Object Detection with Geometrical Context Feedback Loop

Min Sun*
sunmin@umich.edu
Sid Ying-Ze Bao*
yingze@eecs.umich.edu
Silvio Savarese
silvio@eecs.umich.edu

Department of EELS
University of Michigan
Ann Arbor, USA
*indicates equal contributions

Abstract

We propose a new coherent framework for joint object detection, 3D layout estimation, and object supporting region segmentation from a single image. Our approach is based on the mutual interactions among three novel modules: i) object detector; ii) scene 3D layout estimator; iii) object supporting region segmenter. The interactions between such modules capture the contextual geometrical relationship between objects, the physical space including these objects, and the observer. An important property of our algorithm is that the object detector module is capable of adaptively changing its confidence in establishing whether a certain region of interest contains an object (or not) as new evidence is gathered about the scene layout. This enables an iterative estimation procedure where the detector becomes more and more accurate as additional evidence about a specific scene becomes available. Extensive quantitative and qualitative experiments are conducted on a new in-house dataset [27] and two publicly available datasets [17, 25], and demonstrate competitive object detection, 3D layout estimation, and object supporting region segmentation results.

1 Introduction

As more and more powerful object recognition methodologies become available [4, 7, 8, 10, 11, 13, 19, 23, 24, 26, 28], increasing attention has been devoted to the design of algorithms that go beyond the individual object detection problem and seek to coherently interpret complex scenes such as the one in the center of Fig. 1. Coherent scene interpretation requires the joint identification of object semantic labels (object classification), the estimation of object 2D/3D location in the physical scene space (2D object localization, depth inference) as well as the estimation of the geometrical structure of the physical space in relationship with the observer. The latter includes the 3D geometry of supporting planes (horizon lines, 3D orientation, vanishing points) and 2D extent of the supporting planes (ground plane region segmentation). Researchers have recognized the value of contextual reasoning as an important tool for achieving coherent scene understanding. Semantic context captures the typical semantic relationship among object classes co-occurring in the same scene category [14, 20, 21, 25] (e.g. cars and roads are likely to co-occur within an urban scene). Geometrical context captures typical spatial and geometrical relationships between object classes and

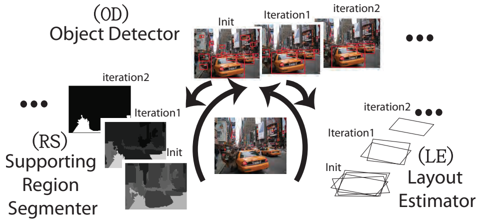
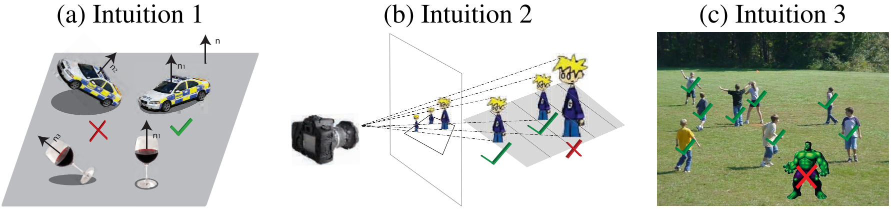


Figure 1: The Context Feedback Loop. We demonstrate that scene layout estimation and object detection can be part of a joint inference process. In this process a supporting region segmentation module (RS) and a scene layout estimation module (LE) provides evidence so as to improve the accuracy of an object detector module (OD). In turn, the OD module enables a more robust estimation of the scene layout (supporting planes orientation, camera viewing angle) and improves the localization of the supporting regions.

the scene geometric structure (e.g., a car is likely to be located on *top* of the road and unlikely to float in the air). These relationships were explored by [1, 12, 17, 18]. In this work, we present a new way to establish the contextual relationship between objects and the scene geometric structure. Specifically, we are interested in modeling the relationship between: i) **objects and their supporting surface geometry**: Geometrical configuration of objects in space is tightly connected with the geometry (orientation) of the surfaces holding these objects (Fig. 2 - Intuition 1); ii) **objects and observer geometry**: Object appearance properties such as the scale and pose are directly related to the observer intrinsic (focal length) and extrinsic properties (camera pose and location) (Fig. 2 - Intuition 2); iii) **objects and supporting regions**: The statistics describing the 2D appearance (features, texture, etc...) of foreground objects are different from those describing the 2D appearance of the supporting surfaces (Fig. 2 - Intuition 3).

