaces are visible. But the goal of vision is to provide information about the entire surroundings, not just what is in the line of sight. Consider the images in Fig. 1. We humans have a strong sense of the locations and extents of the sidewalks, roads, and buildings, even though many of them are largely or even fully occluded. However, if we stick to what we can actually see, huge portions of the scene are unknowable.

In this paper, our goal is to label both visible and occluded regions into background categories. For example, a car pixel should be labeled as "road", "sidewalk", or "building", depending on what is behind it. A few recent efforts have been made in this direction, applying strong domain-specific priors and constraints to enable inference of occluded regions. For example, Geiger et al. [1] infer the full road layout from video, and Hedau et al. [2] and follow-up works infer the full extent of the floor from an image. We are seeking a more general approach that would work for a variety of scenes, both indoor and outdoor, without hand-defined priors. Such a general approach could be used as a default system and adapted to particular domains where appropriate.

We incorporate three basic types of information. First, we classify visible background regions. An occluded patch that is surrounded by road, for example, is likely to be more road. Second, we classify visible foreground regions and apply object detectors to localize common objects, such as cars and pedestrians. Location of foreground objects is predictive of background regions. For example,

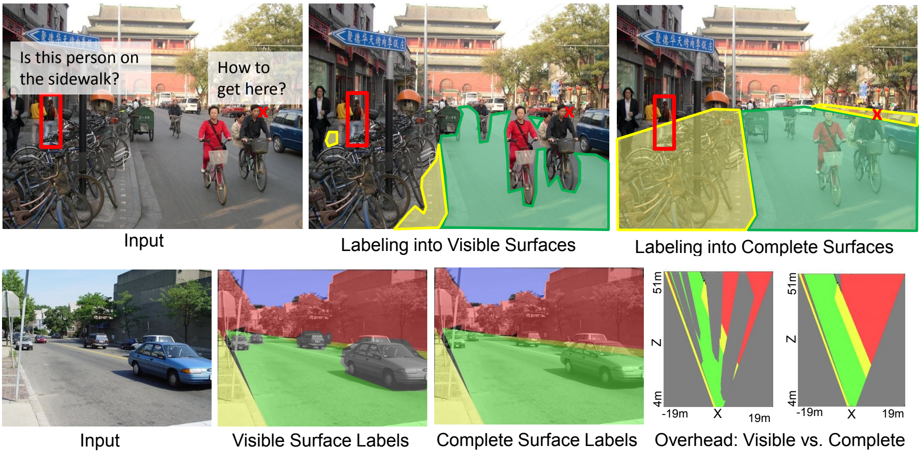

Is this person on the sidewalk?
How to get here?
Input
Labeling into Visible Surfaces
Labeling into Complete Surfaces
Input
Visible Surface Labels
Complete Surface Labels
51m
Z
4m
-19m
X
19m
51m
Z
4m
-19m
X
19m
Overhead: Visible vs. Complete

Fig. 1. Motivation. Scene parsing is often viewed as a problem of labeling pixels into visible categories. But these representations leave much of the underlying scene unknowable. For example, because the woman (top row) is occluded, we cannot determine what she is standing on without inferring that the bicycles are occluding the sidewalk. Likewise, finding paths through cluttered scenes is nearly impossible without reasoning about the underlying surfaces. Below, we project the ground into an overhead view (yellow=sidewalk; green=road; red=blocked by building or trees; gray=unknown). Without more complete estimates of the background, huge portions of the scene are left unknown.

cars are often on the road, and people are often on sidewalks. After obtaining label confidences for the visible regions and object bounding boxes, we predict the complete underlying labels with a feedforward contextual classifier, in the spirit of Tu and Bai's autocontext [3]. Third, we incorporate global scene priors and region shape priors from training images. Often, researchers consider background to be shapeless regions defined by object occlusions, but when we infer the underlying surfaces, their inherent structure remains. For example, sidewalks and roads have characteristic patterns; trees have complex shapes; walls have simple quadrilateral shapes. We use these priors without explicitly defining them, by copying whole polygonal regions from the training set that match our scene according to current likelihood estimates. These transferred regions can then be used to improve label estimates, and they also provide a more structured scene representation in terms of a few polygons (rather than maps of pixel confidences).

To demonstrate the generality of our approach, we perform experiments on the CBCL StreetScenes dataset [4], Hedau et al. [2]'s indoor scene dataset as well as the SUN09 dataset. Each dataset has polygonal labels that can be used to evaluate identification of visible surfaces (as is usually done) or labeling of underlying surfaces (as is our interest). We show that our approach outperforms competent baselines such as classification into visible regions and filling in occluded background regions using nearest visible labels or graph cuts.

