Int J Comput Vis (2015) 112:172–187
DOI 10.1007/s11263-014-0776-7

Labeling Complete Surfaces in Scene Understanding

Ruiqi Guo · Derek Hoiem

Received: 29 January 2014 / Accepted: 14 October 2014 / Published online: 12 November 2014

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 three datasets: the outdoor StreetScenes dataset, IndoorScene dataset and SUN09 dataset, all using the same approach. Furthermore, the proposed approach is extended to 3D space to find layered support surfaces in RGB-Depth scenes. Our experiments and analysis show that our method outperforms competent baselines.

Keywords Scene understanding · Image parsing · Geometric layout · RGB-depth

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. (2011) infer the full road layout from video, and Hedau et al. (2009) 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 more likely to be 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, 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 (Tu and Bai 2010). 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;

Communicated by Derek Hoiem, James Hays, Jianxiong Xiao, and Aditya Khosla.

R. Guo (✉) · D. Hoiem
Department of Computer Science, University of Illinois at Urbana-Champaign, Champaign, USA
e-mail: guo29@illinois.edu

D. Hoiem
e-mail: dhoiem@uiuc.edu

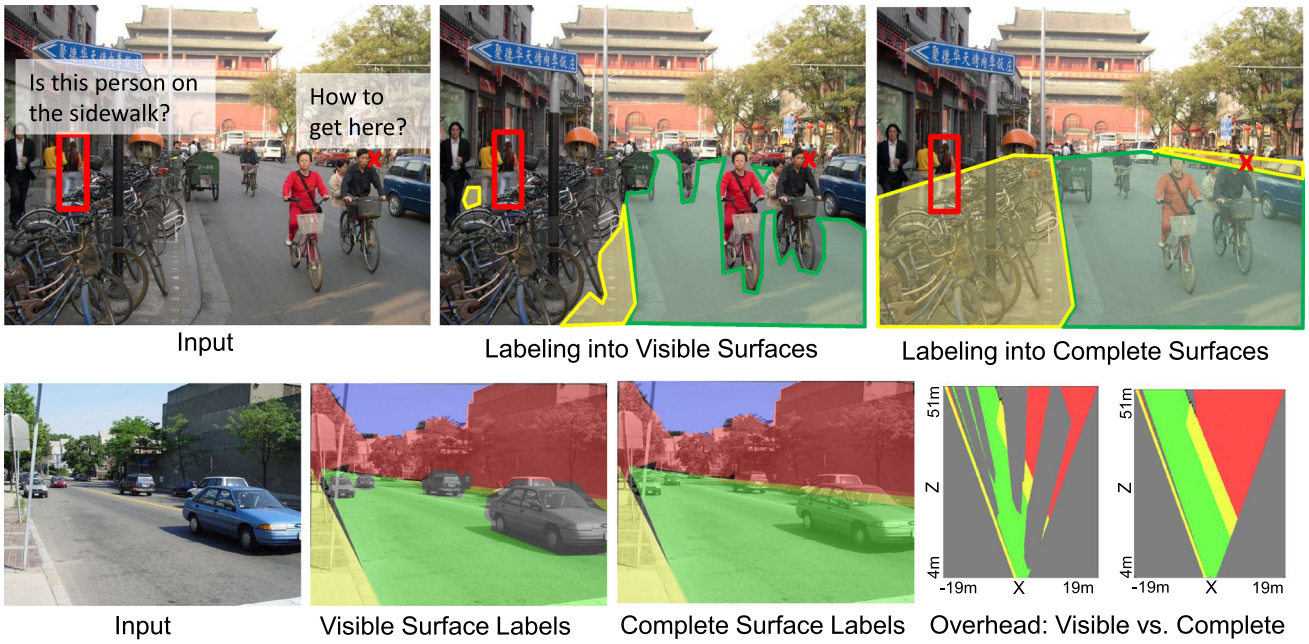


Fig. 1 Motivation Scene parsing is often viewed as a problem of labeling pixels into visible categories. But these representations leave much of the underlying scene unknowable. For example, because the woman (*top row*) is occluded, we cannot determine what she is standing on without inferring that 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 (Color figure online)

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. The 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.

To demonstrate the generality of our approach, we perform experiments on the CBCL StreetScenes dataset (Bileschi 2006), Hedau et al. (2009)'s indoor scene dataset as well as the more diverse SUN09 dataset (Choi et al. 2010). 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 pixel-wise MRF solved with graph cuts. To analyze different components of the algorithm, we studied the performance with respect to different parameters of the feed-forward procedure. The transferred polygons can be seen as a guess of the configuration of scene, and thus provide more structured understanding than pure pixel label predictions. To analyze the performance of such polygon prediction, we treat the problem as a polygon detection problem, evaluating how much of the polygons have been correctly predicted based on intersection over union criteria. So far, our representation encodes only a single background layer, projected onto the image plane. In Sect. 4, we describe an extension to multiple layers accounting for 3D geometry, predicting layered support surfaces in an overhead view in RGB-D images. Earlier versions of this work were described in Guo and Hoiem (2012, 2013).

1.1 Related Work

1.1.1 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 (Shotton et al. 2006), Pascal VOC Segmentation (Everingham et al. 2008), SUN'09 (Choi et al. 2010), Geometric Context (Hoiem et al. 2007), CamVid (Brostow et al. 2008), and the Stanford Background Dataset (Gould et al. 2009). Even methods that are focused on scene geometry, such as Hoiem et al.'s surface layout (Hoiem et al. 2007) are restricted to labeling visible surfaces. Notable exceptions include recently proposed indoor scene datasets (Lee et al. 2009; Hedau et al. 2009), Geiger et al.'s work (Geiger et al. 2011) to infer the upcoming road plan from a vehicle-mounted video camera, and Gupta et al.'s blocks world revisited (Gupta et al. 2010). 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. (2007). 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 (Russell et al. 2005) and CBCL StreetScenes (Bileschi 2006) instruct labelers to draw polygons around the entire background region, ignoring occluders. In fact, Bileschi's thesis (Bileschi 2006) 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.

1.1.2 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 (Tu and Bai 2010), 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 the original auto-context framework, 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., Li et al. 2010; Hoiem et al. 2008; Gould et al. 2008). We favor auto-context approach for its simplicity, efficiency, flexibility, and intuitive behavior as a form of belief propagation in an MRF (Tu and Bai 2010; Ross et al. 2011).

1.1.3 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 (Malisiewicz and Efros 2009), Tighe and Lazebnik's superparsing (Tighe and Lazebnik 2010), and other label transfer methods by Liu et al. (2011) and Zhang et al. (2010). 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. (2011) and Zhang et al. (2010) 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.

1.1.4 Applications

Many applications can be derived from understanding the portion of the scene that is not directly visible. Isola and Liu (2013) are interested in composing a scene with the right depth ordering using the a collection of un-occluded polygons, which can be used in scene synthesis and image editing. Silberman et al. (2014) propose a method for finding the complete extent of 3D surfaces, and created an augmented reality application with the completed surfaces. Recently, Khosla et al. (2014) proposed to find places when they are not visible in the scene. For example, they can predict if there is a Starbucks nearby using visual features even if there is not a Starbucks in sight.