Following these intuitions, our work's main contributions are: **1.** A new coherent framework to model contextual reasoning for object detection, 3D layout estimation, and object supporting region segmentation, which is based on the mutual interactions among three modules: i) object detector; ii) scene 3D layout estimator; iii) object supporting region segmenter (Fig. 1). The interaction between such modules captures the contextual relationships discussed above. **2.** Our approach leverages the information returned by the detector (i.e, class label, object location, scale, and pose) in order to establish such contextual relationship. Thus, it does not require using external holistic or local surface detectors [15, 16] or explicit 3D data [2, 3]. **3.** Unlike other methods [14, 20, 21, 25] where the typical co-occurrence between objects and background (e.g., a car on road) is learnt during a training stage and used to provide semantic context, our method exploits the local appearance coherency of objects and supporting surfaces (within a specific image) as well as the typical joint spatial arrangement of objects and supporting surfaces in order to reinforce (or weaken) the presence of objects and segment the object from its supporting surface. **4.** The estimation of the scene 3D layout (orientation and location of the supporting planes, location of objects in 3D and camera parameters (focal length)) is carried out from just one un-calibrated single image. **5.** Most importantly, we introduce a new paradigm where the object detector module is capable of adaptively changing the confidence in establishing whether a certain region of interest contains an object (or not) as new evidence is gathered from the plane 3D layout estimator and supporting region segmenter. Our method is conceptually different from other methods such as [3, 18] where geometric context only modifies the confidence of the object detector *a posteriori* (i.e., the detector always produces the same confidence output which is subsequently modified by a geometric context module). This enables an iterative estimation procedure where the detector *itself* becomes more and more accurate as additional evidence about a specific scene becomes available.

We validated our method against a new in-house dataset [27] (so as to test the system


(a) Intuition 1

(b) Intuition 2

(c) Intuition 3

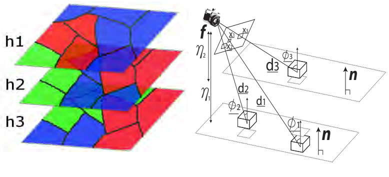
Figure 2: List of intuitions in our paper and comparison with related works. (a) Intuition 1: Rigid objects typically lie up-right on the supporting plane. The coherence between object pose and plane normal is used by our algorithm [1], but not in [12, 17, 18]. (b) Intuition 2: Under the perspective camera model, the size of an object in the 2D image is an inversely proportional function of its distance to the camera when the object pose is fixed. [17, 18] use this relationship too. (c) Intuition 3: The statistics describing the 2D appearance (features, texture, etc...) of foreground objects are different enough from those describing the 2D appearance of the supporting surfaces. Unlike [14, 20, 21, 25] where the typical co-occurrence between objects and background is used to provide semantic context, we exploit the local appearance coherency of objects and supporting surfaces (within a specific image) as well as the typical joint spatial arrangement of objects and supporting surfaces.

level properties of our framework) as well as on existing databases (viz. Label-Me [22] and Office [25] datasets). The experiments demonstrate that our method: i) is scalable to generic scenes (indoors, outdoors) and object categories; ii) achieves state-of-the-art detection results; iii) can successfully infer scene 3D layout information and reason about supporting regions from a single image in challenging and cluttered scenes.

The rest of this paper is organized as follows: in section 2, we first describe in detail our object detector, 3D layout estimator, and object supporting region segmenter modules; we then summarize the type of interactions we aim to model. In section 3, we show quantitative and qualitative experimental results on three different datasets. Finally, we draw conclusions in section 4.

2 Geometrical Context Feedback Loop

In this section we describe in detail the object detector (OD), layout estimator (LE), object supporting region segmenter (RS) modules, and how they act in a coherent fashion (Fig. 1). The critical building block of our system is the object detector as it generates cues (object scale, location, and pose) that can be fed to the LE and RS modules. We use a novel detector called Depth-Encoded-Hough-Voting (DEHV) which is based on our own work [27]. DEHV has the crucial capability to produce an object detection confidence score which is not just a function of the image local appearance but also a function of the geometric structure of the scene. This is related to the object's likely scale and pose, and the object supporting region (if these pieces of information are available). At the beginning of the inference process (iteration 1 of the loop), no information about object scale, pose, and object supporting region is available so the detector returns a number of detection hypotheses by exploring the complete scale space, all possible object poses, and all background/foreground configurations in the image. Each detection hypothesis is associated to the object location, scale (bounding box) and pose (zenith and azimuth angles). This information is fed to both the LE and RS modules. In turn, the LE module produces an estimate of the layout (that is, local orientation of the object supporting planes, camera pose and focal length). By following intuitions 1 and 2 (Fig. 2), this can be done if at least three objects are detected in the image (see the LE modules for details). Moreover, as we shall see in the RS module, using the object's location and the bounding box provided by the detector, the RS module returns segmentation labels