1.1 Related Work

Pixel Labeling Pixel labeling problems are nearly always posed as the assignment of labels for the depicted objects or surfaces, as can be seen in the most widely used datasets such as MSRC21 [5], Pascal VOC Segmentation [6], SUN'09 [7], Geometric Context [8], CamVid [9], and the Stanford Background Dataset [10]. Even methods that are focused on scene geometry, such as Hoiem et al.'s surface layout [8] are restricted to labeling visible surfaces. Notable exceptions include recently proposed indoor scene datasets [11,2], Geiger et al.'s work [1] to infer the upcoming road plan from a vehicle-mounted video camera, and Gupta et al.'s blocks world revisited [12]. These approaches regularize visible evidence with simplified models and geometric priors that are suitable for a particular type of scene or application.

We benefit from current methods to label visible foreground and background surfaces. In particular, we use the generic region classifier made available by Hoiem et al. [8]. In contrast to existing work on inferring complete scene layout, we aim to develop a general approach that can be applied to a wide variety of problems. Although our problem definition is not the usual, we can use some existing datasets. People often find it easier to label background regions with polygons that cover the entire surface, rather than drawing around occluding objects. This is one reason that datasets such as LabelMe [13] and CBCL StreetScenes [4] to instruct labelers to draw polygons around the entire background region, ignoring occluders. In fact, Bileschi's thesis [4] reports that labelers often want to label portions of objects, such as tree trunks that cannot be seen at all. It is difficult for people not to see beyond their line of sight. Usually, these datasets "flatten" the label map based on figure/ground ordering, assigning each pixel to one label. We define our task based on the original polygons, predicting, for example, the building pixels that are occluded by a car.

Contextual Methods Many contextual methods have been developed, most often to aid the recognition of "things", such as cars or pedestrians. We build closely on Tu and Bai's auto-context [3], a feedforward mechanism for iteratively classifying a pixel given its features and the current confidences of surrounding pixels. Given initial confidence maps, created by our region classifiers and object detectors, we iteratively re-classify each pixel into a background label. In original auto-context framework the target labels were consistent throughout the process. In our case, we start with confidences for visible surfaces and objects and use them to iteratively re-predict confidences for both visible and occluded regions. A variety of other feed-forward contextual approaches have been reported in the literature (e.g., [14,15,16]). We favor auto-context approach for its simplicity, efficiency, flexibility, and intuitive behavior as a form of belief propagation on an MRF [3].

Region-transfer Methods After we infer as much as possible from visible surfaces and objects, we incorporate global scene priors and shape priors by matching our confidence maps to the labeled ground truth of training images. In this, we relate to several region-transfer methods, such as Maliesiwicz and Efros' visual memex [17], Tighe and Lazebnik's superparsing [18], and other label transfer methods by Liu et al. [19] and Zhang et al [20]. Our method differs from some in that we match purely between our predictions and training ground truth (without considering appearance), so that we are transferring a layout prior, rather than using matched regions as the primary cues for labeling. Our transfer method differs from Liu et al. [19] and Zhang et al. [20] in its simplicity: we simply copy regions in-place, either from one or several training images whose scene-wide labels match well. In some applications, deforming transferred regions may be advantageous, but in our experiments, limiting deformation helped to preserve the original scene layout priors.

1.2 Contributions

Our primary contribution is a general approach to infer underlying surfaces based on estimates of visible surfaces, detected objects, and non-parametric priors on scene layout. We initially attempted a complicated method involving appearance-based image retrieval, geometric reasoning, and MRF inference, but our current, relatively simple approach, performs better. Our approach incorporates existing techniques, such as region classification [8], feed-forward contextual prediction [3], and non-parametric label transfer [19], but it is simple, efficient, and generally applicable. We expect that contextual recognition algorithms and domain-specific scene layout algorithms would benefit from having our more complete scene layout estimates as a starting point for more complex reasoning.

2 Labeling the Complete Scene Surfaces

In this section, we describe our labeling algorithm for complete scene layout. We first describe how to label visible part of the scene, using an off-the-shelf image labeling algorithm and pre-trained object detectors. We then incorporate visible information into a feed-forward contextual prediction to infer the occluded part of the background regions. Next, polygons are matched based on the current label confidences to provide a shape prior for each label. The final pixel prediction incorporates visible surface predictions and the transferred polygons to provide the final complete background labeling. The full pipeline to find the complete scene layout is summarized in Fig. 2. We also describe three baseline methods for inferring occluded background labels.

We define foreground and background categories by hand. For example, on the StreetScene dataset, “building”, “road”, “sidewalk”,“sky”,“store” and “tree” are defined as background, and “car”, “pedestrian”, “bicycle” are foreground.