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. Our approach incorporates existing techniques, such as region classification (Hoiem et al. 2007), feed-forward contextual prediction (Tu and Bai 2010), and non-parametric label transfer (Liu et al. 2011), 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

We summarize the process to estimate complete scene layout in Fig. 2. We first label the 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

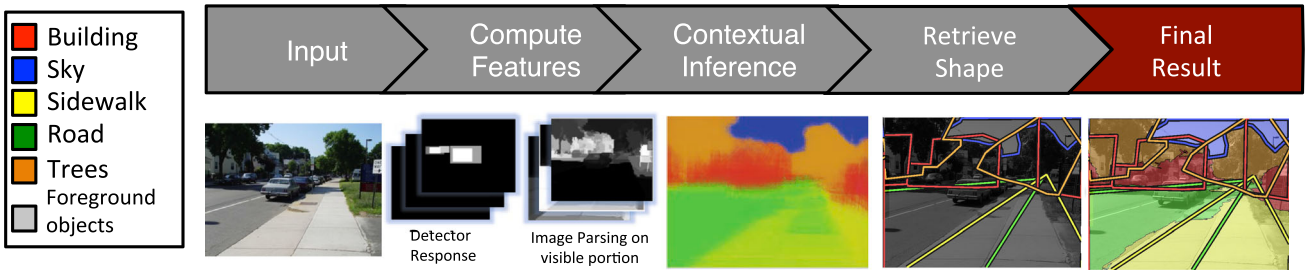


Fig. 2 Given an input image, we predict the labels of visible foreground and background surfaces and estimate object locations using detectors. We then apply a feed-forward contextual method to infer the labels of occluded background regions. We transfer training regions that match current estimates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

surface predictions and the transferred polygons to provide the final complete background labeling.

It is an interesting problem whether we can automatically determine which categories are foreground and which categories are background. However, we feel that the definition of background and foreground categories can sometimes be ambiguous without the ground truth depth ordering. For simplicity, 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. In Sect. 4, we present the RGB-D extension of our algorithm, in which the extent of support surfaces are multi-layered, and their ordering is determined bytheir heights.

2.1 Labeling Visible Surfaces and Objects

We apply the region classifier from (Hoiem et al. 2007) 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, 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 (Tighe and Lazebnik 2010). We use Felzenswalb et al.'s pre-trained object detector (Felzenszwalb et al. 2009) 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. We use both the background parsing results and detection confidence map as input to our contextual algorithm. The features maps are visualized in the second graph from the left in Fig. 1.

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 (Tu and Bai 2010), 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, which is illustrated in Algorithm 1. 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 (2010), 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 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 s per image (400×300 pixels), whereas the original algorithm runs at 30–70 s on a 300×200 image.

Due to its discriminative training, auto-context can go beyond simple smoothing. For example, in Fig. 3, auto-context 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

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 $\mathbf{P}^{(\mathbf{t-1})}$.

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 $\mathbf{P}^*$

Algorithm 1 Outline of our full algorithm. We use two types of features: (1) visible parsing results, (2) object detection. Then, we apply our major intuitions: spatial contexts and retrieved shape prior are combined to infer the complete label map and the polygonal layout

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 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-matching polygons and directly lay them down 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.

The most confident set of retrieved polygons provides our best guess of the configuration of the scene. Similarly, the set of second most confident polygons provides an alternative "guess". We show those guesses in the third and fourth column of our qualitative results (Sect. 3.5).

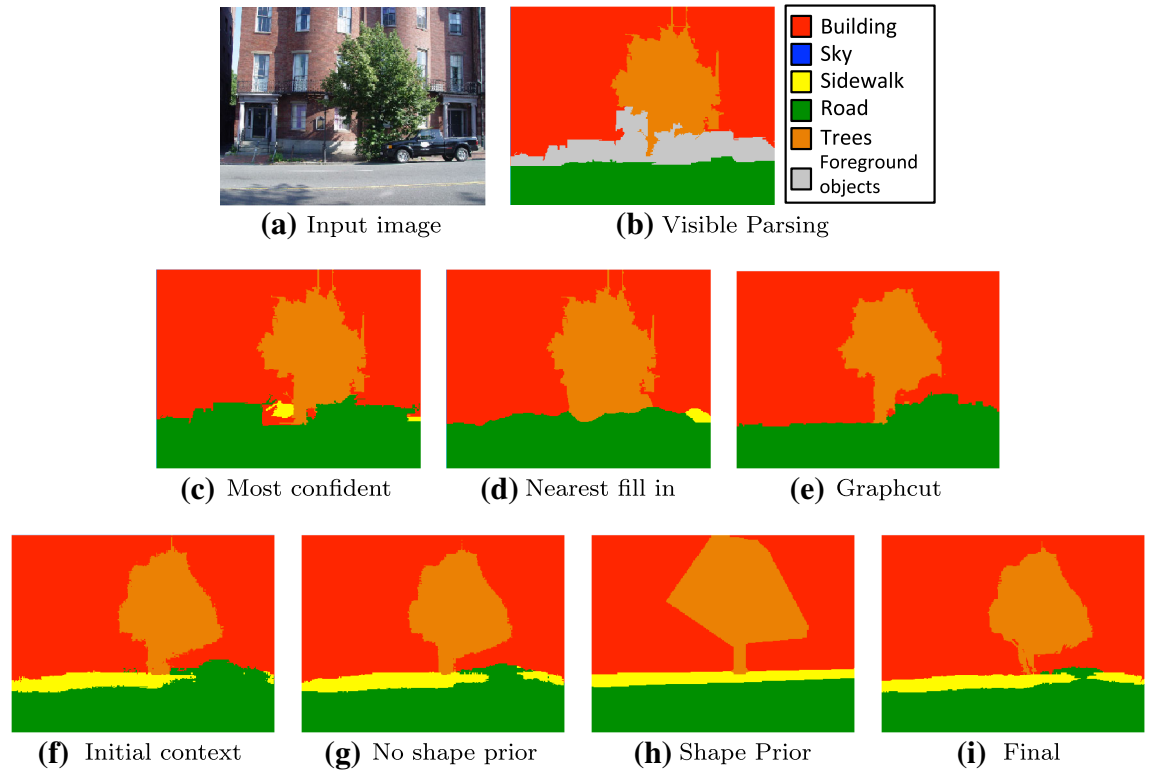


Fig. 3 Illustration of baseline methods. **a** shows the original image and **b** gives the parsing map of the visible part; **c**–**e** Baseline methods do not go beyond smoothing. **f** and **g** shows that incorporating context using classifier does more than a smoothing term, recovering the some sidewalk area. After polygons are retrieved, the flattened shape prior (**h**) helps to regularize the layout and gives final output (**i**) (Color figure online)

The polygon prediction provides a shape prior for background surfaces. Also, the polygons provide a more structured representation than pixels, enabling a connected set of building pixels to be represented as two separate buildings or a set of road pixels as a single road. In this way, the predicted polygons better reflect the worlds structure and may be useful for further inference. Multiple guesses for polygonal layout enable reasoning about uncertainty while representing whole surfaces.

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.

Graph-Cut implements a pixel-wise MRF, which is often used in post-processing stage of semantic segmentation. The setup of the MRF is similar to that of Shotton et al. (2006), where each pixel is represented a node in the graph. The *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. The *pairwise term* enforces contrast-sensitive boundaries in the visible regions and uniform smoothing in occluded regions for adjacent pixels x_1 and x_2:

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

where $P(occ|x_1)$ is probability that x_1 is occluded by the 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. We use alpha-beta swaps (Kolmogorov and Zabih 2004) to solve the MRF.

3 Experiments

We show both quantitative and qualitative results for predicting scene layout on three different datasets, StreetScenes,

Table 1 This table 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

Method	Complete	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

Pixel accuracy on StreetScene dataset

The best result is in bold

Table 2 Explores the effectiveness of individual component 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. 3h. "Ours, w/o Shape Prior" has everything except transferring shape prior. "Ours, w/o Detection" uses everything except object detection cues

Method	Complete	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**	**0.715**	**0.843**

The effectiveness of different cues of our framework

The best result is in bold

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" are often heavily occluded by foreground objects like "car" and "pedestrian".

All three baselines described in Sect. 2.4 are tested on this dataset and provide very similar performance (Table 1). 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. 3e, 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 (Tighe and Lazebnik 2010), 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 Table 2.

