Int J Comput Vis (2012) 100:154–169
DOI 10.1007/s11263-012-0547-2

Object Detection using Geometrical Context Feedback

Min Sun · Sid Yingze Bao · Silvio Savarese

Received: 17 December 2010 / Accepted: 16 July 2012 / Published online: 2 August 2012

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 the table-top dataset (Sun et al. in ECCV, 2010b) and two publicly available datasets (Hoiem et al. in CVPR, 2006; Sudderth et al. in IJCV, 2008), and demonstrate competitive object detection, 3D layout estimation, and segmentation results.

Keywords Scene understanding · Object recognition · Object detection · Focal length estimation · 3D reconstruction · Surface estimation · Viewpoint estimation

Contribution of M. Sun and S.Y. Bao is equal in this paper.

M. Sun (✉) · S.Y. Bao · S. Savarese
University of Michigan, Ann Arbor, MI, USA
e-mail: sunmin@umich.edu

S.Y. Bao
e-mail: yingze@umich.edu

S. Savarese
e-mail: silvio@eecs.umich.edu

1 Introduction

As more and more reliable and accurate object recognition methodologies become available, 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 the supporting surfaces (i.e., orientation and location of the surfaces that are supporting objects in the scene) as well as their 2D extent in the image (supporting surface segmentation).

Researchers have recognized the value of contextual reasoning as an important tool for achieving coherent scene understanding. Two main types of contextual information have been explored: Semantic context and geometrical context. Semantic context captures the typical semantic relationship among object classes co-occurring in the same scene category (Torralba et al. 2003; Li and Fei-Fei 2007; Li et al. 2009; Ladicky et al. 2010; Gonfaus et al. 2010; Rabinovich et al. 2007) (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 the scene geometric structure (Gupta and Davis 2008; Sudderth et al. 2008; Hoiem et al. 2006, 2008; Gould et al. 2009; Hedau et al. 2009; Heitz et al. 2008; Li et al. 2010; Saxena et al. 2009; Bao et al. 2010) (e.g., a car is likely to be located on top of the road and unlikely to float in the air).

In this paper, we present a new way to establish the contextual relationship between objects and the scene geometric

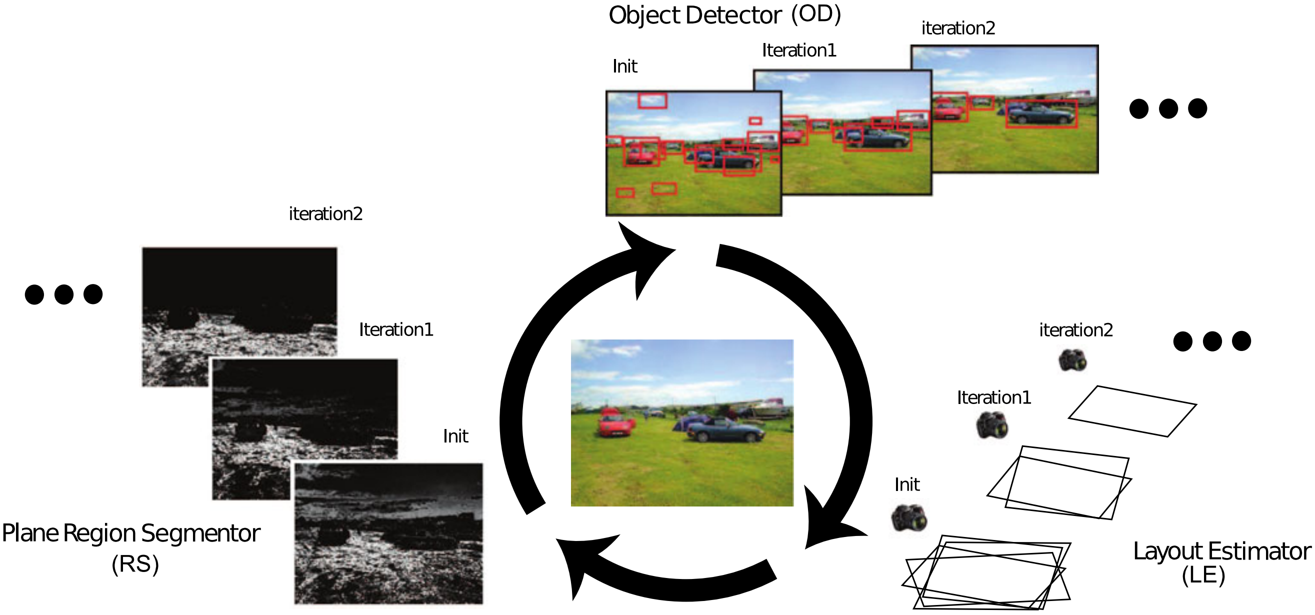


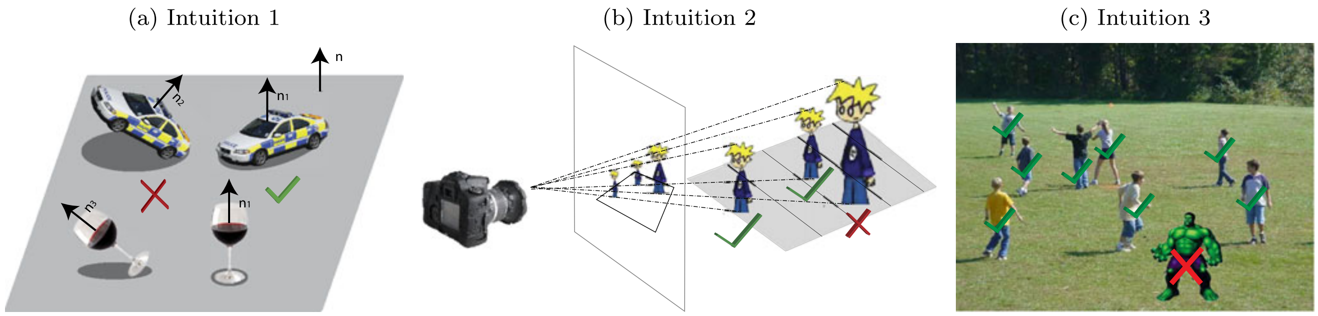
Fig. 1 The Context Feedback Loop. We demonstrate that scene layout estimation and object detection can be part of a joint inference process. In this process a supporting region segmentation module (RS) and a scene layout estimation module (LE) provides evidence so as to improve the accuracy of an object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

structure. Specifically, we are interested in modeling the relationship between:

– **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);
– **objects and observer's geometry.** Object appearance properties such as the scale and pose are directly related to the observer's intrinsic (focal length) and extrinsic properties (camera pose and location) (Fig. 2—Intuition 2);
– **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 interactions between such modules capture the contextual relationships discussed above.
2. Our approach leverages the estimations returned by the detector (i.e., class label, object location, scale, and pose) in order to establish such contextual relationship. Thus, it does not rely on using external holistic or local surface detectors (Hoiem et al. 2005; Hedau et al. 2009) or explicit 3D data (Cornelis et al. 2006; Brostow et al. 2008).
3. Unlike other methods such as Li et al. (2009), Ladicky et al. (2010), Gonfaus et al. (2010), Rabinovich et al. (2007), Li and Fei-Fei (2007), Torralba et al. (2003) 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 to 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. Unlike other methods such as Hoiem et al. (2005), Hedau et al. (2009) wherein assumptions about the relationship between the geometry of the ground plane and the camera parameters are made (e.g., the camera is located at given height from the ground plane and only one ground plane is allowed), our approach can handle multiple supporting planes and arbitrary observer viewing directions.
5. Most importantly, we introduce a new paradigm where the object detector module is capable of adaptively


(a) Intuition 1

(b) Intuition 2

(c) Intuition 3

Fig. 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 as well as our preliminary work (Bao et al. 2010; Sun et al. 2010a), but not in Hoiem et al. (2006), Gould et al. (2009), Hoiem et al. (2008). (**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. Hoiem et al. (2006, 2008) use this relationship too. (**c**) Intuition 3: The statistics describing the 2D appearance (features, texture, etc.) of foreground objects are likely to be different enough from those describing the 2D appearance of the supporting surfaces (e.g., we rarely see *green hulk* playing on grass.). Unlike Rabinovich et al. (2007), Gupta and Davis (2008), Li et al. (2009), Sudderth et al. (2008) 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 (Color figure online)

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 Hoiem et al. (2008), Cornelis et al. (2006) 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.

6. We validated our method against an augmented table-top dataset (Sun et al. 2010b) (so as to test the system level properties of our framework) as well as on existing databases (viz. labelme (Russell et al. 2008) and Office (Sudderth et al. 2008) datasets). The experiments demonstrate that our method: (i) is scalable to generic scenes (indoors, outdoors) and generic 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 Sect. 1.1 we review the related work. In Sect. 2, we first describe in detail the model representation and learning procedure of our object detector, 3D layout estimator, and object supporting region segmenter modules; we then summarize the types of interactions we used during inference. In Sect. 3, we show quantitative and qualitative experimental results on three different datasets. Finally, we draw conclusions in Sect. 4.

1.1 Related Work

In this section we review some of the key methods that use semantic and geometrical contextual cues for enhancing the process of jointly recognizing scene elements and reconstructing the scene layout. Torralba et al. (2003), Li and Fei-Fei (2007), Li et al. (2009), Ladicky et al. (2010), Gonfaus et al. (2010), Rabinovich et al. (2007) leverage semantic context to capture the typical relationship among object classes co-occurring within each image (e.g., cars and roads are likely to co-occur) (Ladicky et al. 2010; Gonfaus et al. 2010; Rabinovich et al. 2007), or between object classes and scene categories (e.g., cars are likely to occur within an urban scene) (Torralba et al. 2003; Li and Fei-Fei 2007; Li et al. 2009). Unlike semantic context, geometrical context captures typical spatial and geometrical relationships between object classes and the scene geometric structure. While Gupta and Davis (2008), Sudderth et al. (2008) propose to model 2D relationships among scene components in the 2D image plane (e.g., a person is likely to be on-top of the ground), Hoiem et al. (2006) investigate the possibility of integrating cues from the 3D scene such as vertical and ground surfaces (Hoiem et al. 2005) into the process of jointly detecting objects and estimating the scene layout. Moreover, Hoiem et al. (2006) use object scale as a critical cue for modeling the interaction between objects and the scene as well as determining the distance (depth) of the objects from the camera. Hoiem et al. (2008), Gould et al. (2009) propose an interactive approach wherein additional cues such as the occluding boundaries between objects and the scene background in the image are injected into the inference procedure. Hedau et al. (2009) models the explicit relationship between the scene layout and objects in 3D using a box representation (i.e. a representation that approximates the scene physical space as a 3D box). Other methods leverage the ability to infer the scene depth maps via probabilistic inference to model spatial relationships among scene elements in 2.5D (Heitz et al. 2008; Li et al. 2010; Saxena et al. 2009; Payet and Todorovic 2011). Finally, Bao et al. (2010) capture the interaction between the object pose and the 3D supporting surfaces to estimate the 3D layout even when cues from the underlying scene (e.g., vanishing lines or scene surface orientations) are not available.

It is clear that the possibility of exploiting geometric context for scene understanding is often coupled with the ability of the object detector to extract object geometrical properties such as position, scale and pose. For example, object location is used to determine 2D relationships such as "below" or "on top" (Gupta and Davis 2008; Sudderth et al. 2008). Object scale is used to determine the relative distance between objects and cameras (Hoiem et al. 2006). Object pose is used to determine the relationship between the object and its supporting surface in 3D (Bao et al. 2010). Unfortunately, most of the methods in the vision literature have focused on just determining object properties such as location and scale (by estimating the object bounding boxes). Notable examples are the 2D models or mixture of 2D models presented by Viola and Jones (2002), Fei-Fei et al. (2003), Felzenszwalb and Huttenlocher (2005), Grauman and Darrell (2005), Leibe et al. (2004), Fergus et al. (2005). Only recently, a number of techniques have been proposed that are capable of capturing the intrinsic 3D nature of object categories and, in turn, of estimating 3D object properties such as the object 3D pose (Thomas et al. 2006; Savarese and Fei-Fei 2007; Sun et al. 2009, 2010b; Su et al. 2009; Liebelt and Schmid 2010). In particular, the work by Sun et al. (2010b) is a key building block for the method presented in this paper in that it has the unique ability to infer both pose, distance and rough 3D shape of the object from a single image.

2 Geometrical Context Feedback Loop

In this section we first give an overview of our model which fuses the information from the object detector (OD), layout estimator (LE), object supporting region segmenter (RS) modules in a coherent fashion (Fig. 1).

2.1 Model Overview

The critical building block of our system is the object detector as it generates cues (e.g., object scale, location, and pose) that can be fed to the layout estimator and the region segmenter modules. We use a novel detector called Depth-Encoded-Hough-Voting (DEHV) which is based on our own work (Sun et al. 2010b). 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 (i.e., the 3D layout information **L** and supporting region information **S**). This information restricts the object's likely scale, pose, and background/foreground configurations. At the beginning of the inference process (iteration 1 of the loop), no information about 3D layout information **L** and supporting region information **S** is available so the detector returns a number of detection hypotheses by exploring all object categories, the complete scale space, all possible object poses, and all background/foreground configurations in the image. Each detection hypothesis is associated to the object class O, location x, scale (1-to-1 mapped to depth d^o (see (2))), and pose ϕ^o (zenith and azimuth angles). This information is fed to both the layout estimator and region segmenter modules. In turn, the layout estimator module produces an estimate of the 3D layout of the scene. The layout information **L** includes the camera focal length f and a set of supporting planes L_i, where L_i is parameterized by camera-to-plane height η and 3D orientation n in the camera reference system. By following intuitions 1 and 2 (Fig. 2), this can be done if at least three objects are detected in the image (proved by Bao et al. 2010, see Appendix B for details). Moreover, as we shall see in the region segmenter module, using the object's location and scale provided by the detector, the region segmenter module returns probability of each pixel belonging to a supporting region $\mathbf{S}(l)$, where l specifies the 2D location of the pixel. This information allows us to identify the extent of the supporting region. 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 scale are given by the detector) and supporting regions in the image. In turn, the outputs for the layout estimator and region segmenter modules are fed back to the object detector module and are used to help reduce the detector's search space (i.e., object scale, location, and pose). Specifically, location and orientation of the supporting planes in the camera reference system, and camera focal length (returned by the layout estimator) simplify the complexity of the scale and pose search space. Moreover, the estimation of the object supporting surface (returned by the region segmenter) 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 following the iterative inference procedure described in Sect. 2.4. An overview of the inference procedure is shown in Algorithm 1.