2.1 Labeling Visible Surfaces and Objects

We apply the region classifier from Hoiem et al. [8] to label pixels into visible foreground and background regions. The image is first over-segmented into superpixels, which are then grouped into multiple segmentations. Color, texture,

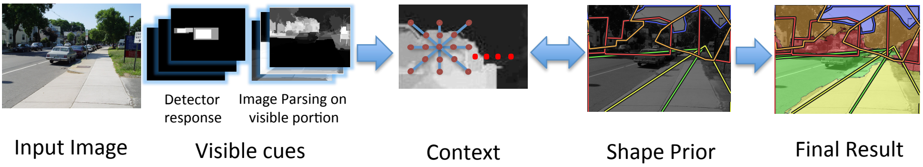


Fig. 2. Given an input image, we predict the labels of visible foreground and background surfaces and estimate object locations using detectors. We then apply a feed-forward contextual method to infer the labels of occluded background regions. We transfer training regions that match current estimates and use them to provide a structured hypothesis and a shape prior which is used to refine the background pixel labels.

edge, and vanishing point cues are then computed for each superpixel. Finally a boosted decision tree classifier combines the prediction and estimates the likelihood of each possible label for each pixel, providing a confidence map for each label.

This algorithm is designed to parse geometric classes and works well for background regions such as building, road and trees, outperforming a more recent algorithm on these classes [18] (see Fig. 5 in the experiment section). We use Felzenswalb et al.'s pre-trained object detector [21] to detect foreground objects such as "cars" and "pedestrians". For each category, we convert the detection bounding boxes to a detection confidence map, where the value of each pixel is set to the maximum of the scores of the detection window that contains that pixel.

2.2 Using Context to Infer Labels of Hidden Surfaces

We base our approach to infer background labels of occluded pixels on the auto-context framework. In the original auto-context paper, the algorithm starts by learning an appearance classifier using image patch features and a boosting algorithm. After applying the trained classifier, the confidence map is then fed as contextual information to train the next classifier. The surrounding label confidences of a pixel are used as features to construct a new training feature set. A new boosted classifier based on the training set is learned and it, in turn, updates the confidence map. The algorithm iterates this process, improving the ground truth likelihood of training labels in each iteration.

We use the same feed-forward idea. However, rather than using appearance features (which would only describe the visible surfaces), we instead directly rely on the outputs of our region classifier and object detectors. As suggested in Tu and Bai [3], we sample the contextual features in the form of sparse, radially distributed points. We use logistic regression for classification instead of boosting as our classifier so that feature computation and applying of the classifier can be done by 2D convolution operations and linear additions over the whole image. This allows us to reduce the testing time to under 1 second per image (400×300

```
Complete Scene Labeling Algorithm
  /* Training */
  Perform object detections and image parsing of visible surfaces in all training images
  Let $\{\mathbf{V_i}\}$ be the confidence maps of region classification and detection for each image $i$
  /* Main loop of auto-context */
  Sample image $i$ and position $x$ from $\mathbf{V}$ and $\mathbf{Y}$ to build training set $\mathbf{S}_0 = \{Y_{ix}, \mathbf{V}_{i,N(x)}\}$,
  where $Y$ is the ground truth map of the complete scene and $N(x)$ is the radial distributed
  neighborhood of position $x$. Train classifiers using logistic regression and obtain parameters $\mathbf{w}_0$.
  For $t = 1 \ldots T$
    Apply previous classifiers with parameters $\mathbf{w}_{t-1}$ to all training images.
    The resulting probability maps of label predictions are $\mathbf{P^{(t-1)}}$.
    Build a new training set, now with feed-forward context $\mathbf{S_t} = \{Y_{ix}, [\mathbf{V}_{i,N(x)}; \mathbf{P}^{(t-1)}_{i,N(x)}]\}$.
    Train classifiers using logistic regression on $\mathbf{S}_t$; the learned parameters are $\mathbf{w}_t$.
  End
  /* Incorporate shape prior */
  For each training image $i$, retrieve polygons from training images that match the $\mathbf{P}^{(T)}_i$.
  Compute shape prior $\{\mathbf{Q}_i\}$ by flattening retrieved polygons.
  Construct final training set $\mathbf{S_{final}} = (Y_{ix}, [\mathbf{V_{i,N(x)}}; \mathbf{P}^{(T)}_{i,N(x)}; \mathbf{Q}_{i,N(x)}])$
  Train the final classifier with $\mathbf{S}^*$ with parameter $\mathbf{w}^*$

  /* Testing */
  For a testing image $k$, do object detections and region classification of visible surfaces.
  Let the result be $\mathbf{V}_k$. Apply classifier $\mathbf{w}_0$ to $\mathbf{V}_{k,N(x)}$, for each position $x$.
  The resulting confidence map of label prediction is $\mathbf{P}^{(0)}_k$.
  For $t = 1 \ldots T$
    Apply classifier $\mathbf{w}_t$ to $[\mathbf{V}_{k,N(x)}; \mathbf{P}^{(t-1)}_{k,N(x)}]$, for each position $x$. The result is $\mathbf{P}^{(t-1)}_k$.
  End
  Compute shape prior $\mathbf{Q}_k$ by retrieving polygons from training images.
  Apply $\mathbf{w}^*$ to $[\mathbf{V_{k,N(x)}}; \mathbf{P}^{(T)}_{k,N(x)}; \mathbf{Q}_{k,N(x)}])$ to get final prediction map $\mathbf{P}^*$
```