Table 3 Overall accuracy on testing set of IndoorScenes dataset: our method greatly helped occluded part of the scene, and made slight improvement on the visible portion

Method	Complete Scene	Occluded	Visible
Most Confident	0.700	0.658	0.710
Ours	0.739	0.729	0.742
Hedau et al. (2009)	0.796	0.698	0.821

Pixel accuracy on IndoorScenes

Table 4 Overall accuracy on testing set of SUN09 dataset: our method greatly helped occluded part of the scene, and made slight improvement on the visible portion

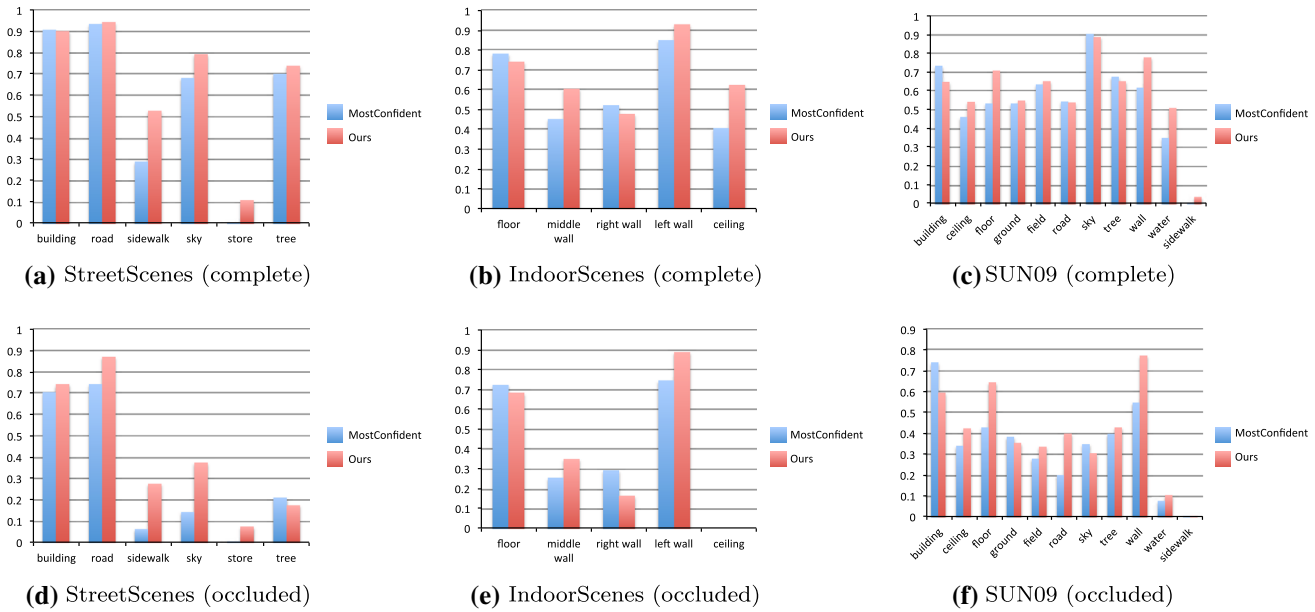
Method	Complete Scene	Occluded	Visible
Most Confident	0.639	0.498	0.665
Ours	0.691	0.661	0.695

Pixel accuracy on SUN09

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 3.9 % (Table 3). For the occluded region, the pixel accuracy was improved from 65.8 to 72.9 %. Hedau et al.'s method (Hedau et al. 2009) was specifically designed for



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

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

Fig. 4 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. **a** StreetScenes (complete), **b** IndoorScenes (complete), **c** SUN09 (complete), **d** StreetScenes (occluded), **e** IndoorScenes (occluded), **f** SUN09 (occluded) (Color figure online)

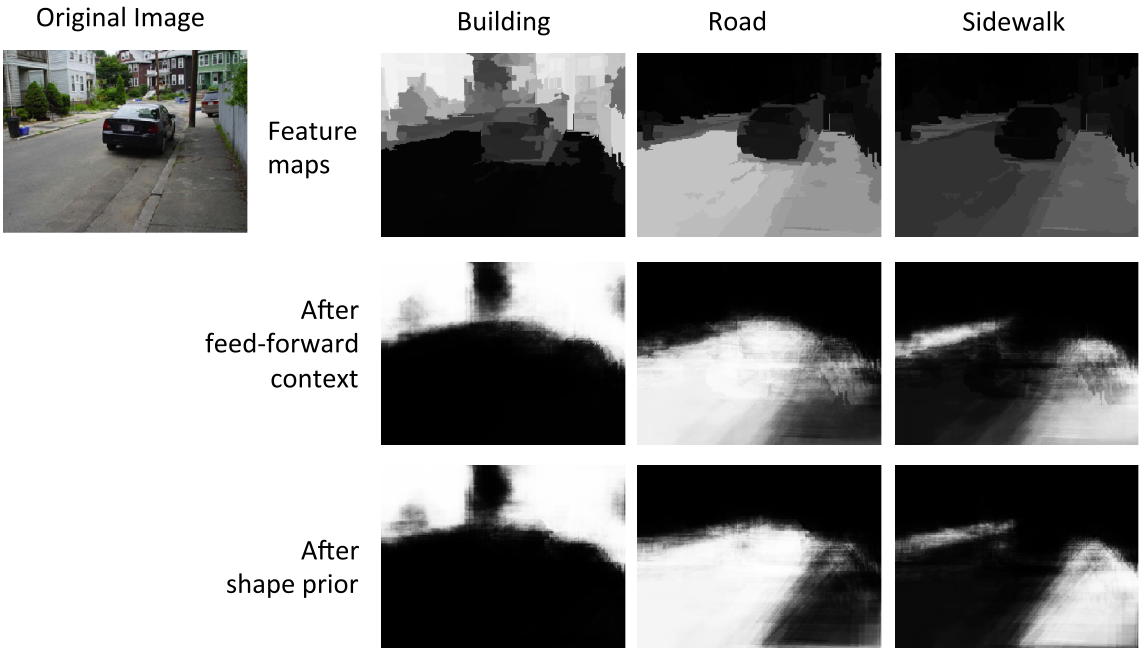


Fig. 5 The evolution of the probability map of semantic labels in StreetScene dataset. The original feature map has missing portion because of foreground occlusions. After three iterations of feed-forward contextual inference, the missing part is filled. Finally, adding the shape prior improves the coherence of the estimates

this task and incorporates vanishing point estimates, and obtained 79.6 % on overall accuracy. However, we outperform them on the occluded portion of the scene by 3.1 %.