(a) (b)

Figure 3: Panel (a) illustrates the concept of multiple segmentation hypotheses. Here we show three segmentation hypotheses, where each color indicates a region, and the image is partitioned into 9 superpixels indicated by the dark boundaries. Panel (b) Illustrates the notations used in the LB module. The bold italic fonts indicate parameters that are estimated by the LE module. The underline fonts indicate parameters that are estimated by the OD module. In this example, two planes are visualized.

which allow us to identify the region surrounding or below the detected object. Following intuition 3 (Fig. 2), this can be done by using a superpixel representation to capture local appearance coherency of objects and supporting surfaces and by exploiting the typical joint spatial arrangement of objects (whose location and bounding box is given by the detector) and supporting regions in the image. In turn, the outputs for the LE and RS modules are fed back to the OD module and are used to help reduce the detector's search space. Specifically, local orientation of the supporting planes, camera pose and focal length (returned by LE) simplify the complexity of the scale and pose search space. Moreover, the estimation of the object supporting surface (returned by RS) helps remove spurious patches (features) that are used to build the hough voting score in the DEHV. Overall, the detector leverages these additional pieces of evidence to increase the confidence of true positives and decrease that of false alarms.

Object Detector Module (OD). We employ a modified version of the Depth-Encoded-Hough-Voting (DEHV) object categorical detector [27]. Similar to [19], the DEHV detector constructs a voting space $V(O,x|D)$ (Eq. 1), where O is object class (i.e. an object category with a unique pose), x is the object's 2D image location and size (i.e. a 2D bounding boxes enclosing the object), and D is the depth information (i.e, the distance from the camera to the object). The voting space V is constructed by collecting probabilistic votes cast by the set of patches describing object class O. Notice that the voting space $V(O,x|D)$ depends on the geometric structure of the scene since the object hypothesis (O,x) is related to D. This novel feature gives DEHV the ability to detect objects whose locations and poses are compatible with the underlying layout of the scene.

Let $\{C_j, d_j^p, l_j\}$ be a set of patch attributes, where C_j denotes the appearance of image patch j centered at image location l_j, and d_j^p denotes the distance from the camera center to the corresponding 3D location of a patch. Appearance C_j is modeled by a codeword label [5]. Notice that each patch is associated with a physical 3D distance to the camera which affects the size of the patch in 2D. We define $V(O,x|D)$ as the sum of individual contribution over all 3D geometrically consistent images patches, i.e,

$$V(O,x|D) \propto \sum_j p(x|O,C_j,d_j^p,l_j)p(O|C_j)p(d_j^p|l_j) \tag{1}$$

The first term $p(x|O,C_j,d_j^p,l_j)$ characterizes the distribution of object location x given the predicted object class O and patch attributes $\{C_j, d_j^p, l_j\}$. The second term, $p(O|C_j)$ captures the probability that each codeword belongs to an object class O. Finally, $p(d_j^p|l_j)$ models the uncertainty of the depth information of patch j. For details on how to learn $p(x|O,C_j,d_j^p,l_j)$ and $p(O|C_j)$ see [27]. $p(d_j^p|l_j)$ can be initialized as a uniform distribution (no depth information is available at the beginning of the loop).

One of the main contributions of this paper is that the detector can modify its behavior

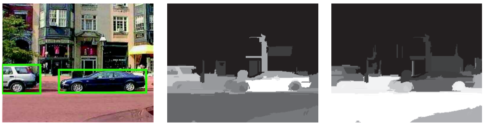

(a) Ori. Img. (b) Avg. S1 (c) Avg. S2

Figure 4: Illustration of the segmentation statistics. Panel (a) shows the original image overlaid with ground truth supporting region (red) and ground truth object bounding boxes (green). Panel (b) and (c) show the average statistics over multiple segmentation hypotheses for S1, S2, respectively. Notice that white indicates higher value.

as knowledge about the scene layout and the object supporting plane are available. The RS module provides knowledge about the supporting region and affects $p(O|C_j)$. The LE module provides knowledge about the scene layout and affects $p(d_j^p|l_j)$. We replace $p(O|C_j)$ with $p(O|C_j, fg_j, \mathbf{S}) = p(O|C_j)p(fg_j|\mathbf{S})$, where $p(fg_j|\mathbf{S})$ is the probability that the image patch j comes from a foreground region (i.e, $fg_j = 1$) and we define $\mathbf{S}$ as a collection of probabilities for each pixel belonging to a supporting region averaged over a number of segmentation hypotheses (see RS module for details). Thus, in the first iteration, $p(fg_j|\mathbf{S})$ is one for every image patch since the 2D extent of the supporting region is unknown. After the first iteration, $p(fg_j|\mathbf{S})$ can be computed as the average probability that each pixel within the image patch j does not belong to a supporting region. This probability is estimated by RS and allows to reduce the importance of patches that are likely to belong to the supporting region.