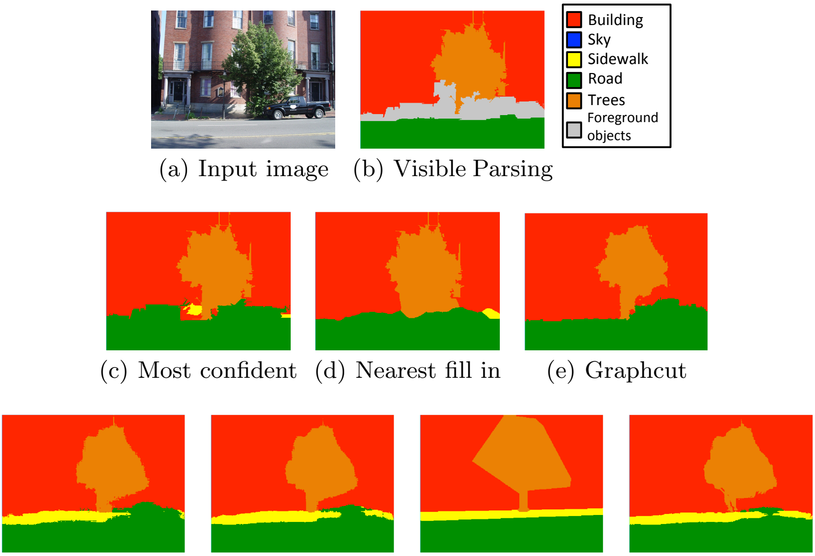
Fig. 3. Outline of our full algorithm. Visible parsing results, object detection, previous label predictions and polygon shape prior are combined to infer the complete label map and polygonal layout.

pixels), whereas the original algorithm which runs at 30 to 70 seconds on a 300 × 200 image.

Due to its discriminative training, auto-context can go beyond simple smoothing. For example, in Fig. 4, it can recover the sidewalk region by looking at the prediction of nearby pixels: if there is building on top of it and road below it, then it is more likely to be sidewalk. An initially missed sidewalk region is recovered after three iterations of auto-context.

2.3 Region Overlay as a Scene and Shape Prior

Intuitively, the overall pattern of labels should be similar to other images observed in the training set, and the pattern of a particular type of label is likely to match some training image quite closely. We operationalize this intuition by finding polygons in the training set that match our current label predictions. These polygons provide a scene prior (because the training image that they come from should have similar labels overall) and a shape prior (because the transferred region maintains its shape). The transferred regions can be used to refine our per-pixel background labels, and the set of transferred regions provide



(a) Input image (b) Visible Parsing

(c) Most confident (d) Nearest fill in (e) Graphcut

(f) Initial context (g) No shape prior (h) Shape Prior (i) Final

Fig. 4. Illustration of baseline methods. (a) shows the original image and (b) gives the parsing map of visible part; (c)-(e) Baseline methods do not go beyond smoothing. (f) and (g) shows incorporating context using classifier does more than a smoothing term, recovering the some sidewalk area. After polygons are retrieved, the flattened shape prior (h) helps to regularize the layout and gives final output (i).

alternative coherent yet compact hypotheses about the hidden portions of the scene.

We find polygons for the each background class separately using the intersection over union criteria. From the previous steps, for each query image k we computed probability map of the background label l as $\mathbf{P}_{l,k}$. We then find the best polygons that matches it and directly lay down the polygons as our region prediction for image k, label l.

The ground truth mask of background label l in training image i is $\mathbf{G}_{l,i}$, where $G_{l,i,x} = 1$ if pixel x of image i is of the label l in the ground truth annotation and 0 otherwise. Then the fitting score of image i's polygons to $\mathbf{P}_{l,k}$ is defined as:

$$Score(\mathbf{G_{l,i}}, \mathbf{P_{l,k}}) = \frac{\sum_x min(G_{l,i,x}, P_{l,k,x})}{\sum_x max(G_{l,i,x}, P_{l,k,x})}$$

This can be interpreted as a weighted version of region overlap score for two polygons. For each class l, we select the top image i from the training images set whose matching score is highest for query image k and is bigger than a threshold $t = 0.3$.

Similarly, we define global matching score as:

$$Score(\mathbf{G_i}, \mathbf{P_k}) = \frac{\sum_l \sum_x min(G_{l,i,x}, P_{l,k,x})}{\sum_l \sum_x max(G_{l,i,x}, P_{l,k,x})}$$

To preserve global layout similarity, we only consider matching polygons whose image global matching scores $Score(\mathbf{G}_i, \mathbf{P}_k)$ are among the highest $R = 200$ images.