2.2 Model Representation

We introduce in detail our three modules (object detector, layout estimator, and supporting region segmenter) in this section.

2.2.1 *Object Detector Module*

We employ a modified version of the Depth-Encoded-Hough-Voting (DEHV) object categorical detector (Sun et al. 2010b) to obtain an estimate of the object location, scale, pose, and depth. Similar to Leibe et al. (2004), the DEHV detector constructs a voting space $V(O,x|D)$ (see (1)), where O is object class (i.e. an object category with a unique pose), x is the object's 2D image location and scale (i.e. a 2D bounding boxes enclosing the object), D is the depth information (i.e., the distance from the camera to the object), and different poses are encoded as different object classes. 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 property gives DEHV the ability to detect objects whose locations and poses are compatible with the underlying layout of the scene.

The DEHV Detector 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 a discrete codeword label (Dance et al. 2004). 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.,

$$V(O,x|D) \propto \sum_j p\big(x|O,C_j,d_j^p,l_j\big)p(O|C_j)p\big(d_j^p|l_j\big) \quad (1)$$

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. Please see Appendix A for details of the derivation.

Similar to Sun et al. (2010b), our detector enforces a 1-to-1 mapping m between scale s and depth d for each patch. This way, given the 3D information, our method deterministically selects the scale of the patch at each location l, and given the selected patches, our method can infer the underlying 3D information (Fig. 3). In detail, given the camera focal length f, the corresponding scale s at location $l=(u,v)$ can be computed as $s=m(d,l)$ and the depth d can be inferred from $d=m^{-1}(s,l)$. The mapping m obeys the following relations:

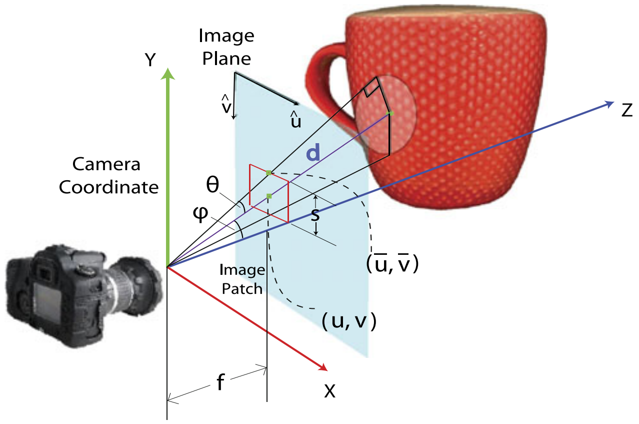


Fig. 3 The physical interpretation of (2). Under the assumption that image patch (*red bounding box*) tightly encloses the 3D sphere with radius r, the patch scale s is directly related to the depth d given camera focal length f and the center $l=(u,v)$ of the image patch. Notice that this is a simplified illustration where the patch center is on the yz plane. This figure is best viewed in color (Color figure online)

$$\begin{aligned} & s = 2(\bar{v} - v); \qquad \bar{v} = \tan(\theta+\varphi)f; \\ & \theta = \arcsin\left(\frac{r}{d_{yz}}\right); \qquad \varphi = \arctan\left(\frac{v}{f}\right) \\ & d_{yz} = \frac{d\sqrt{f^2+v^2}}{\sqrt{u^2+v^2+f^2}} : \quad d \text{ projected onto } yz \text{ plane} \end{aligned} \quad (2)$$

Generating Object Hypotheses After accumulating votes into the Hough voting space $V(O,x|D)$, a set of detection hypotheses $\{(O_i,x_i)\}$ corresponding to peaks in the voting space can be obtained. Given the object class O and the 2D location x, the image patches that cast votes for the hypothesis can be retrieved (later referred as supporting image patches). Hence, the depth to image patch information described in (2) can be used to calculate the depths of all image patches $\{d_j^p\}$. The depth of the object d^o is defined as the median depth of the depths of image patches d_j^p. Similarly, we can also retrieve the corresponding zenith angles $\{\phi_j^p\}$ corresponding to all supporting image patches, and we use a verification support-vector-machine (SVM) classifier to find the most likely zenith angle of the object ϕ^o among the candidate zenith angles. Hence, the final output of the detector is a set of hypotheses $\{(O_i,x_i,\phi_i^o,d_i^o)\}$.

One of the main contributions of this paper is that the detector can modify its behavior as knowledge about the scene layout (denoted by **L**) and the object supporting regions (denoted by **S**) are available.

Knowledge of Supporting Region **S** The region segmenter module provides knowledge about the supporting region **S** and affects $p(O|C_j)$. We replace $p(O|C_j)$ with $p(O|C_j, l_j, \mathbf{S})$, and we show that it can be decomposed as follows,

$$p(O|C_j, l_j, \mathbf{S}) = p(O, O \notin bg|C_j, l_j, \mathbf{S}) \quad (3)$$

$$= p(O|O \notin bg, C_j)p(O \notin bg|C_j, l_j, \mathbf{S}) \quad (4)$$

where $O \notin bg$ means the object class does not belong to the background class. The first equality is true since we only need to evaluate object classes that belong to the foreground object classes during Hough voting. The second equality follows the chain rule in probability theory and conditional independent assumption between $(l_j, \mathbf{S})$ and O given $(O \notin bg, C_j)$. As a result, only the second term $p(O \notin bg|C_j, l_j, \mathbf{S})$ is related to the supporting region **S**. We define $p(O \notin bg|C_j, l_j, \mathbf{S})$ as follows,

$$p(O \notin bg|C_j, l_j, \mathbf{S}) := p(O \notin bg|C_j)\big(1 - \mathbf{S}(l_j)\big) \quad (5)$$

where $p(O \notin bg|C_j)$ is the probability that the codeword C_j does not belong to the background class, and it is reweighed by $1 - \mathbf{S}(l_j)$. Here, $\mathbf{S}(l_j)$ is the probability that a pixel at location l_j belongs to a supporting region which is equivalent to saying that such a pixel belongs to a background region (see the region segmenter module for details). This probability is estimated by the region segmenter and allows the algorithm to reduce the importance of patches that are likely to belong to the supporting region.