3.3 SUN09 Dataset

We use the subset of 8684 images from the SUN09 dataset, containing both indoor and outdoor images. We manu-

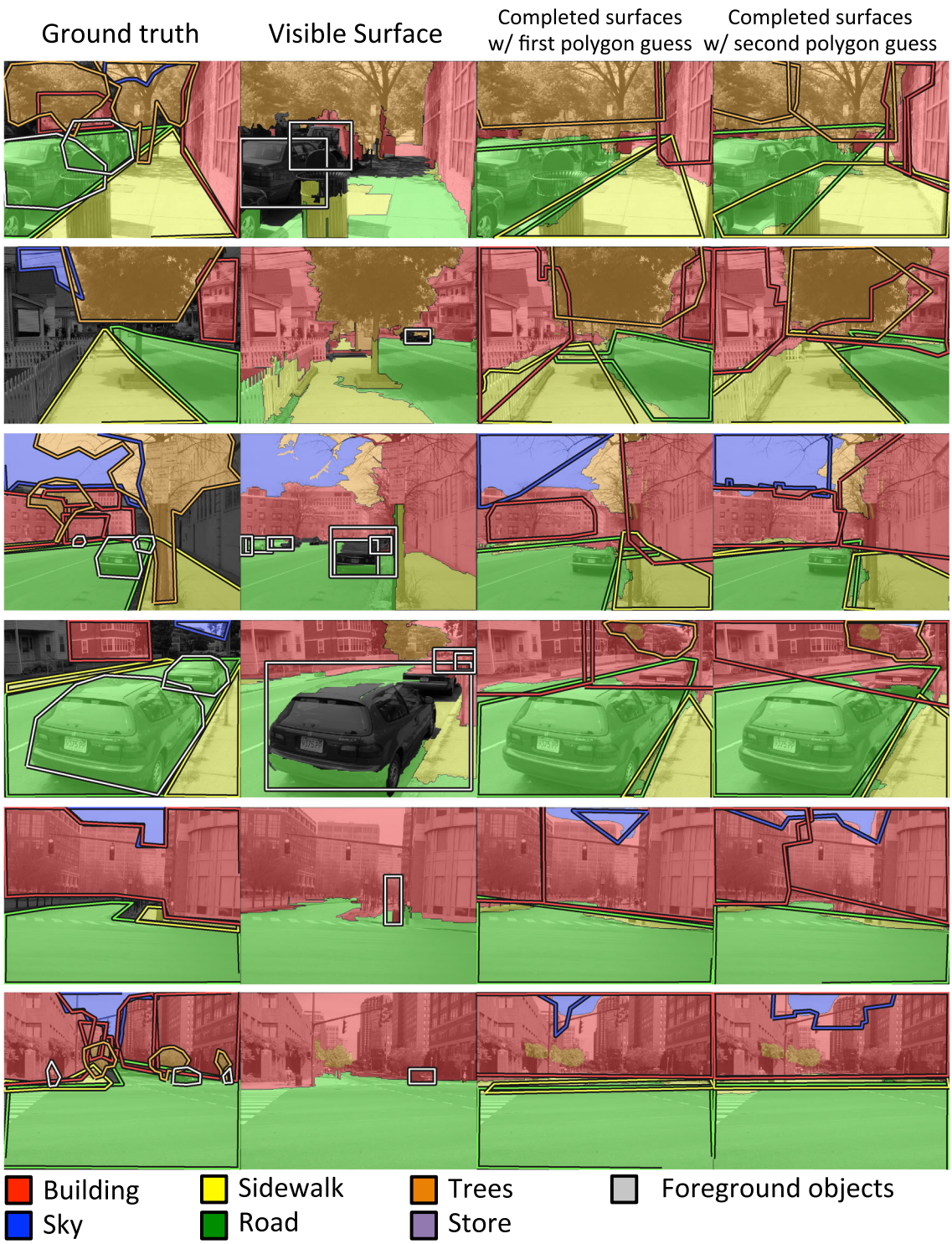


Fig. 6 Qualitative results on street scenes *Left* to *right* ground truth; labeling into visible surfaces and detected objects; labeling of completed surfaces with *first polygon* guess; same labeling with *second polygon* guess. In each image, the region *colors* indicate pixel labels. The *polygons* 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 (Color figure online)

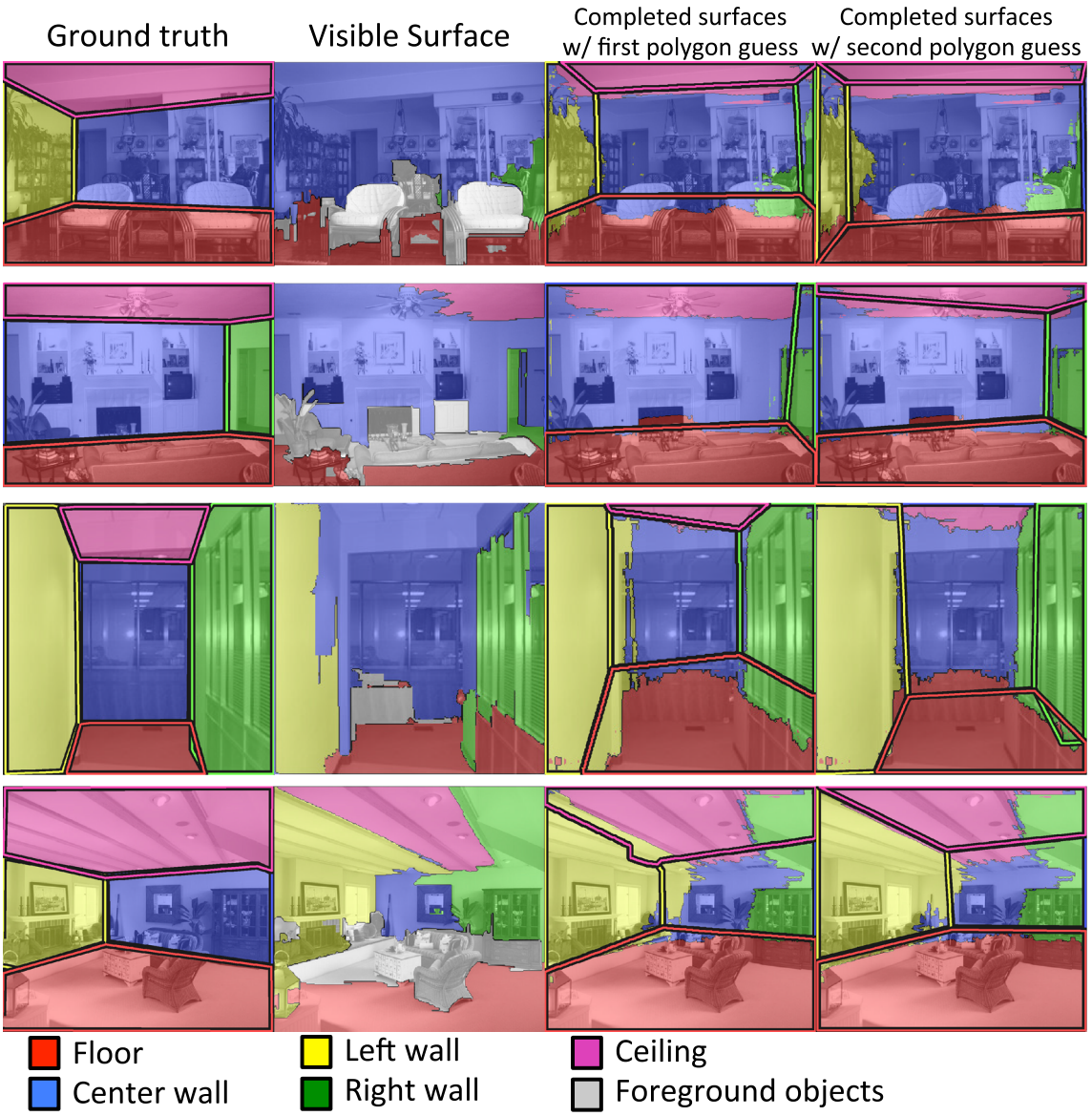


Fig. 7 Qualitative results on Indoor dataset: *Left* to *right*: ground truth; labeling into visible surfaces; labeling of completed surfaces with first polygon guess; same labeling with second polygon guess. In each image, the region *colors* indicate pixel labels. We can infer the room structure using the same process as for outdoor scenes. Although our method does not outperform Hedau et al.'s domain-specific method (Hedau et al. 2009) that incorporates strong geometric priors, our method does outperform the initial surface labeler used by them (Color figure online)

ally 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 Table 4. 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 Details

In the StreetScene dataset, 7.8 % of the pixels are occluded. IndoorScene and SUN09 has 23.1 and 15.5 %, respectively. In our experimental setting, we use a template with a radius of 49 and images are resized to 400 × 300. Each iteration of auto-contexts takes 0.1 s on a single CPU core while the shape retrieval takes 2 s with 17022 exemplar polygons. We