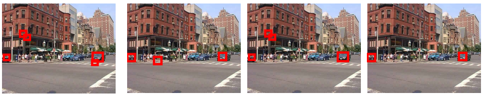
Similarly, in order to explicitly incorporate knowledge about the scene layout, the term $p(d_j^p|l_j)$ is calculated as follows:

$$p(d_j^p|l_j, \mathbf{L}) \propto \sum_{i \in |L|} \delta(t_j^i) \tag{2}$$

where $\mathbf{L}$ captures the scene layout and is defined as $(\{L_i = (n, \eta)\}, f)$; and t_j^i is the distance from the 3D location of the image patch j to the ith plane parameterized by its normal direction n and camera height η (See LE module for detail). Notice that, in the first iteration, $p(d_j^p|l_j, \mathbf{L})$ is uniformly distributed among a finite range of depths. After the first iteration, knowledge about the layout allows to estimate the probability that image patch j is located at depth d_j^p from the camera. Overall, more accurate estimates of $p(d_j^p|l_j, \mathbf{L})$ and $p(O|C_j, fg_j, \mathbf{S})$ will result in reducing the noise in the voting space $V(O, x|D)$ and, in turn, reducing the number of false detections produced by the detector.

To summarize, DEHV takes into account the supporting region segmentation S by removing votes from background image patches to the space $V(O, x|D)$. Furthermore, the uncertainty of the corresponding depth d_j^p of each image patch j is reduced by iteratively estimating the underlying surface layout $\mathbf{L}$. Hence, the noise in the voting space $V(O, x|D)$ is reduced and the number of false detections decreases. Notice that the detection hypotheses $\{O, x\}$ may also be further pruned by checking if the object bounding box x is consistent with the underlying layout information L (similarly to [17]). Similar to [1], the detector is modified to return an estimate of the depth d^o (distance) of each candidate object along with an estimate of the object zenith pose ϕ (Fig. 3 (b)). This information is used by the LE module.

3D Layout Estimator Module (LE). The goal of the 3D layout estimator is to estimate the 3D layout $\mathbf{L}$ associated with a single image from candidate object detections. Our LE module is built upon [1] and uses Hough voting to robustly gather evidence for the 3D layout. As shown in Fig. 3 (b), $\mathbf{L}$ contains the camera focal length f and a set of supporting planes $\{L_i\}$ each parameterized by camera height η and 3D orientation n, where



(a) OD (b) OD+LE (c) OD+RS (d) Full sys.

Figure 5: Interactions between different modules contribute to improve the detection performance. Panels show the results of the baseline detection (a), joint detection and 3D layout estimation (b), joint detection and supporting region segmentation (c), and our full system (d).

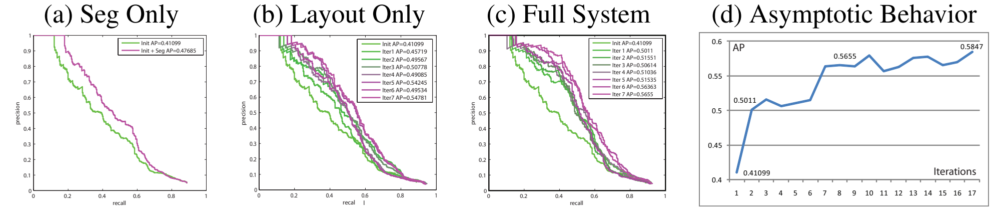
(n,η) specifies a unique plane in 3D[1]. Notice that the orientation n is a normalized vector such that $\|n\|_2 = \sqrt{n_1^2+n_2^2+n_3^2} = 1$. We use intuitions 1 and 2, and formulate the plane estimation problem as a probabilistic hough-voting problem. Each candidate object detection $\left(O_j, x_j, \phi_j, d_j^o\right)$ casts votes for a set of camera focal length $\{f\}$ and supporting plane $\{n,\eta\}$. A hough voting space Q is constructed with axes associated with the plane's orientation n, the camera height h, and the focal length f. The individual contributions over all candidate object detections $\{O, x, \phi, d^o\}$ are accumulated onto the voting space $Q(n,\eta,f) = \sum_j p(n,\eta,f|O_j,x_j,\phi_j,d_j^o)p(O_j,x_j,\phi_j,d_j^o)$, where $p(O_j,x_j,\phi_j,d_j^o)$ is the normalized object detection score $V(O_j,x_j)$ from the detector, and $p(n,\eta,f|O_j,x_j,\phi_j,d_j^o)$ is the probability that the jth object detection $\{O_j,x_j,\phi_j,d_j^o\}$ is supported by different planes parameterized by (n,f,η). See technical report (http://www.eecs.umich.edu/~sunmin) for details on how $p(n,\eta,f|O_j,x_j,\phi_j,d_j^o)$ is computed.