Knowledge of Scene Layout **L** The layout estimator module provides knowledge about the scene layout **L** and affects $p(d_j^p|l_j)$. In order to explicitly incorporate knowledge about the scene layout, the term $p(d_j^p|l_j)$ is calculated as follows:

$$p\big(d_j^p|l_j, \mathbf{L}\big) \propto \sum_{i \in |\mathbf{L}|} \delta\big(t_j^i\big) \quad (6)$$

where 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 (7) in section of the layout estimator module for details); and $|\mathbf{L}|$ denotes the number of plane hypotheses. Notice that knowledge of the scene layout **L** allows the algorithm to estimate the probability that an image patch j is located at depth d_j^p from the camera. Hence, effectively, the search space of object scale in 2D is reduced.

To summarize, modified DEHV takes into account the knowledge of supporting region S by deemphasizing votes from supporting regions 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 the knowledge of surface layout **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 Hoiem et al. 2006).

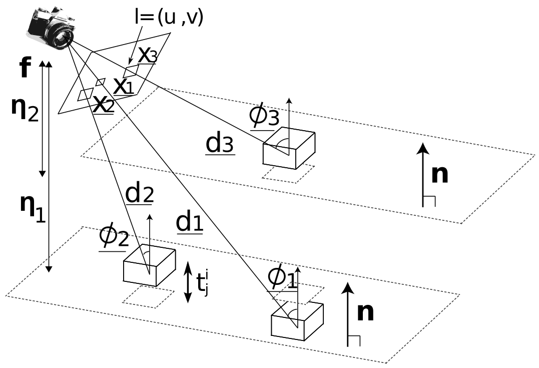


Fig. 4 The notations used in the layout estimator module. The *bold fonts* indicate parameters that are estimated by the layout estimator module. The *underline fonts* indicate parameters that are estimated by the object detector module. In this example, two planes are visualized. The measurements are: x: object's 2D image location and scale; d: object-to-camera distance; ϕ: observed object pose; t: the 3D object center to supporting place distance; l: observed patch image location. The unknowns are: f: camera focal length; n: the plane normal; η: the camera-to-plane distance

2.2.2 3D Layout Estimator Module

The goal of the 3D layout estimator is to estimate the 3D layout **L** associated with a single image from candidate object detections. Our layout estimator module is built upon Bao et al. (2010). However, instead of using the probability inference in Bao et al. (2010), we employ Hough voting to efficiently estimate the 3D layout. As shown in Fig. 4, **L** contains the camera focal length f and a set of supporting planes $\{L_i\}$ each parameterized by camera-to-plane height η and 3D orientation n. Notice that the orientation n is a normalized vector such that $\|n\|_2 = \sqrt{n_1^2 + n_2^2 + n_3^2} = 1$, and (n, η) specifies a unique plane in 3D such that any 3D point $q \in R^3$ lying on the plane satisfies $q^T n = \eta$. Moreover, the closest distance t from a 3D point q to a plane parameterized by (n, η) can be calculated as follows,

$$t = \big|q^T n - \eta\big| \quad (7)$$

The 3D point q corresponding to the 2D point $l = (u, v)$ with depth d is equivalent to the normalized ray from camera center to the 2D point on the image plane times the depth:

$$q = \frac{[u\ v\ f]^T}{\sqrt{u^2 + v^2 + f^2}} \times d \quad (8)$$

Following intuitions 1 and 2, we formulate the plane estimation problem as a Hough-voting problem. A Hough voting space Q is constructed with axes associated with the plane's orientation n, the camera height η, and the focal length f. Each candidate object detection $(O_i, x_i, \phi_i^o, d_i^o)$ casts votes in Q for a set of camera focal length $\{f\}$ and supporting plane $\{n, \eta\}$ following the distribution $p(n, \eta, f | O_i, x_i, \phi_i^o, d_i^o)$. We use geometrical constraints to help compute $p(n, \eta, f | O_i, x_i, \phi_i^o, d_i^o)$. The geometrical relationship relating object detections and the supporting planes is derived in the same manner as Bao et al. (2010). As illustrated in Fig. 4, let (u_i, v_i) be the center location of object detection location x_i. The zenith angle ϕ_i[1] is the angle between the light ray (u_i, v_i, f) from the camera to the object and the plane normal n_1, n_2, n_3. The layout $\{f, \eta, n\}$ and its supporting object satisfies the following equations,

$$\begin{cases} u_i n_1 + v_i n_2 + f n_3 = -\cos(\phi_i) \| u_i \; v_i \; f \|_2 \\ \sqrt{(n_1)^2 + (n_2)^2 + (n_3)^3} = 1 \\ \eta = d_i^o * \cos(\phi_i) \end{cases} \tag{9}$$

Given a candidate object detection $(O_i, x_i, \phi_i^o, d_i^o)$, we compute $p(n, \eta, f | O_i, x_i, \phi_i^o, d_i^o)$ as the following:

$$p\big(n, \eta, f | O_i, x_i, \phi_i^o, d_i^o\big) \propto \begin{cases} 1 & \text{if } (n, \eta, f) \text{ satisfies (9)} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

The final voting space $Q(n, \eta, f)$ is defined as the weighted sum over distribution of each candidate detection as follow,

$$Q(n, \eta, f) = \sum_i p\big(n, \eta, f | O_i, x_i, \phi_i^o, d_i^o\big) V(O_i, x_i) \tag{11}$$

such that the contribution of each candidate detection is weighed by the detection score $V(O_i, x_i)$.

As a result, high values in the layout voting space Q is accumulated by geometrically consistent detection candidates. This model can easily incorporate scene layout with multiple supporting planes by associating each plane to a peak in the Hough voting space Q. However, in order to regularize the co-occurrence of multiple supporting planes, we assume all the supporting planes are parallel to each other similarly to the assumption in Bao et al. (2010). This allow us to compress the Hough voting space to a lower dimension space $\hat{Q}(n, f)$ by summing over the axis of η in $Q(n, \eta, f)$. We first find the peak (n^*, f^*) in $\hat{Q}(n, f)$. Then, we select multiple peaks of $\{\eta\}$ in $Q(n^*, \eta, f^*)$. As shown in Bao et al. (2010), it is necessary to have at least 3 non-collinear object supported by parallel planes to have a unique peak in $\hat{Q}(n, f)$ (see Appendix B for the proof). It is important to point out that the 3 non-collinear objects do not have to be located on the *same* supporting plane. More specifically, since we assume that multiple planes are parallel to each other, we only need at least 3 objects to estimate the plane orientation, and each plane height can be estimated from one single object. Finally, the estimated layout $\mathbf{L} = (\{n, \eta\}, f)$ is fed to the detector to further reduce the uncertainty of the patches' depth distribution $p(d_j^p | l_j, \mathbf{L})$, as already described in (6).