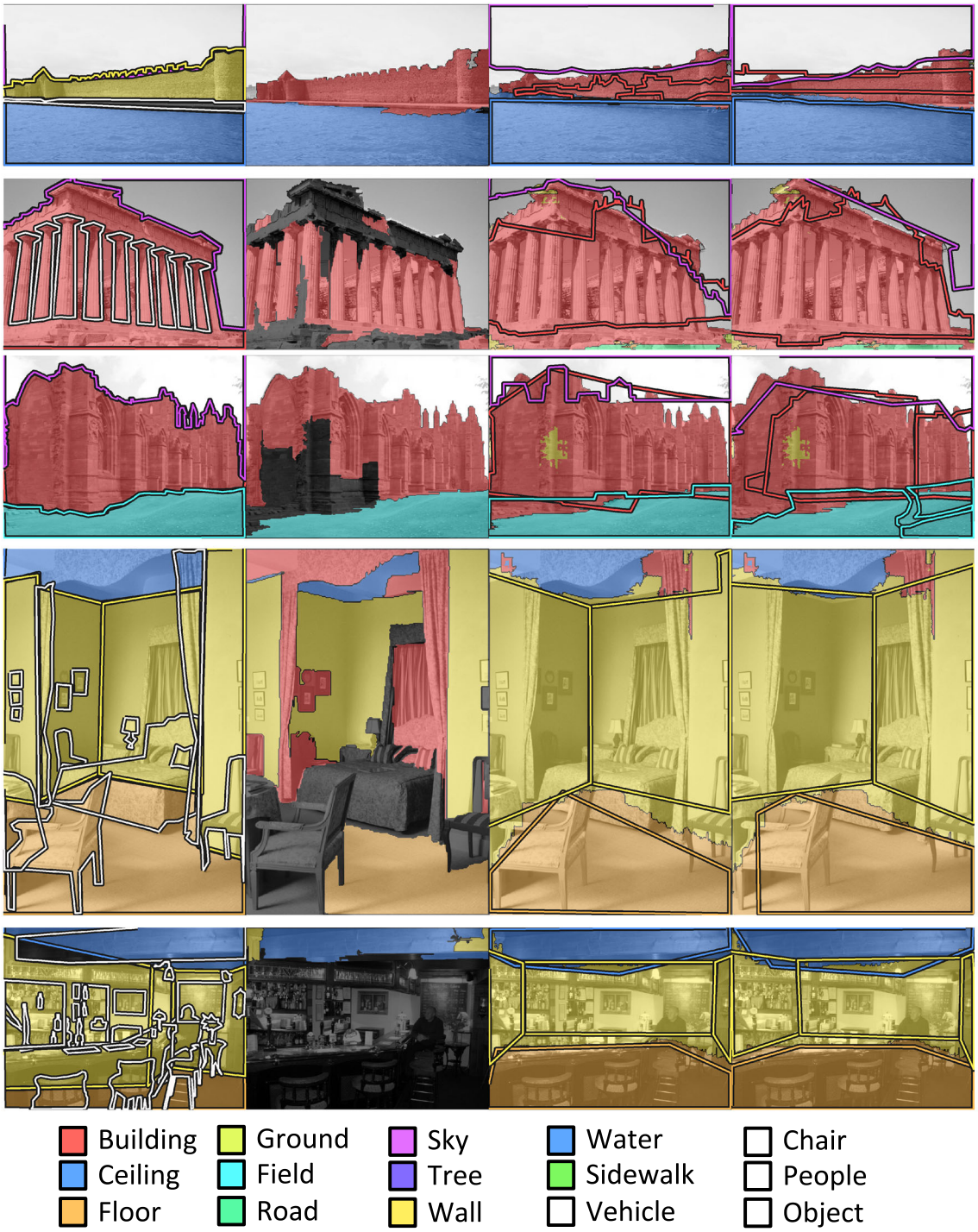


Fig. 8 Qualitative results on SUN09 dataset *Left* to *right*: ground truth; labeling into visible surfaces; labeling of completed surfaces with first polygon guess; same labeling with second polygon guess. In each image, the region *colors* indicate pixel labels. SUN09 features a variety of both indoor and outdoor images, and a broader range of foreground and background labels (Color figure online)

also studied the performance of our system with respect to different background region categories, shown in Fig. 4. On StreetScene dataset, the performance boost comes from the categories of road and sidewalk, which are often occluded by pedestrians and cars. On SUN09, the gain comes from floor, walls and roads, which suffer the most from foreground occlusions. Most of the performance gain is due to the use of context. The polygonal shape prior has better accuracy on occluded portion but does not improve the labeling of visible portions, as shown Fig. 5.

3.5 Qualitative Results

We show qualitative results from the StreetScenes (Fig. 6), IndoorScene dataset (Fig. 7) and SUN09 dataset (Fig. 8). 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.

3.6 Analysis of Contextual Inference

In this section, we examine design decisions we make in the proposed algorithm. Specifically, we want to understand how the performance changes with respect to different parameters in our feed-forward inference. The main parameters of the inference procedure described in this paper are (1) the template it uses, and (2) the number of iterations. With larger templates, we consider a bigger neighborhood and therefore capture longer-range spatially varying interactions of semantic labels. More iterations help the label prediction to converge. However, they also mean a higher computation cost, which grows linearly with the template size and the number of iterations.

In Fig. 9, we show the performance of the feed-forward procedure with respect to different template size and iteration numbers. Overall, the procedure converges within less than 5 iterations. When the template is small (under the radius of 10 pixels), the performance does not change much after just two iterations. The impact of iterations increases when the template is bigger, but the most improvement comes from the first few iterations. The template size on the other hand makes a huge difference. The improvement on pixel accuracy is 2.1 % when the radius of the template is 1 (thus 3×3 template, blue line in Fig. 9). However it becomes 5.7 % as the template increase to the radius of 169 (gray line in Fig. 9), almost a three-fold improvement. The accuracy gain on the occluded part is 4.0 and 18.2 % on the occluded part for template of radius 1 and 169, respectively.

It is interesting that the performance gain keeps increasing with the template size, until the template increases to the size of whole image. This confirms our intuition that we need to consider long-range interactions: the bigger the size of the template, the better the results. And the big template gives the algorithm an advantage to the pixel-wise MRFs with non spatially varying weights, such as the GraphCut baseline described in Sect. 2.4.

Our feed-forward procedure is more efficient than the original auto-context in Tu and Bai (2010), because it does not do "auto" feature selection. We tried to use different template configurations, instead of a rigid, ray-shape