As a result, geometrically consistent detections contribute to high peaks in the layout voting space Q and each peak is associated to a possible supporting plane (n,η) holding objects in the scene. Notice that the layout is estimated from noisy detections and that multiple supporting planes can be nicely handled by our framework. Afterwards, the estimated layout $\mathbf{L} = (\{n,\eta\},f)$ is fed to the detector to further reduce the uncertainty of the patches' depth distribution $p(d_j^p|l_j)$, as already described in Eq. 2.

Supporting Region Segmenter Module (RS). Following the observation that the supporting region is likely to have consistent appearance in the surrounding of the object and following intuition 3, our RS module is capable of segmenting out the object from its supporting surface. We use a superpixel decomposition to identify regions with consistent appearance. Specifically, we adopt the multiple segmentation approach [16], to decompose the images into multiple groups of regions. Thus the image is first grouped into superpixels [9], then the multiple segmentation hypotheses $H = \{h_j\}$ of groups of superpixels is generated similarly to [16] (Fig. 3 (a)), where a group of superpixels defines a region r. We model the intuition 3 by introducing the statistics described below. Such statistics capture the joint typical spatial arrangement of objects (whose location and bounding box are given by the detector) and the object supporting regions in the image. Using these statistics, superpixels can be eventually labeled as supporting regions or not. Based on the candidate object detections $\{x,O\}$, our statistics are: **S1**: The median detection confidence of those candidate object detections that sufficiently overlap with a candidate supporting region[2]. Intuitively, the lower the statistic, the likelier the region belongs to a supporting region (Fig. 4 (b)). **S2**: The 95th percentile of the detection confidence of the candidate object detections supported by the image region. Intuitively, the higher the statistic, the likelier the region belongs to a supporting region (Fig. 4 (c)). Using the designed statistics, we train a logistic regression classifier to predict

[1] A 3D point $q \in R^3$ lies on the plane if $q^T n = h$

[2] When the area of the intersection between the foreground region (fg) and the object bounding box over the area of the object bounding box is bigger than 0.5, the object is considered as sufficient overlap with the foreground region.


(a) Seg Only

(b) Layout Only

(c) Full System

(d) Asymptotic Behavior

Figure 6: Detection performance using precision-recall measurement. Panel (a) compares baseline detection results with our system using only supporting region segmentation. Notice that joint object detection and supporting region segmentation lead to a one-time improvement only. Panel (b) shows results combining detector and layout estimator for 7 iterations. Panel (c) shows the results when all modules (OD, LE, RS) are used in the loop for 7 iterations. Panel (d) shows that the performance of our full system asymptotically converges to a steady state.

the probability $P(y|r,\{x,O\})$ which captures how likely the region r, which contains a group of superpixels, belongs to a supporting region (i.e. $y=1$) or not. By averaging out the contribution of each segmentation hypothesis, we obtain the probability of a superpixel i belonging to a supporting region (i.e, $P(y_i|\{x,O\},I) = \sum_j^n P(y_i|h_j(i),\{x,O\})P(h_j(i)|I)$). [3] Notice, I is the image, $h_j(i)$ is the image region including the ith superpixel in the jth segmentation hypothesis. Notice that the output of the logistic regression $P(y_i|h_j(i),\{x,O\})$ is weighted by $P(h_j(i)|I)$ which indicates the probability that $h_j(i)$ is an image region with the same class. Given the probability that each superpixel belongs to a supporting region $P(y_i|\{x,O\},I)$, and the mapping between pixel index to superpixel index, we obtain the probability (confidence) s that each pixel belongs to a supporting region. Finally, we denote by $\mathbf{S}$ the collection of probabilities $\{s_1,s_2,\dots\}$ for all pixels in the image. This allows to calculate the probability $p(fg|S)$ that an image patch does not belong to the supporting region.