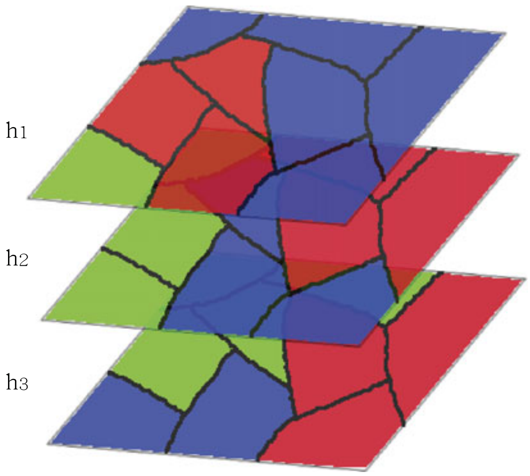


Fig. 5 Illustration of the concept of multiple segmentation hypotheses, where different hypotheses are shown at different layers. Here we show three segmentation hypotheses, where *each color* indicates a region corresponding to a set of superpixels, and the image is partitioned into 9 superpixels separated by the dark boundaries (Color figure online)

2.2.3 Supporting Region Segmenter Module

Following the observation that the supporting region is likely to have consistent appearance in the surrounding of the object and following intuition 3, our region segmenter module is capable of segmenting out the object from its supporting surface. We use a superpixel decomposition method (Felzenszwalb and Huttenlocher 2004) to identify regions with consistent appearance. Similarly to Hoiem et al. (2005), multiple segmentation hypotheses $H = \{h_j\}$ are used to mitigate the problem of segmentation errors. A segmentation hypothesis h_j is an ensemble of disjoint set of superpixels which fully cover the image. Each set of superpixels is a unique region r (Fig. 5).

Given a region $r \in h_j$ from the jth segmentation hypothesis, we train a logistic regression classifier to predict the probability $P(y|r, \{x, O\})$ which captures how likely the region r 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 as follows,

[1] Here we omit the superscript o to have a concise notation.

$$P\big(y_i|\{x,O\},I\big)=\sum_j P\big(y_i h_j(i)|\{x,O\},I\big) \tag{12}$$

$$=\sum_j P\big(y_i|h_j(i),\{x,O\}\big)P\big(h_j(i)|I\big) \tag{13}$$

where I is the image and $h_j(i)$ is the image region including the ith superpixel in the jth segmentation hypothesis h_j. Notice that the output of the logistic regression $P(y_i|h_j(i),\{x,O\})$ is weighed by $P(h_j(i)|I)$ which indicates the probability that $h_j(i)$ is a region containing superpixel i given the image evidence. 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 the algorithm to calculate the probability $p(O\notin bg|C_j,l_j,\mathbf{S})$ in (4) that an image patch does not belong to the background.

2.3 Model Learning

In this section, we describe how the model parameters are learned in the object detector and supporting region segmenter modules in detail.

2.3.1 *Object Detector Module*

Recall that the Hough voting space $V(O,x|D)$ in (1) aggregates votes from each unique combination of image patch location l_j, codeword label C_j, and depth d_j^p. Our goal is to learn the codebook mapping for a codeword C, the distributions of object class $p(O|.)$ (voting weight) and location $p(x|.)$ (voting direction). Notice that computation of $p(d|.)$ is already described in (6).

Similar to Sun et al. (2010b), we assume that for a number of training object instances, the 3D reconstruction D of the object is available. This corresponds to having available the distance (depth) of each object patch from its physical location in 3D.

Here we define location x of an object as a bounding box with center position q, height h, and aspect ratio a. We sample each image patch centered at location l and select the scale $s=m(l,d)$ using the 1-to-1 mapping described in (2). Then the appearance $I(l,s)$ is extracted from the patch (l,s). When the image patch comes from a foreground object, we cache: (1) the information of the relative voting direction b as $\frac{q-l}{s}$; (2) the relative object-height/patch-scale ratio w as $\frac{h}{s}$; (3) the object aspect ratio a.

Random Forest Codebook We use both the foreground patches (positive examples) and background patches (negative examples) to train a random forest discriminative codebook. Hence, the mapping $C(I(l,s))$ is a unique index of the leaf node in the random forest.

Voting Weight $p(O|.)$ For each codeword entry C_j, we use the training data to estimate $p(O|O\notin bg,C_j)$ and $p(O\notin bg|C_j)$ by counting the frequency that patches of O falls in the codebook entry C. Then, we can calculate $p(O|C_j,l_j,\mathbf{S})$ using (4).

Voting Direction $p(x|O,C,s,l)$ can be evaluated given the cached information $\{(b_k,w_k,a_k)\}$ as follows:

$$\begin{aligned}&p(x|O,C,s,l)\\&\quad=p\big((q,h,a)|O,C,s,l\big)\\&\quad\propto\sum_{k\in g(O,C)}\delta(q-b_k\cdot s+l,h-w_k\cdot s,a-a_k)\end{aligned}$$

where $g(O,C)$ is a set of patches from O mapped to codebook entry C. Notice that $p(x|O,C,s,l)$ is equivalent to $p(x|O,C,d^p,l)$ in (1), since $s=m(d^p,l)$.

2.3.2 *Supporting Region Segmenter Module*

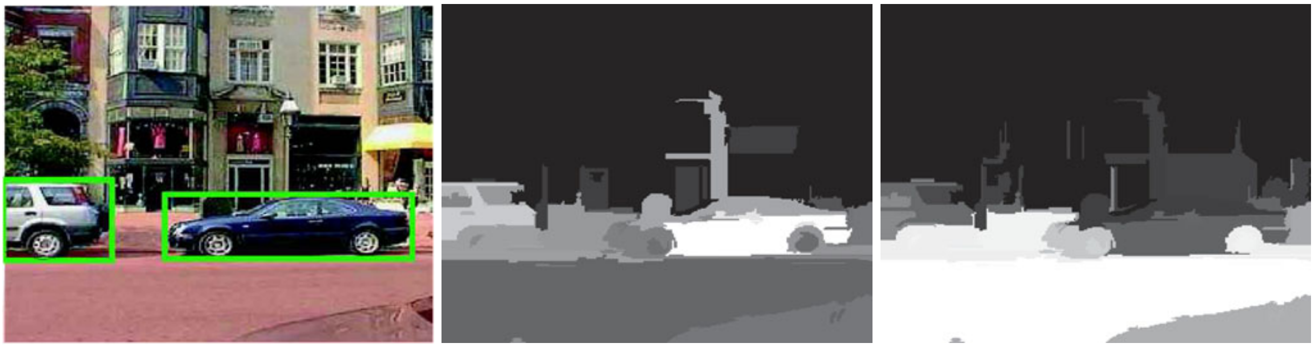
Here, we describe how to learn the probability $P(h_j(i)|I)$ and $P(y_i|h_j(i),\{x,O\})$ in (13).

We model the intuition 3 by introducing the region-based 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, each region can be eventually labeled as supporting regions or not. Based on the candidate object detections $\{x,O\}$, our statistics are:

- The median detection confidence of those candidate object detections that sufficiently overlap with a candidate supporting region.[2] Intuitively speaking, the higher the statistic, the more likely the region belongs to the foreground region and the less likely belongs to a supporting region (Fig. 6(b)).
- 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. 6(c)).

Using the designed statistics for each region r, we train a logistic regression classifier to estimate the probability $P(y|r,\{x,O\})$. Finally, $P(h_j(i)|I)$ is trained similarly to the segment homogeneity classifier described in Hoiem et al. (2007).

[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) Ori. Img.