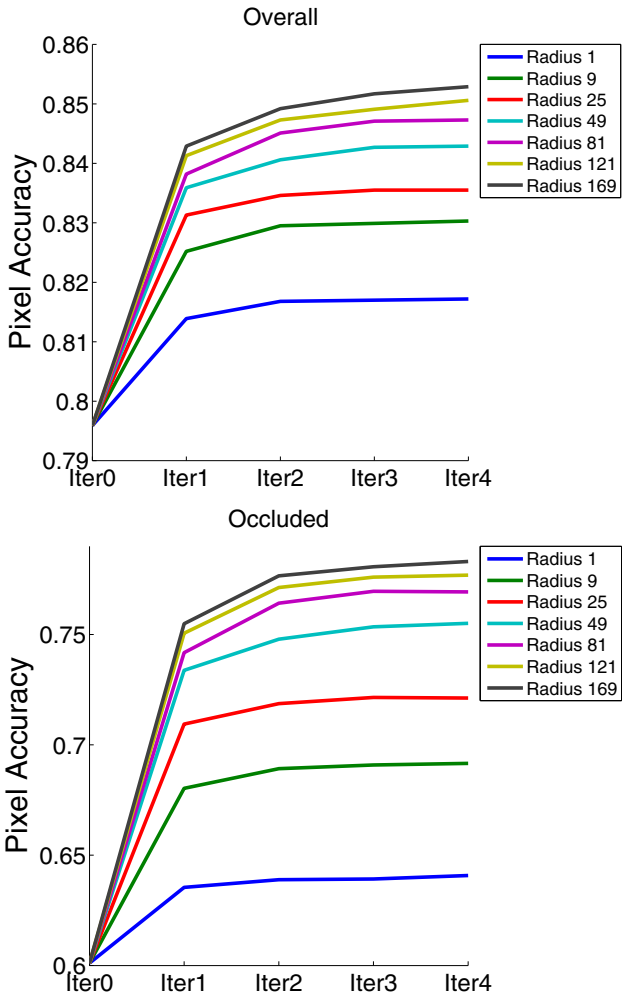
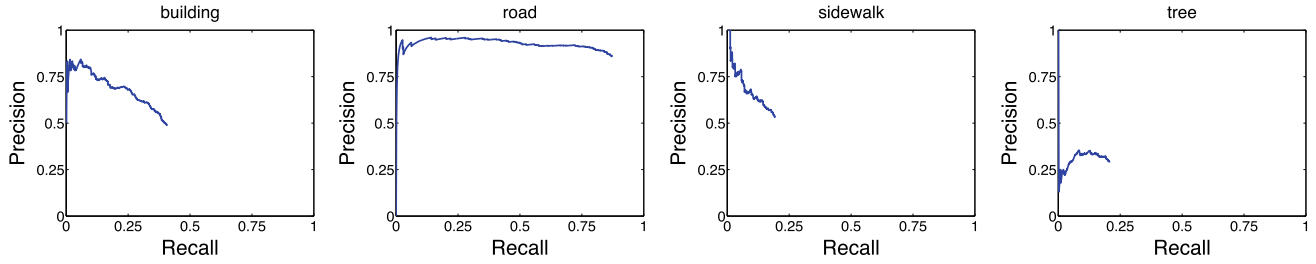


Fig. 9 The performance change on the StreetScene dataset with respect to the parameters. **1** template size and **2** number of iterations, on both the overall and the occluded portion of the scene. In both cases, the performance increases as the template becomes larger, until the radius become 169 pixels, for a total diameter of 339 pixels, nearly covering the 400×300 image (Color figure online)

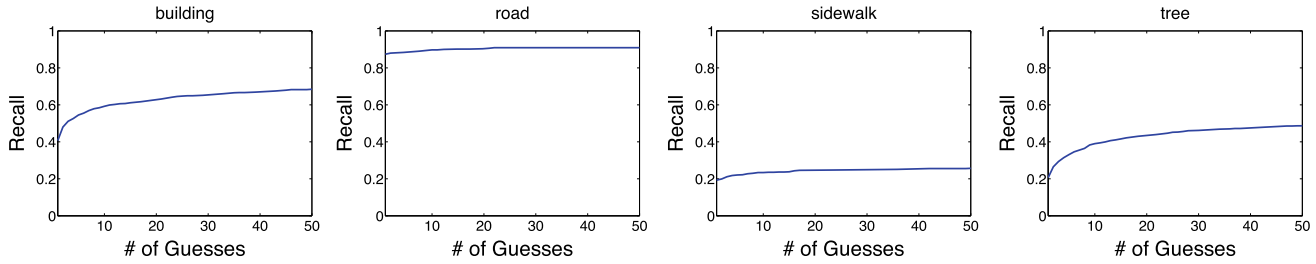
template. We sampled the templates from 2D Gaussian, Laplacian or triangle distributions to produce a randomized template and then cross-validate for the best-performing one. However, we found it does not yield noticeable improvement, and thus decided to keep things simple by using a ray-shape template. The other concern is the computation speed. For consistency and speed reasons, we use a template with the radius of 49 and 3 iterations in all our previous results.

3.7 Scene Understanding as Polygon Detections

The shape matching procedure can give us both the shape prior of the scene and the transferred polygons, which make up possible configuration of the scene. Predicting polygons is by itself an interesting way to understand scenes. Compared to bounding boxes, polygons are more expressive and can



(a) PASCAL style Precision Recall Curve of Polygons



(b) Highest possible recall at K layout proposals

Fig. 10 Evaluation of polygon detection. **a** Precision-Recall curve of polygon detections in the StreetScene dataset, similar to the way bounding box detections are evaluated. **b** Highest possible recall versus the number of polygon proposals. With more proposals, it is more likely that we detect the right object polygons

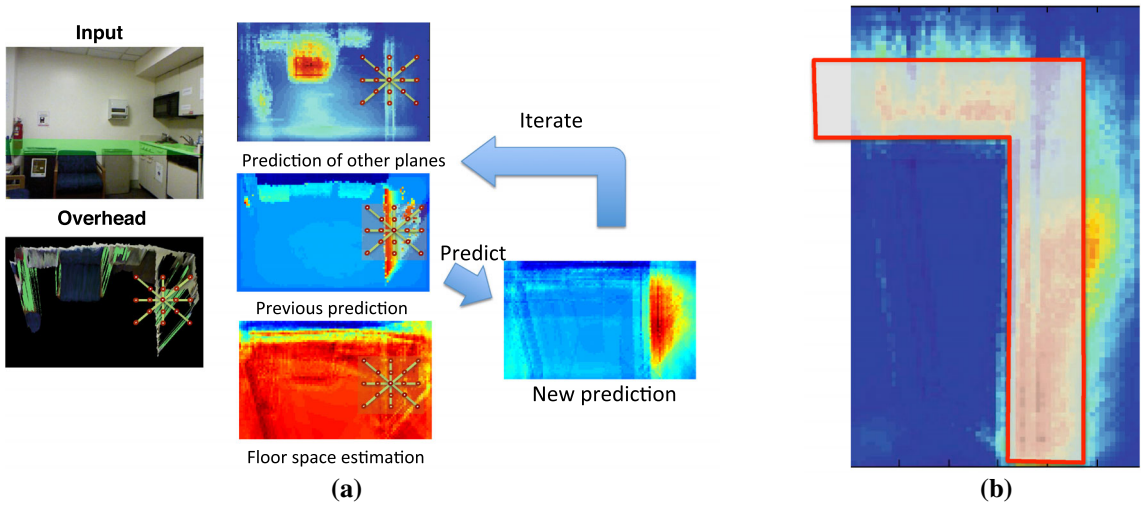


Fig. 11 Extension to RGB-D surfaces. **a** Adapting auto-context in 3D surface extent prediction: we consider the previous predictions of surfaces in the scenes all as feature maps **b** Shape matching in 3D: we compute the score of template matching based on the overlap, with positive score with the high probability area and negative scores on freespace (Color figure online)

represent a greater variety of shapes. Compare to pixel labeling, polygons are more structured because they are closer to human annotation/understanding and respect the shape regularity of individual entities. For example, a bunch of connected tree pixels can be one big tree or a row of trees. However, pixel labeling cannot tell the difference between them while polygonal representation can.

We follow the evaluation paradigm as in PASCAL object detection challenge (Everingham et al. 2008): a polygon detection is considered as correct if the intersection-over-union (IOU) score is over 0.5, and multiple detections are penalized. Note that this is a stricter criteria than IOU of bounding boxes, since regions can be of more flexible shapes.