Our initial set of polygons might overlap or leave some pixels unlabeled. However, they preserve the simplicity of the real world regions and can be helpful for further inference. To resolve the overlapping issue, we assign the overlapping region exclusively to the polygon which has the highest confidence in the overlapping part.

We then create a polygonal shape prior by putting down the polygons and assign the unlabeled pixel to its nearest polygon region. This gives a clean, polygonal layout of the complete scene and is fed back into our context classifier for final training. Since visible part of the sky and trees cannot be occluded by other background regions and usually has complicated boundaries, we do not put any shape prior on those regions.

2.4 Baselines

Other general-purpose methods to infer labels of occluded regions do not exist in the literature, so we provide several baselines. Each method attempts to predict the labels of the underlying surfaces, given the label confidences for the visible surfaces.

Most confident background assigns each foreground pixel to the most confident background label.

Nearest method assigns occluded background pixels to the nearest (in image location) visible background pixel.

We also tried using **graph cut segmentation** with alpha-beta swaps [22]. Each pixel is represented a node in the graph. Our *unary term* contains the log probability that a pixel x of a query image k has background class label l: $\psi_{unary}(x, l) = \log P(l; x) = \log P_{l,k,x}$ directly from the output of our visible parser. Our *pairwise term* enforces contrast-sensitive boundaries in visible regions and uniform smoothing in occluded regions for adjacent pixels x_1 and x_2:

$$\psi_{pairwise}(x_1, x_2, l_1, l_2) = \mathbf{1}(l_1 \neq l_2)[\lambda_1 P(fg|x_1) + \lambda_2(1 - P(fg|x_1))e^{\frac{(I(x_1)-I(x_2))^2)}{\sigma^2}}]$$

where $P(fg|x_1)$ is probability that x_1 is in a foreground region. σ is a parameter that controls the amount of smoothing. λ_1 and λ_2 modulate how much label smoothing we want for visible and occluded portions of the image.

3 Experiments

In this section we show both quantitative and qualitative results for predicting scene layout on three different datasets, StreetScenes, IndoorScenes and SUN09.

We evaluate the pixel labeling accuracy of the background categories on all three datasets. For each experiment, we use 3 iterations of feed-forward training. The ground truth pixel map is created with only complete background classes, as if the foreground objects are not there. Unlabeled regions in the ground truth are ignored for evaluation. Our pixel accuracy is reported as the average pixel accuracy over images.

3.1 StreetScenes

The StreetScenes dataset consists of 3547 high quality images of urban environments, in which 710 images are for testing. The dataset is hand-annotated with polygonal, complete region labels. In this dataset, background classes as "road" and "sidewalk" is often heavily occluded by foreground objects like "car" and "pedestrian".

All three baselines described in Section 2.4 are tested on this dataset and provide very similar performance (Fig. 5(a)). Most Confident and Nearest baselines methods leave the pixels predicted as background intact; they cannot correct any mistakes on the visible part of the scene. The Nearest method mainly works if the visible prediction is smooth and correct, but the method fails when the visible prediction is cluttered. Notice that Graphcut also made almost no improvement compared to Most Confident, because it does not go beyond smoothing the label map. As shown on Fig. 4(e), Graphcut smooths out the sidewalk region just like the other two baselines. However, using the feed forward discriminative learning approach, the sidewalk is correctly recovered. We also tested SuperParsing [18], which performs similarly with the baseline methods on the visible part, but poorly on occluded regions, because it depends heavily on local visible features.

Our full system outperforms all baselines, eliminating 18% of the error from the Most Confident method. We evaluated the effectiveness of different components of our system in Fig. 5(b). The polygonal shape prior has better accuracy on occluded portion but does not improve the labeling of visible portions.

3.2 IndoorScene dataset

The IndoorScene dataset has 308 indoor images with 5 ground truth layout surfaces, annotated also in polygons: floor, left wall, middle wall, right wall, and ceiling. The challenge in the IndoorScene dataset is that the foreground "clutter" such as furniture occludes background regions.

One adaption we made for this dataset is that we transfer polygons all from one image to the query image instead of transferring from different images so that the box surfaces agree on geometry with each other in the final prediction. For evaluation, we compare to the original baseline, the confidence map trained using Geometric Context classifier. Our method increases the labeling overall by 1.6% (Fig. 6(a)). For the occluded region, the pixel accuracy was improved from 0.658 to 0.729. Hedau et al.'s method [2], which incorporates vanishing point estimates, outperforms us with 0.801 on overall accuracy, though the method applies only to indoor scenes.