(b) Avg. S1

(c) Avg. S2

Fig. 6 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 (Color figure online)

Algorithm 1 Context Feedback Loop

```
S := empty
L := empty
for iter ≤ MaxIter do
    $\{(O, x, d^o, \phi^o)\} = \mathrm{OD}(\mathbf{S}, \mathbf{L})$
    $\mathbf{L} = \mathrm{LE}(\{(O, x, d^o, \phi^o)\})$
    $\mathbf{S} = \mathrm{RS}(\{(O, x, d^o, \phi^o)\})$
    iter = iter + 1
end for
```

2.4 Model Inference using Context Feedback Loop

Our inference algorithm (see Algorithm 1) starts by applying the object detector module assuming no prior knowledge about the scene layout **L** and supporting regions **S** is available. Hence, $p(d^p|.)$ in (1) is an uniform distribution which implies image patches can appear at any depth, and $p(O|C)$ in (1) is equal to $p(O|O \notin bg, C_j)p(O \notin bg|C_j)$.

The object detector returns the first set of candidate results $\{(O, x, d, \phi)\}$ (Fig. 7(a)). Given the initial, possibly noisy, detections and pose estimations, the layout estimator generates an estimation of the possible layout parameters **L** which can be further used to improve detection (Fig. 7(b)). Similarly, the region segmenter takes the noisy detection results to estimate the likely location of the supporting region which can be further used to improve detection (Fig. 7(c)). In practice, the layout estimator and region segmenter act simultaneously and contribute to obtain more accurate detections which in turn yields more accurate layout and supporting region estimates (Fig. 7(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.

2.5 Implementation details

For the object detector, we use the following binning in the Hough voting space: (i) 60 scales (from the 0.05 of the original scale multiplied by 1.05 to the original scale); (ii) 10 object aspect ratio (from 0.6113 to 2.1611); (iii) Each object class is discretized into 8 object poses corresponding to different azimuth angles; (iv) For each scale, the object hypothesis is shifted in both horizontal and vertical directions by 2 pixels.

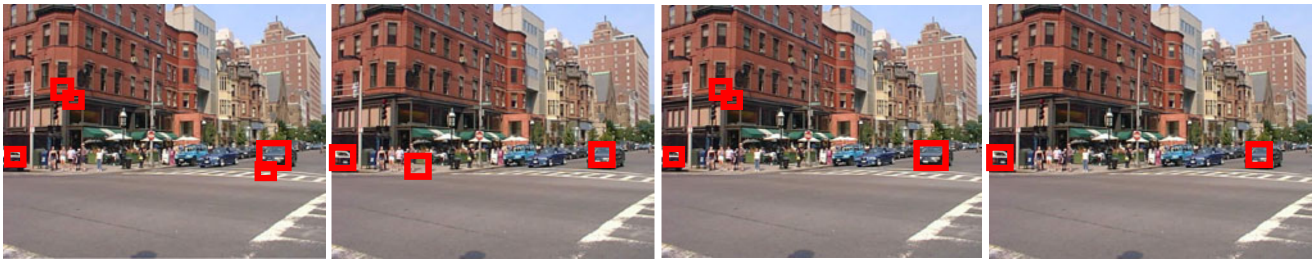
For layout estimator, the binning in the Hough voting process is: (i) plane normal has 20 bins for tilt direction from 15° to 75° and 5 bins for camera-rotation from -10° to 10°, (ii) plane height has 20 bins from 30 cm to 80 cm for office dataset and from 1.5 m to 2 m for street dataset. (iii) camera focal length has 20 bins from 0.8 to 1.25 fraction of an initial camera focal length guess.

3 Experiment

We evaluate quantitatively and qualitatively our system on three datasets. The first dataset is an augmented table-top object dataset (Sun et al. 2010b) 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 datasets: a subset of label-me dataset (Russell et al. 2008) (so as to compare our performance with the state-of-the-art method (Hoiem et al. 2006)) and the office dataset (Sudderth et al. 2008). Typical results on these 3 datasets are shown in Figs. 12, 13, 14.

3.1 Table-Top Object Dataset

We test our system on an augmented table-top object dataset proposed in Sun et al. (2010b) which contains three common table-top object categories: computer mice, mugs, and



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

Fig. 7 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**)

	Loop Iterations	e_n (radius)	e_η (%)	e_f (%)	e^{FA}_{seg} (%)	e^{MS}_{seg} (%)
1 Plane	First Loop	0.125	25.9	13.2	2.07	51.2
	Final Loop	0.118	21.2	11.7	1.79	56.9
2 Plane	First Loop	0.133	24.9	12.1	4.00	49.0
	Final Loop	0.121	29.9	11.1	3.50	51.0

Fig. 8 Estimation errors (refer to Sect. 3.1 for definition of errors) of surface layout parameters (n, η, f), and supporting regions. Columns 3∼5 show the errors of the estimated surface normal e_n, camera height e_η, and focal length e_f. The least two columns show the two types of errors of the supporting region segmentation. All five types of errors are further reduced as the number of iterations increases (the table reports results for the 1st and 7th iteration)

staplers, where each image is associated to depth range data collected using a structure-light stereo camera. This allows us to easily estimate the ground truth 3D layout and supporting plane segmentation. The images are captured in daily office place under generic lighting conditions. Please see the last three rows in Fig. 12 for examples. We follow the training procedure described in Sun et al. (2010b) to train the DEHV detector using 200 images with their corresponding 3D information. The remaining 100 images (some with a single plane (80 images) and some with 2 planes (20 images)) are used for testing. Notice that the original dataset proposed in Sun et al. (2010b) contains 80 images. Each image from either training or testing sets contains 3∼8 object instances in random poses and locations.[3] During the testing stage, we only use 2D images and all the 3D information is inferred by our algorithm. Figure 9 shows the overall Precision Recall curve (i.e., combining three classes (computer mice, mugs, and staplers)). We use the same definition of precision-recall as in Everingham et al. (2007). That is the precision is defined as $\frac{NumOfTruePositive}{NumOfDetection}$, the recall is defined as $\frac{NumOfTruePositive}{NumOfTrueObject}$, where a detection is considered to be true if

$$a_o = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})} \geq 0.5$$

where B_p is the predicted bounding box and B_{gt} is the ground truth bounding box.

Moreover, multiple detections of the same object in an image are considered false detections. The precision recall curve is calculated by varying the threshold to select the detections.

To obtain the initial detection result (first iteration of the loop), we apply the baseline detector (Sun et al. 2010b) with no information provided by the region segmenter and the layout estimator. Figure 8 further shows the accuracy in estimating the layout parameters (n, η, f) and segmenting the supporting regions. 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 report two types of segmentation errors: e^{FA}_{seg} and e^{MS}_{seg} 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 denotes the supporting region predicted by our model, I_{SR} denotes the ground-truth supporting regions, and I_F denotes the ground truth foreground objects. We define $e^{FA}_{seg} = \frac{|I_P \bigcap I_F|}{|I_F|}$ and $e^{MS}_{seg} = \frac{|I_P \bigcap I_{SR}|}{|I_{SR}|}$, where $| \bullet |$ counts the pixel number. The smaller e^{FA}_{seg}, the lower the false alarm rate is for confusing foreground pixels as background. The higher e^{MS}_{seg}, the larger the area our algorithm can classify as supporting region.