Context Feedback Loop. Our framework starts from only object detection and pose estimation results. Hence, the OD returns the first set of candidate results without using any prior information about the scene 3D layout and supporting region segmentation (Fig. 5 (a)). Given the initial, possibly noisy, detections and pose estimations, the LE generates an estimation of the possible layout parameters $\mathbf{L}$ which can be further used to improve detection (Fig. 5 (b)). Similarly, the RS takes the noisy detection results to estimate the likely location of the supporting region which can be further used to improve detection (Fig. 5 (c)). In practice, LE and RS act simultaneously and contribute to improve detection accuracy which in turn yield more accurate layout and supporting region estimates (Fig. 5 (d)). The system gradually converges into a steady state where the final object detection, pose estimation, layout estimation, and supporting region segmentation results are consistent with each other. Although we do not have a theoretical proof of convergence, experimental results suggest that such a point of convergence exists in most cases.

3 Experiment

We evaluate quantitatively and qualitatively our system on three datasets. The first dataset is the in-house table-top object dataset [27] with ground truth depth and foreground/background segmentation. We conduct experiments on object detection, plane layout estimation, and supporting region segmentation. We also evaluate our system on two publicly available

[3] $P(y_i|h_j(i),\{x,O\}) = P(r^j_{h_j(i)}|\{x,O\})$,where $r^j_{h_j(i)}$ is the region containing the ith superpixel in the jth hypothesis.

Loop Iterations	e_n (radius)	e_η (%)	e_f (%)	e_{seg}^{FA} (%)	e_{seg}^{MS} (%)
First Loop	0.125	25.9	13.2	2.07	51.2
Final Loop	0.118	21.2	11.7	1.79	56.9

Table 1: Estimation errors of surface layout parameters (n, η, f), and supporting regions. The first three columns show the errors of the estimated surface normal e_n, camera height e_η, and surface normal e_f. Each of the errors are defined as follows: $e_n = \arccos(n_{est} n_{gt})$, $e_\eta = \frac{\|\eta_{est} - \eta_{gt}\|}{\eta_{est}}$, and $e_f = \frac{\|f_{est} - f_{gt}\|}{f_{est}}$, where subscript labels *est* and *gt* indicate estimated and ground truth values respectively. The last two columns reports two types of segmentation errors: e_{seg}^{FA} and e_{seg}^{MS} are the amount to which the segmenter mistakenly predicts a foreground region as supporting region and the segmenter misses the truth supporting region, respectively. In detail, let I_P denote the supporting region predicted by our model, I_{SR} denote the ground-truth supporting region, and I_F denote the ground truth foreground objects. We define $e_{seg}^{FA} = \frac{|I_P \bigcap I_F|}{|I_F|}$ and $e_{seg}^{MS} = \frac{|I_P \bigcap I_{RS}|}{|I_{RS}|}$, where $|\bullet|$ counts the pixel number. The smaller e_{seg}^{FA}, the lower the false alarm rate is for confusing foreground pixels as background. The higher e_{seg}^{MS}, the larger the area our algorithm can classify as supporting region. All five types of errors are further reduced as the number of iterations increases (the table reports results for the 1st and 7th iteration).

datasets (Labelme) [22] (so as to compare our performance with the state-of-the-art method [17]) and the office dataset [25]. Anecdotal results on these 3 datasets are shown in Fig. 8.

Table-top Object Dataset. We test our system on the in-house table-top object dataset proposed in [27] which contains three common table-top object categories: mice, mugs, and staplers, where each image comes with depth information collected using a structure-light stereo camera. This allows us to obtain the ground truth 3D layout and supporting plane segmentation. We follow the training procedure described in [27] to train the DEHV detector using 200 images with their corresponding 3D information. The remaining 80 images are used for testing. Each image from either training or testing sets contains $3 \sim 5$ object instances in random poses and locations[4]. During the testing stage, we only use 2D images and all the 3D information is inferred by our algorithm. Fig. 6 shows the overall Precision Recall curve (i.e, combining three classes). Table 1 further shows the accuracy in estimating the layout parameters (n, η, f) and segmenting the supporting region. Both Table 1 and Fig. 6 (d) demonstrate that the feedback loop is effective in improving i) object detection performance, ii) 3D layout estimation and supporting region segmentation.