On StreetScene dataset, the first guess of configuration has 3.41 polygons on average while the ground truth annotation has 3.33 polygons on average. We report the Precision-Recall curve in Fig. 10a. Note that the performance is very high for categories of "road" and "buildings", since they are often

large and continuous in outdoor scenes. "Trees" and "sidewalk", however, are usually separated and small, and thus 0.5 IOU becomes a harsh creteria for these categories. We also show the plot of recall versus the number of polygon layout we guess (Fig.10b). With increasing number of polygons proposals, the chance of detecting individual polygons become larger. We want to note that this is still a very simple baseline which does not consider diversity of proposals or the consistency between polygons. The proposed polygon may also have small overlap, but this does not usually happen because it is implicitly penalized through the matching procedure with probability map.

4 Extension to Modeling Overlapping 3D Surfaces

One limitation of our algorithm is that it works in 2D image plane and therefore it is possible that the polygons we produce do not agree in terms of viewpoint and geometry. Therefore we extend our inference into 3D space and work with RGB-D data. We propose a similar algorithm to find complete extent of surfaces in 3D scenes. We define support surfaces to be horizontal, planar surfaces that can physically support objects and humans. Given a RGB-D image, our goal is to localize the height and full extent of such surfaces in 3D space. First, we align the room with the dominate directions of the surfaces and then detect the the heights where support surfaces occur. Next, we formulate problem of finding the complete extent of support surfaces as parsing in the overhead view of the indoor scene. To this end, we adapt our current 2D approach to 3D space, by making two modifications: (1) use 3D features instead of 2D features (2) allow objects to translate and shift when doing the shape matching. Finally the predict support surface extent is evaluated against the manually annotated 3D models (Guo and Hoiem 2013) of the NYUv2 dataset (Silberman et al. 2012).

4.1 Approach

First, we label visible pixels into "floor", "wall", "ceiling" and "object" using the RGBD region classifier from Silberman et al. (2012) and then project these pixels into an overhead view using the depth signal, based on the same scene rotation matrix found in previous section. We then predict which heights are likely to contain a support surface based on a variety of 2D and 3D features. This includes (1) observed upward planar surfaces, (2) observed geometric labels, (3) edgemap, (4) voxel occupancy, (5) volumetric difference, (6) support height prior, (7) relative location prior. We refer the readers to the original paper (Guo and Hoiem 2013) for details on the 3D features.

When predicting support heights, the features are aggregated, followed by SVM classification and non maximum suppression. For extent prediction, the above feature maps are used and the iterative prediction procedure is applied. In addition to the previous predictions of the current plane, we also look at the prediction of support planes above and below it. The parameters of the feed-forward contexts used here is the same as the in the 2D cases: 3 iterations and 49 radius. The procedure of the inference is shown in Fig. 11. As in the case of 2D labeling, template matching can also serve to help. This time, we want to modify the matching procedure so as to allow translation and is done through FFT. Essentially, we want to encourage the template to overlap with high probability area in the probability map and penalizes the overlap with the free space, as illustrated in Fig. 11.

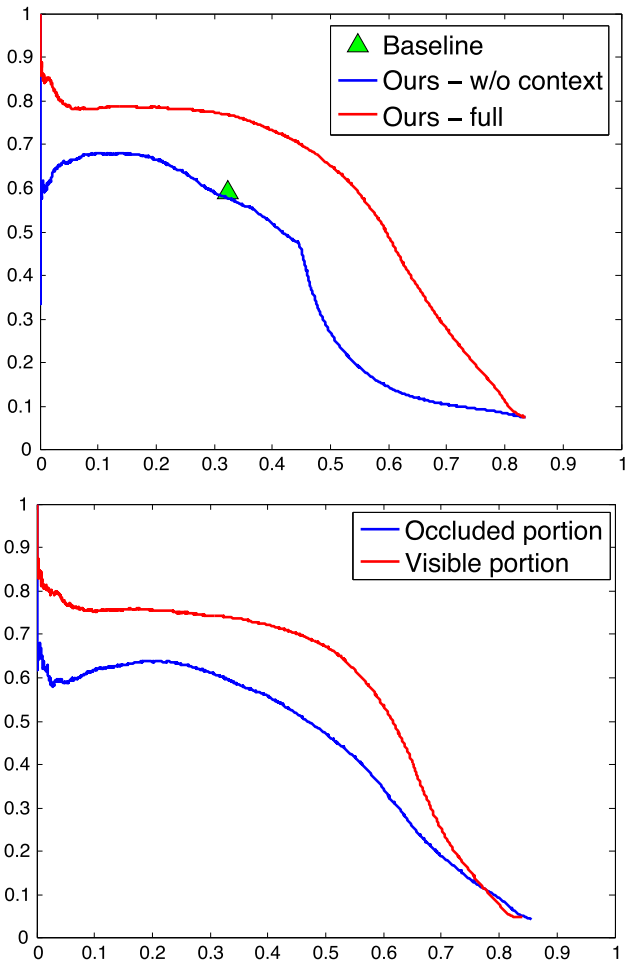


Fig. 12 Quantitative evaluation on support surface prediction. **a** PR curve of support extent prediction with and without auto-context, and the baseline accuracy. **b** Prediction on visible and occluded portion of the surface extent (Color figure online)

4.2 Evaluation

Support heights and extent can be naturally extracted from the 3D scene annotations in Guo and Hoiem (2013). To make evaluation less sensitive to noise in localization, we make the

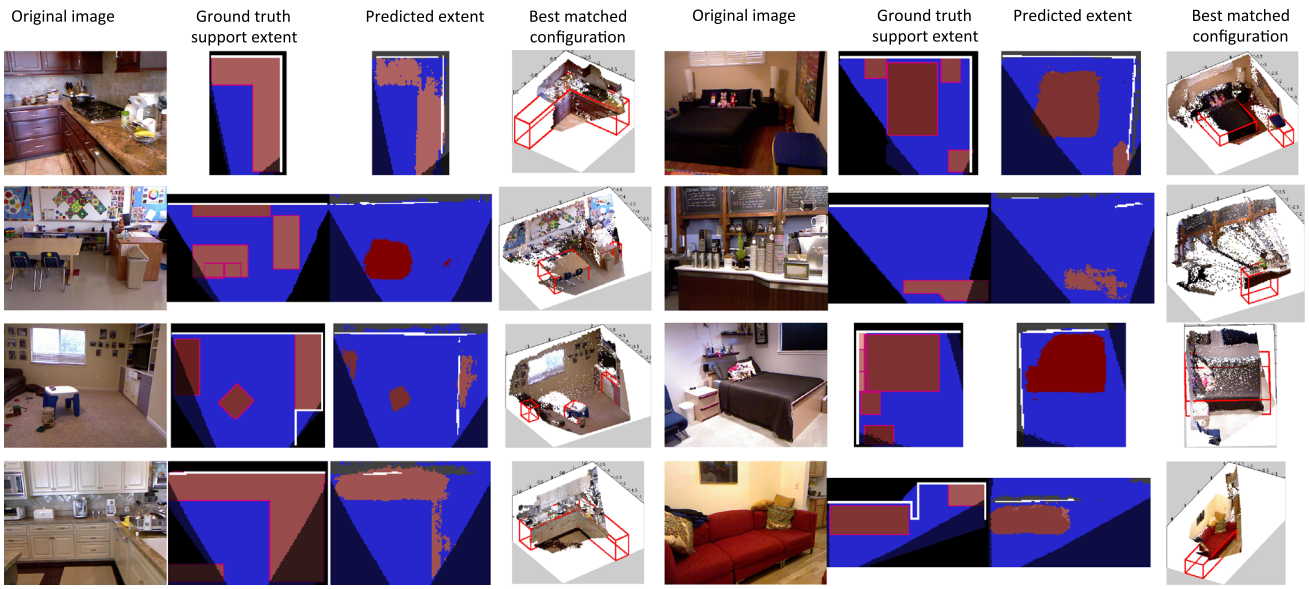


Fig. 13 Overhead visualization. *Green* and *blue* and *red* areas are estimated walls, floor and support surfaces respectively. The *brighter colors* of support surfaces indicate higher vertical heights relative to the floor. *Dark* areas are out of the field of view. The *first* and *second column* show in the original scene and its corresponding ground truth support surfaces. The *third column* shows our final prediction, by thresholding the probability map at the threshold which correspond to 0.5 recall. The *fourth column* shows the most confidently matched surface configuration (Color figure online)

area around boundary of support surface within a thickness of ϵ to be "don't care". ϵ is empirical error of Kinect over the dataset, which is set to 0.15 m. We also do not evaluate the area that is out of the field of view. In all, there are 5495 support surfaces in 1449 RGBD images, so on average 3.79 support surfaces per scene. In those support surfaces, 5095 are below the camera, while 400 are above it.