Both Figs. 8 and 9(d) validate that the feedback loop is effective in improving (i) object detection, (ii) scene lay-

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

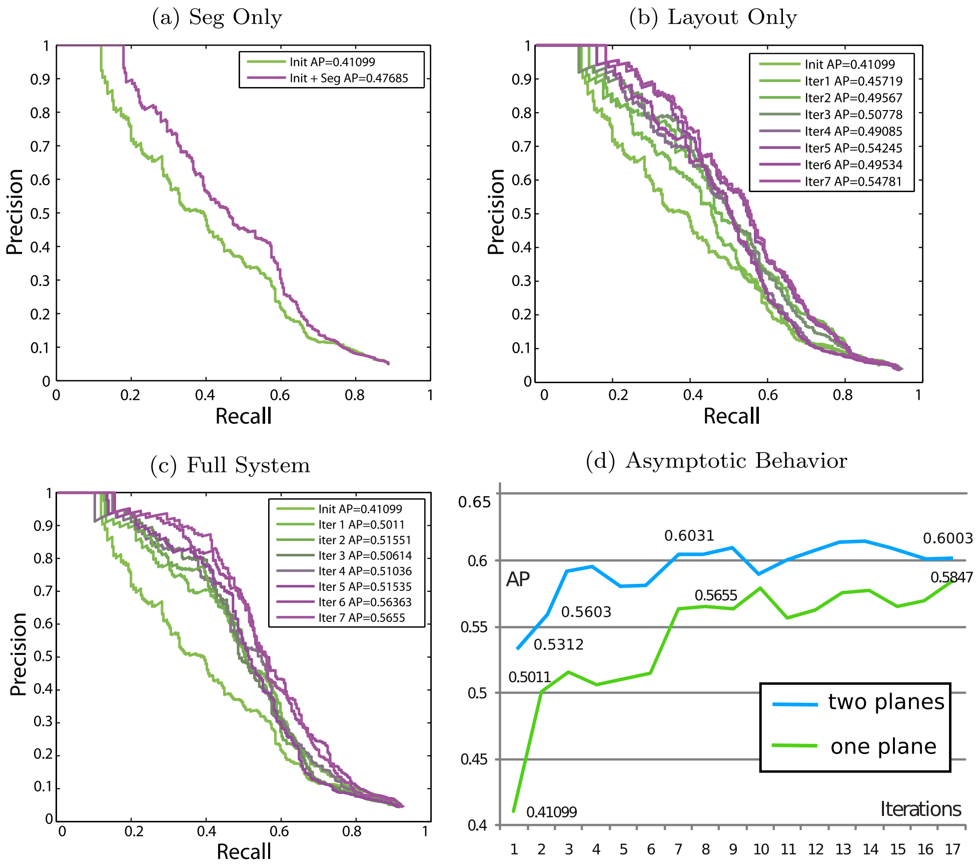


Fig. 9 Detection performance using precision-recall measurement. Panel (**a**) compares baseline detection results (Sun et al. 2010b) with our system using only supporting region segmentation. Notice that joint object detection and supporting region segmentation lead to an one-time improvement only. Panel (**b**) shows results combining detector and layout estimator for 7 iterations. Panel (**c**) shows the results when all modules (the object detector, layout estimator, region segmenter) are used in the loop for 7 iterations. Notice the results in panels (**a**, **b**, **c**) are evaluated in the testset containing one single plane. Panel (**d**) shows that the performances of our full system for the single plane and two planes cases. Notice that the system appear to asymptotically converge to a steady state on both scenarios (with a single plane and with 2 planes)

out orientation estimation, (iii) focal length estimation. The improvement of surface height η estimation is less consistent since the algorithm uses fewer objects to estimate the height of the surface (compared to the orientation of the surface). The improvement in segmenting the supporting region is also less significant. We believe that if more object categories are used, larger evidence about the appearance properties of the supporting regions can be produced, which in turn should produce more accurate segmentation results.

3.2 Label-Me Outdoor Dataset

We compare our system with another state-of-the-art method (Hoiem et al. 2006) 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 Hoiem et al. (2006).[4] The training images for our detector are extracted from Pascal 2007 cars training set (Everingham et al. 2007). Figure 10(a) compares the detection performance of our full model at different iterations with Hoiem et al. (2006). Although both methods rely on different baseline detectors, similar to Hoiem et al. (2008), our method 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 surfaces 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 object detection performance of our model with supporting region information provided by different methods (Fig. 10(b)). The detection performance of our model with supporting region information provided by our segmenter (AP = 27.6 %) is comparable to the performance (AP = 28.8 %) of our model with the ideal supporting region information. We generate the ideal supporting regions by using the ground truth bounding boxes to remove mistaken supporting regions predicted by Hoiem et al. (2006). Our proposed segmenter is also flexible in that it can easily incorporate ground plane segmentation results provided by Hoiem et al. (2006) as an additional cue. This leads to the best detection performance AP = 28.4 %. We further

[4] As explained in Bao et al. (2010) and in Sect. 2.2.2, at least 3 objects are necessary for estimating the layout.

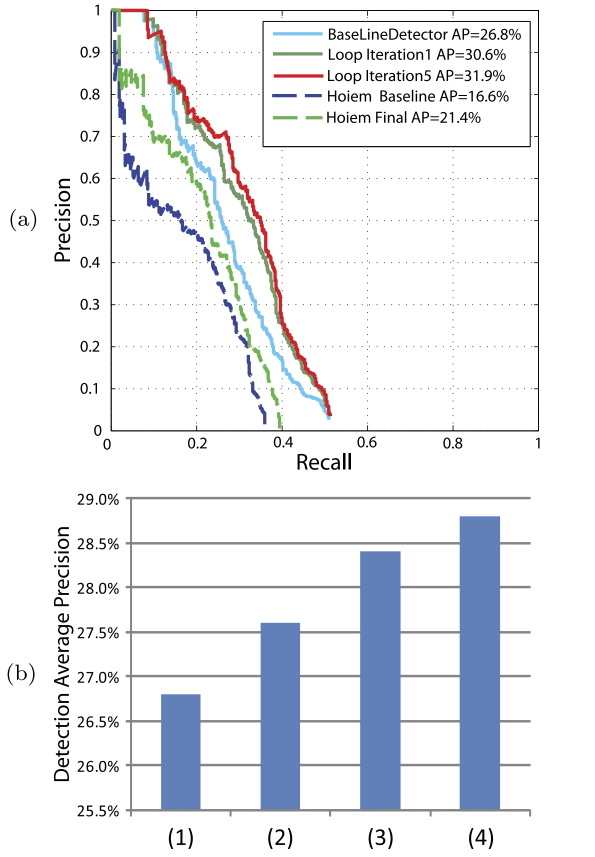


Fig. 10 Detection performance on labelme (Russell et al. 2008) dataset. Panel (**a**) shows the results after applying the full system from iteration 1 to 5. This figure also shows the results of Hoiem et al. (2006) and its baseline method (Dalal and Triggs 2005). Panel (**b**) shows average detection precision (at the final iteration) using (1) the baseline detector (Sun et al. 2010b), (2) our supporting region segmenter module, (3) supporting regions provided by Hoiem et al. (2005) as an additional cue to our region segmenter module, (4) ideal supporting regions provided by Hoiem et al. (2005) where mistaken supporting regions are removed by using ground truth object bounding boxes