Label-Me Outdoor Dataset. We compare our system with another state-of-the-art method [17] that uses geometrical contextual reasoning for improving object detection rates and estimating scene geometrical properties such as the horizon line. The experiment is conducted on ~ 100 images that include at least 3 cars in any single image from Label-Me dataset provided by [17] [5]. The training images for our detector are extracted from Pascal 2007 cars training set [6]. Fig. 7 (a) compares the detection performance of our full model at different iterations with [17]. Similar to [18], our work shows that geometric context provides high-level cues to iteratively improve detection performance. Notice that our algorithm: i) does not require the estimation of horizontal or vertical planes as it extracts spatial contextual information from the object itself (enabling our algorithm to work even if the ground plane is not visible at all); ii) it works even if objects are supported by multiple planes located at different heights with respect to the camera. We further evaluate the performance of our proposed supporting region segmenter in Fig. 7 (b). The detection performance using our segmenter (AP=27.6%) is comparable to performance (AP=28.8%) using the supporting region provided by [17] and ground truth bounding boxes. Our proposed segmenter is also flexible in that it can easily incorporate ground plane segmentation results provided by [17]

[4] The training instances and testing instances are separated.

[5] As explained in [1], at least 3 objects are necessary for estimating the layout.

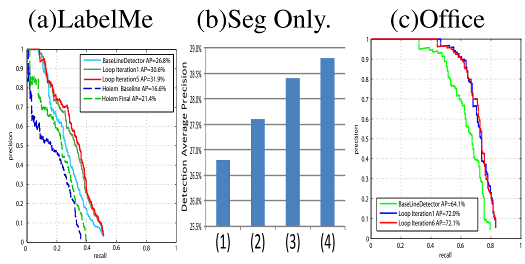


Figure 7: Detection performance using precision-recall measurement. Panel (a) shows the results after applying the full system from iteration 1 to 5 on labelme dataset. Panel (b) shows average detection precision (on LabelMe) using (1) the baseline detector, (2) our supporting RS module, (3) supporting regions provided by [16] as an additional cue to our RS module, (4) supporting regions provided by [16] plus ground truth object bounding boxes. Panel (c) shows the results using full system form iteration 1 to 6 on the office dataset [25]. Notice the improvement of almost 8%.

as an additional cue. This leads to the best detection performance AP= 28.4%. We further evaluate the performance of our 3D layout estimation algorithm by comparing the estimated vanishing lines (i.e, corresponding to the most confident plane estimated by our full algorithm) with the ground truth vanishing lines. At the first iteration, the relative L_1 error [6] is 6.6%. And at the final (5th) iteration , the relative L_1 error is 4.2%.

Office Dataset. We used the office dataset [25] for additional evaluation. 150 images are randomly selected for training and the remaining 54 images (which contain at least 3 objects of interest) are used for testing. Average overall detection performances for mouse, screen, and keyboard are shown in Fig. 7 (c). Typical examples are shown in the last row of Fig. 8.

4 Conclusion

We have presented a framework for jointly detecting objects, estimating the scene layout and segmenting the supporting surfaces holding these objects. Our approach is built upon an iterative estimation procedure wherein the object detector becomes more and more accurate as evidence about the scene 3D layout and the object supporting regions become available and vice versa. Quantitative and qualitative experimental results on both indoor (table-top [27], [25]) and outdoor [17] dataset demonstrated our theoretical claims.

Acknowledgments. We acknowledge the support of NSF (Grant CNS 0931474) and the Gigascale Systems Research Center, one of six research centers funded under the Focus Center Research Program (FCRP), a Semiconductor Research Corporation Entity. We thank Gary Bradski for supporting the data collection of the table-top object dataset [27].

References

[1] Sid Yingze Bao, Min Sun, and Silvio Savarese. Toward coherent object detection and scenelayout understanding. In *CVPR*, 2010.

[2] Gabriel J. Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. Segmentation and recognition using structure from motion point clouds. In *ECCV*, 2008.

[3] Nico Cornelis, Bastian Leibe, Kurt Cornelis, and Luc Van Gool. 3d city modeling using cognitive loops. In *3DPVT*, 2006.

[6] $e_H = \frac{1}{N}\sum_i |\frac{\widehat{H_i}-H_i}{H_i}|$, where $\widehat{H}_i$ and H_i are the best estimated and ground truth vanishing line.