We evaluate accuracy of support extent prediction with precision-recall curves on support extent prediction. And all other pixels are labeled as negative so that duplicate detections are penalized. The results are shown in Fig. 12. We also compare performance for occluded support surfaces to un-occluded (visible) ones. In qualitative results, we show predictions that have confidence greater than the value corresponding to the 0.5 recall threshold. For support extent prediction we compare to a baseline of plane-fitting, based on the (Silberman et al. 2012) code for plane segmentation. We used their plane estimation which comes from a RANSAC and graph cut procedure, and post-process them using the appearance and surface normals. We see that our method outperforms the baseline by 17 % precision at the same recall level or 13 % recall at the same precision. In addition, we also see that the performance of the visible regions are much better than that of the occluded areas, as expected.

In the qualitative results of Fig. 13, we see that the transferred support planes can give us a rough estimation of individual support objects. Only the best configuration is displayed here, but our system can also generate the best K configurations. Because the configurations are generated in real 3D space, the support detected surface are guaranteed to be consistent in 3D geometry.

5 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 contributions can be summarized as follows: (1) We found the proposed method works well across a number of 2D datasets, including StreetScenes, IndoorScenes and SUN09, outperforming relevant baselines, especially on the occluded part of the surfaces. (2) Our further analysis of the method shows that it is vitally important to consider long range contextual information. (3) Our method proposes multiple polygonal hypotheses for surfaces, better modeling scene structure than the usual per-pixel labels. (4) Our method is extended to 3D space, and finds complete extent of support surfaces in RGB-D indoor scenes. We hope this generic approach for inferring occluded background regions would serve as a good starting point that could be extended with domain-specific priors and constraints.

References

Bileschi, S.M.: Streetscenes: Towards scene understanding in still images. Ph.D. thesis, Cambridge, MA (2006)

Brostow, G. J., Shotton, J., Fauqueur, J., & Cipolla, R.: Segmentation and recognition using structure from motion point clouds. In: *Proceedings of the 10th European Conference on Computer Vision ECCV* (2008).

Choi, M. J., Lim, J. J., Torralba, A., & Willsky, A. S.: Exploiting hierarchical context on a large database of object categories. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition CVPR* (2010).

Everingham, M., Van Gool, L., Williams, C .K. I., Winn, J., & Zisserman, A.: The PASCAL Visual Object Classes Challenge VOC (2008) results. http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html. Accessed 29 Oct 2014.

Felzenszwalb, P., Girshick, R., McAllester, D., & Ramanan, D.: Object detection with discriminatively trained part based models. In: *Proceedings of the IEEE transactions on Pattern Analysis and Machine Intelligence* (2009).

Geiger, A., Wojek, C., & Urtasun, R.: Joint 3d estimation of objects and scene layout. In: *Proceedings of the Advances in Neural Information Processing Systems NIPS* (2011).

Gould, S., Gao, T., & Koller, D.: Region-based segmentation and object detection. In: *Proceedings of the Advances in Neural Information Processing Systems NIPS* (2009).

Gould, S., Rodgers, J., Cohen, D., Elidan, G., & Koller, D. (2008). Multi-class segmentation with relative location prior. *International Journal of Computer Vision*, *80*(3), 300–316.

Guo, R., & Hoiem, D.: Beyond the line of sight: Labeling the underlying surfaces. In: *Proceedings of the 12th European conference on Computer Vision ECCV* (2012).

Guo, R., & Hoiem, D.: Support surface prediction in indoor scenes. In: *Proceedings of the IEEE International Conference on Computer Vision ICCV* (2013).

Gupta, A., Efros, A. A., & Hebert, M.: Blocks world revisited: Image understanding using qualitative geometry and mechanics. In: *Proceedings of the 11th European Conference on Computer Vision ECCV* (2010).

Hedau, V., Hoiem, D., & Forsyth, D.: Recovering the spatial layout of cluttered rooms. In: *Proceedings of the IEEE 12th International Computer Vision ICCV* (2009).

Hoiem, D., Efros, A. A., & Hebert, M. (2007). Recovering surface layout from an image. *International Journal of Computer Vision*, *75*(1), 151–172.

Hoiem, D., Efros, A. A., & Hebert, M.: Closing the loop on scene interpretation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition CVPR* (2008).

Isola, P., & Liu, C.: Scene collaging: analysis and synthesis of natural images with semantic layers. In: *Proceedings of the IEEE International Conference on Computer Vision ICCV* (2013).

Khosla, A., An, B., Lim, J. J., & Torralba, A.: Looking beyond the visible scene. In: *Proceedings of the International Conference on Computer Vision CVPR* (2014).

Kolmogorov, V., & Zabih, R. (2004). What energy functions can be minimized via graph cuts? *Pattern Analysis and Machine Intelligence*, *26*(2), 147–159.

Lee, D. C., Hebert, M., & Kanade, T.: Geometric reasoning for single image structure recovery. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition CVPR* (2009).

Li, C., Kowdle, A., Saxena, A., & Chen, T.: Towards holistic scene understanding: Feedback enabled cascaded classification models. In: *Proceedings of the Advances in Neural Information Processing Systems NIPS* (2010).

Liu, C., Yuen, J., & Torralba, A. (2011). Nonparametric scene parsing via label transfer. *Pattern Analysis and Machine Intelligence*, *33*(12), 2368–2382.

Malisiewicz, T., & Efros, A. A.: Beyond categories: The visual memex model for reasoning about object relationships. In: *Advances in Neural Information Processing Systems NIPS* (2009).

Ross, S., Munoz, D., Hebert, M., & Bagnell, J. A. D.: Learning message-passing inference machines for structured prediction. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition CVPR* (2011).

Russell, B. C., Torralba, A., Murphy, K. P., & Freeman, W. T. (2005). *LabelMe: A database and web-based tool for image annotation*. Technical Report, MIT.

Shotton, J., Winn, J., Rother, C., & Criminisi, A.: Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In: *Proceedings of the 9th European conference on Computer Vision ECCV* (2006).

Silberman, N., Hoiem, D., Kohli, P., & Fergus, R.: Indoor segmentation and support inference from rgbd images. In: *Proceedings of the 12th European Conference on Computer Vision ECCV*, pp. 746–760 (2012).

Silberman, N., Shapira, L., Gal, R., & Kohli, P.: A contour completion model for augmenting surface reconstructions. In: *Proceedings of the European Conference on Computer Vision ECCV* (2014).

Tighe, J., & Lazebnik, S.: Superparsing: Scalable nonparametric image parsing with superpixels. In: *Proceedings of the European Conference on Computer Vision ECCV* (2010).

Tu, Z., & Bai, X. (2010). Auto-context and its application to high-level vision tasks and 3D brain image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(10), 1744–1757.

Zhang, H., Xiao, J., & Quan, L.: Supervised label transfer for semantic segmentation of street scenes. In: *Proceedings of the 11th European Conference on Computer Vision ECCV* (2010).