Method	Complete scene	Occluded	Visible
Most Confident	0.795	0.601	0.810
Nearest	0.798	0.619	0.812
Graphcut	0.803	0.615	0.818
Ours	**0.833**	**0.715**	**0.843**
Superparsing	0.775	0.453	0.800

(a) Pixel accuracy on StreetScene dataset

Method	Complete scene	Occluded	Visible
Most Confident	0.795	0.601	0.810
Shape Prior Only	0.818	0.713	0.826
Ours, w/o Shape Prior	0.832	0.705	0.842
Ours, w/o Detection	0.831	0.705	0.841
Ours, Full	**0.833**	**0.71**5	**0.843**

(b) The effectiveness of different cues of our framework

Fig. 5. (a) shows our result compares favorably to the baselines. The “Most Confident”, “Nearest” and “Graphcut” are based on the confidence maps of training the classifier of Hoiem et al.. The SuperParsing results shown are produced in the Most Confident fashion. Using Nearest or GraphCut on SuperParsing results yields similar performance. (b) Explores the effectiveness components in our framework, also on StreetScene dataset. “Shape Prior Only” correspond to the pixel map of the best polygonal layout guess, as shown in Fig. 4 (h). “Ours, w/o Shape Prior” has everything except transferring shape prior. “Ours, w/o Detection” uses everything except object detection cues.

3.3 SUN09 dataset

We use the subset of 8684 images from SUN09 dataset, containing both indoor and outdoor images. We manually cleaned up the tags, among which “vehicle”, “chair”, “people” and “object” are foreground; “building”, “ceiling”, “floor”, “ground”, “field”, “road”, “sky”, “tree”, “wall”, “water” and “sidewalk” are background. The overall pixel accuracy are reported Fig. 6(b). Similarly to the previous two experiments, we see significant improvement on occluded regions (from 49.8% to 66.1%) and modest increase on the visible regions.

3.4 Qualitative results

We show qualitative results from both StreetScenes (Fig. 8) and IndoorScene dataset (Fig. 9). We first show our visible surfaces and detected objects. The gray area indicates background regions occluded by the foreground objects. Then using feed-forward inference, the missing background regions are completed, and then polygons are fit to those regions creating complete polygonal layout proposals. Finally those polygons are used as shape prior to refine the pixel labels. Qualitative results on SUN09 will be included the supplemental material due to space constraints.

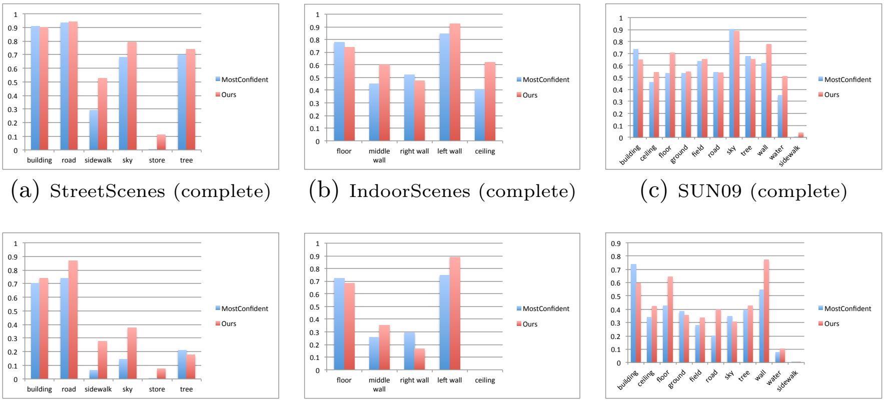
Method	Complete Scene	Occluded	Visible
Most Confident	0.700	0.658	0.710
Ours	0.739	0.729	0.742

(a) Pixel accuracy on IndoorScenes

Method	Complete scene	Occluded	Visible
Most Confident	0.639	0.498	0.665
Ours	0.691	0.661	0.695

(b) Pixel accuracy on SUN09

Fig. 6. The overall accuracy on testing set of IndoorScenes and SUN09 dataset, our method greatly helped occluded part of the scene, and made slight improvement on the visible portion.



(a) StreetScenes (complete) (b) IndoorScenes (complete) (c) SUN09 (complete)

(d) StreetScenes (occluded) (e) IndoorScenes (occluded) (f) SUN09 (occluded)

Fig. 7. Per-class pixel accuracy on all 3 datasets we experimented with. Note that there is no number on "ceiling" for occluded IndoorScene dataset because they are never occluded. For SUN09, our biggest improvement comes from background classes that are often occluded, such as wall, floor and road.

4 Conclusion

We have described a simple and general approach to label both visible and occluded portions of background. Our approach does not require hand-designed priors, but instead applies non-parametric scene priors learned from the training set. Our method works surprisingly well, especially for the StreetScene dataset where many training images are available. For the indoor dataset, our method outperforms our baselines but does not perform as well as Hedau et al.'s method [2] that directly incorporates geometric priors and structured learning. Our generic approach for inferring occluded background regions would serve as a good starting point that could be extended with domain-specific priors and constraints.

Acknowledgements: This research was supported in part by NSF Award IIS 0904209 and ONR MURI Award N000141010934.