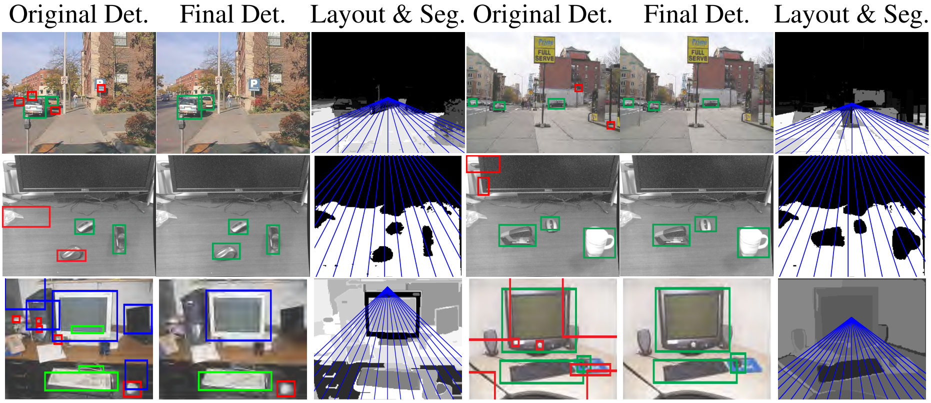


Figure 8: Anecdotal results of joint object detection (green), layout estimation (blue), and original false detections (red). The supporting region is visualized by showing the confidence that a pixel belongs to a supporting region (white indicate high confidence) Results on labelme [17] , table-top [27], and office [25] datasets are shown from row 1 to 3 respectively. Notice that the modules jointly improve the original detection and enable convincing layout estimation and supporting region segmentation results.

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

[5] Chris Dance, Jutta Willamowski, Lixin Fan, Cedric Bray, and Gabriela Csurka. Visual categorization with bags of keypoints. In *ECCV Workshop on Statistical Learning in Computer Vision*, 2004.

[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results.

[7] L. Fei-Fei, R. Fergus, and P. Perona. A Bayesian approach to unsupervised one-Shot learning of object categories. In *ICCV*, 2003.

[8] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008.

[9] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. In *IJCV*, 2004.

[10] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Pictorial structures for object recognition. 2005.

[11] R. Fergus, P. Perona, and A. Zisserman. A sparse object category model for efficient learning and exhaustive recognition. In *CVPR*, 2005.

[12] Stephen Gould, Richard Fulton, and Daphne Koller. Decomposing a scene into geometric and semantically consistent regions. In *ICCV*, 2009.

[13] K. Grauman and T. Darrell. The pyramid match kernel: discriminative classification with sets of image features. In *ICCV*, 2005.

[14] Abhinav Gupta and Larry S. Davis. Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers. In *ECCV*, 2008.

[15] Varsha Hedau, Derek Hoiem, and David Forsyth. Recovering the spatial layout of cluttered rooms. In *ICCV*, 2009.

[16] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Geometric context from a single image. In *ICCV*, 2005.

[17] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Putting objects in perspective. In *CVPR*, 2006.

[18] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Closing the loop on scene interpretation. In *CVPR*, 2008.

[19] Bastian Leibe, Ales Leonardis, and Bernt Schiele. Combined object categorization and segmentation with an implicit shape model. In *ECCV workshop on statistical learning in computer vision*, 2004.

[20] L-J. Li, R. Socher, and L. Fei-Fei. Towards total scene understanding:classification, annotation and segmentation in an automatic framework. In *CVPR*, 2009.

[21] Andrew Rabinovich, Andrea Vedaldi, Carolina Galleguillos, Eric Wiewiora, and Serge Belongie. Objects in context. In *ICCV*, 2007.

[22] Bryan C. Russell, Antonio Torralba, Kevin P. Murphy, and William T. Freeman. Labelme: A database and web-based tool for image annotation. In *IJCV*, 2008.

[23] Silvio Savarese and Li Fei-Fei. 3d generic object categorization, localization and pose estimation. In *CVPR*, 2007.

[24] Hao Su, Min Sun, Li Fei-Fei, and Silvio Savarese. Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories. 2009.

[25] Erik B. Sudderth, Antonio Torralba, William T. Freeman, and Alan S. Willsky. Describing visual scenes using transformed objects and parts. In *IJCV*, 2008.

[26] Min Sun, Hao Su, Silvio Savarese, and Li Fei-Fei. A multi-view probabilistic model for 3d object classes. 2009.

[27] Min Sun, Gary Bradski, Bing-Xin Xu, and Silvio Savarese. Depth-encoded hough voting for coherent object detection, pose estimation, and shape recovery. In *ECCV*, 2010.

[28] Paul Viola and Michael Jones. Robust real-time object detection. In *IJCV*, 2002.