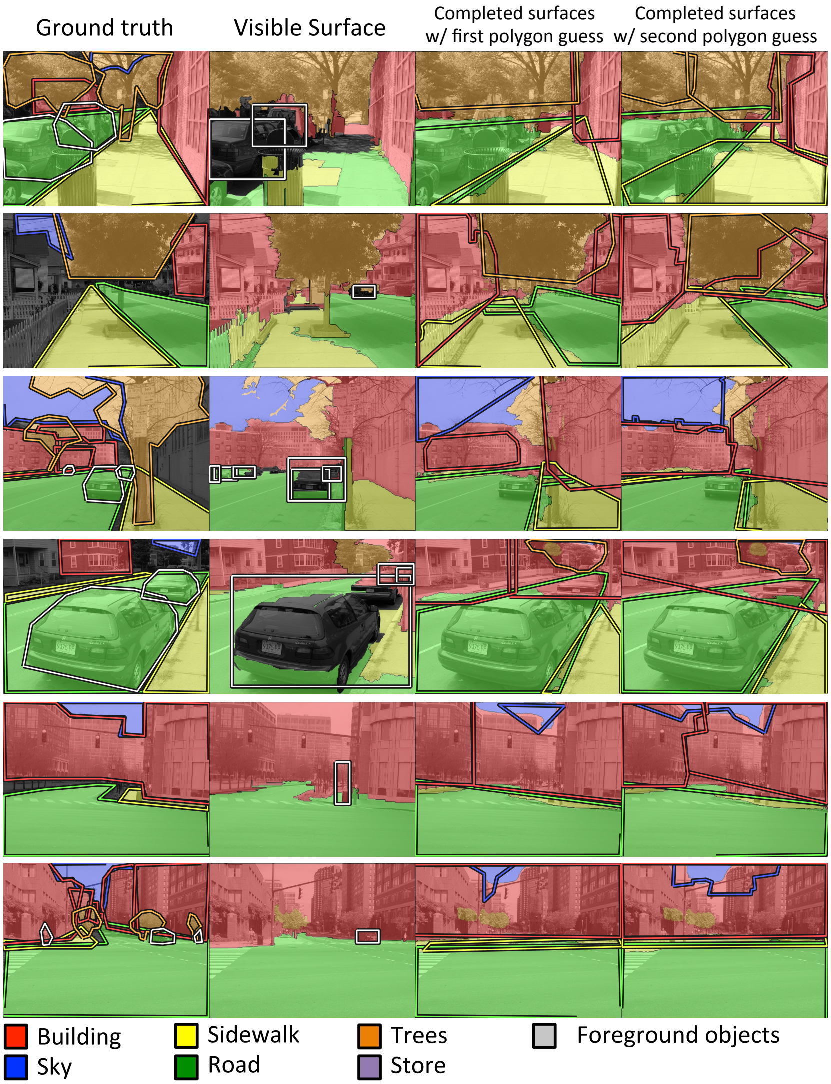

Ground truth
Visible Surface
Completed surfaces w/ first polygon guess
Completed surfaces w/ second polygon guess
Building
Sky
Sidewalk
Road
Trees
Store
Foreground objects

Fig. 8. Qualitative results on street scenes: Left to right: ground truth; labeling into visible surfaces and detected objects; labeling of completed surfaces with first polygon guess; same labeling with second polygon guess. In each image, the region colors indicate pixel labels. The polygons in the right two columns indicate the transferred regions, representing different hypotheses about individual structures. For example, top row: red polygons indicate the possibility that the building region is composed of one building or two. Bottom row: sidewalk is incorrectly hallucinated to cross the road. Note that our system is often able to infer sidewalk regions that are nearly fully occluded.

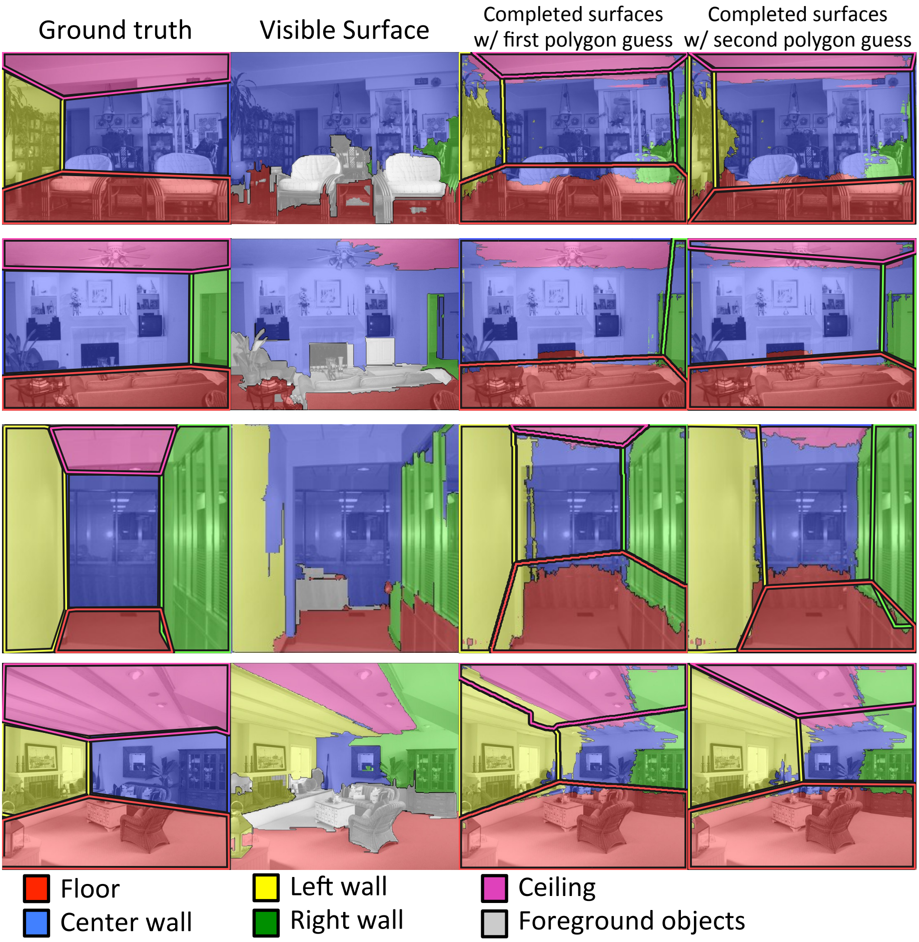


Fig. 9. Qualitative results on indoor dataset: Left to right: ground truth; labeling into visible surfaces; labeling of completed surfaces with first polygon guess; same labeling with second polygon guess. In each image, the region colors indicate pixel labels. We can infer the room structure using the same process as for outdoor scenes. Although our method does not outperform Hedau et al.'s domain-specific method [2] that incorporates strong geometric priors, our method does outperform the initial surface labeler used by them.

References

1. Geiger, A., Wojek, C., Urtasun, R.: Joint 3d estimation of objects and scene layout. In: NIPS. (2011)
2. Hedau, V., Hoiem, D., Forsyth, D.: Recovering the spatial layout of cluttered rooms. In: ICCV. (2009)
3. Tu, Z., Bai, X.: Auto-context and its application to high-level vision tasks and 3d brain image segmentation. PAMI **32**(10)
4. Bileschi, S.M.: Streetscenes: towards scene understanding in still images. PhD thesis, Cambridge, MA, USA (2006) AAI0810070.
5. Shotton, J., Winn, J., Rother, C., Criminisi, A.: Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In: ECCV. (2006)
6. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results. http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html
7. Choi, M.J., Lim, J.J., Torralba, A., Willsky, A.S.: Exploiting hierarchical context on a large database of object categories. In: CVPR. (2010)
8. Hoiem, D., Efros, A.A., Hebert, M.: Recovering surface layout from an image. IJCV **75**(1) (2007) 151–172
9. Brostow, G.J., Shotton, J., Fauqueur, J., Cipolla, R.: Segmentation and recognition using structure from motion point clouds. In: ECCV. (2008)
10. Gould, S., Gao, T., Koller, D.: Region-based segmentation and object detection. In: NIPS. (2009)
11. Lee, D.C., Hebert, M., Kanade, T.: Geometric reasoning for single image structure recovery. In: CVPR. (2009)
12. Gupta, A., Efros, A.A., Hebert, M.: Blocks world revisited: Image understanding using qualitative geometry and mechanics. In: ECCV. (2010)
13. Russell, B.C., Torralba, A., Murphy, K.P., Freeman, W.T.: LabelMe: a database and web-based tool for image annotation. Technical report, MIT (2005)
14. Li, C., Kowdle, A., Saxena, A., Chen, T.: Towards holistic scene understanding: Feedback enabled cascaded classification models. In: NIPS. (2010)
15. Hoiem, D., Efros, A.A., Hebert, M.: Closing the loop on scene interpretation. In: CVPR. (2008)
16. Gould, S., Rodgers, J., Cohen, D., Elidan, G., Koller, D.: Multi-class segmentation with relative location prior. IJCV **80**(3) (2008) 300–316
17. Malisiewicz, T., Efros, A.A.: Beyond categories: The visual memex model for reasoning about object relationships. In: NIPS. (2009)
18. Tighe, J., Lazebnik, S.: Superparsing: Scalable nonparametric image parsing with superpixels. In: ECCV. (2010)
19. Liu, C., Yuen, J., Torralba, A.: Nonparametric scene parsing via label transfer. PAMI **33**(12) (2011)
20. Zhang, H., Xiao, J., , Quan, L.: Supervised label transfer for semantic segmentation of street scenes. In: ECCV. (2010)
21. Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. PAMI (2009)
22. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? PAMI. **26**(2) (2004) 147–159