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[5] is 6.6 %. And at the final (5th) iteration, the relative L_1 error is 4.2 % which is comparable to the 3.8 % error of Hoiem et al. (2006). Typical examples are shown in the first three rows of Fig. 12. All results validate that the feedback loop is effective in improving (i) object detection, (ii) scene layout orientation estimation, (iii) focal length estimation, in an outdoor environment. Moreover, our method is flexible enough to incorporate different cues such as the ground plane segmentation results provided by Hoiem et al. (2006).

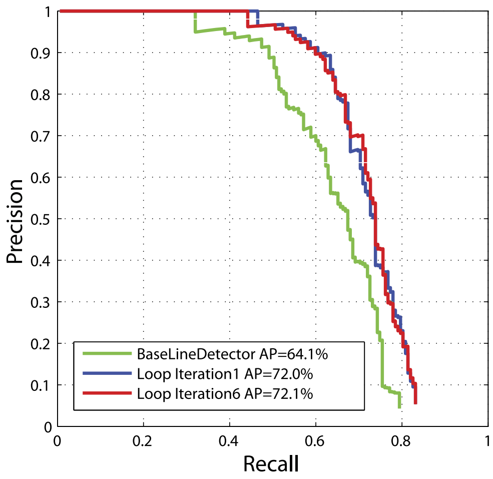


Fig. 11 Detection performance using full system form iteration 1 to 6 on the office dataset (Sudderth et al. 2008). The baseline detector is Sun et al. (2010b)

3.3 Office Dataset

We use the office dataset (Sudderth et al. 2008) 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 computer mice, monitors, and keyboards are shown in Fig. 11. Notice the improvement of almost 8 %. We validate that our method generalizes well to another indoor environment dataset. Typical examples are shown in Fig. 14.

4 Conclusion and Future Work

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 where the object detector becomes more and more accurate as evidence about the scene 3D layout and the object supporting regions becomes available and vice versa. Quantitative and qualitative experimental results on both indoor (Sun et al. 2010b; Sudderth et al. 2008) and outdoor (Hoiem et al. 2006) datasets support our claims empirically.

As future work, we would like to develop an estimation procedure which guarantees convergence and global optimality. Moreover, since, in the current implementation, the model parameters are all learned separately, we plan to develop a learning algorithm which learns all the model parameters in a joint fashion. The limitation of at least three objects per image can be overcome if a prior is placed on the focal length and the support plane parameters. We plane to explore how such prior knowledge can help improving the overall object detection, layout estimation, and supporting region segmentation accuracy.

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

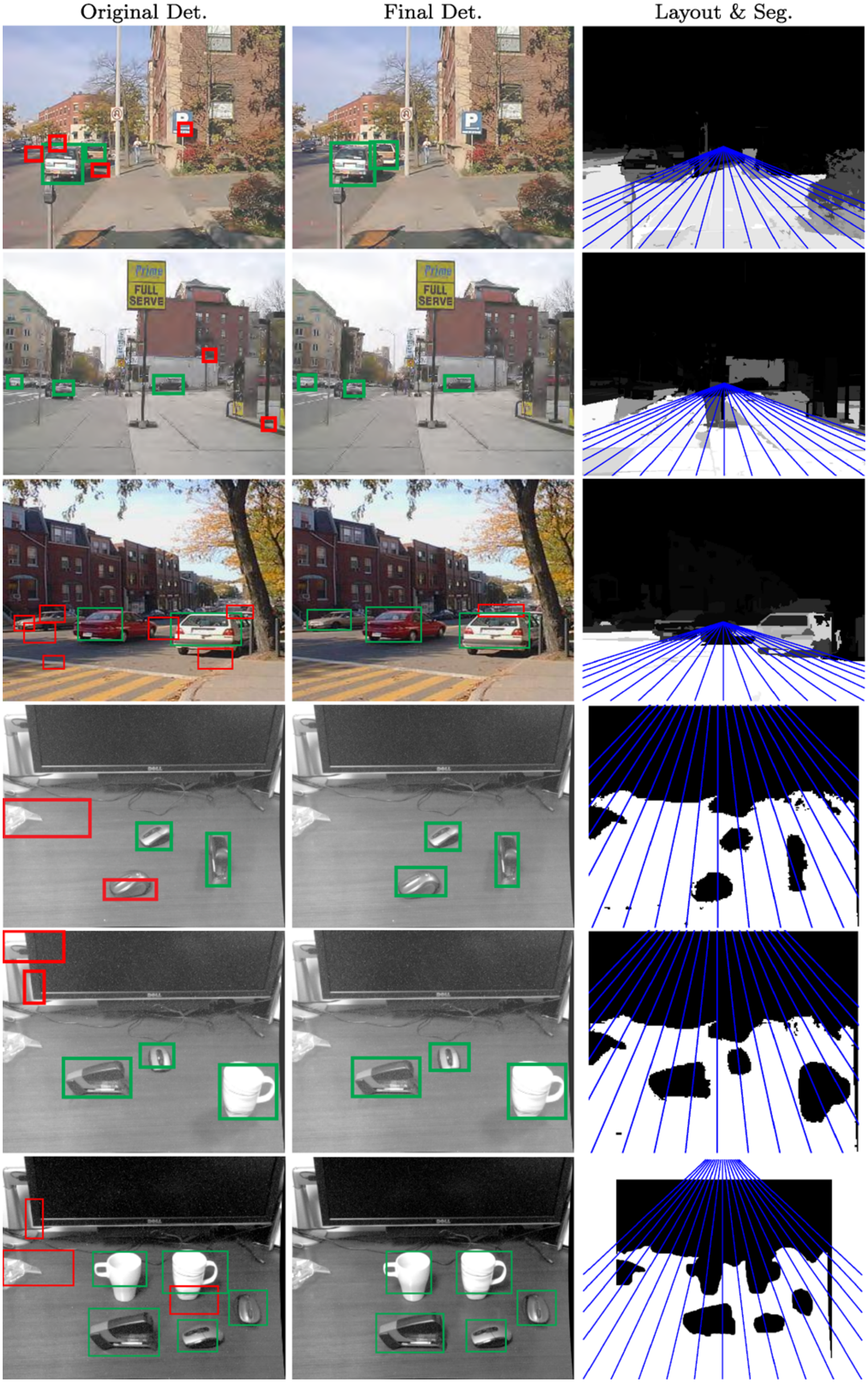


Fig. 12 Typical results of joint object detection (*green*), layout estimation (*blue*), and original false detections (*red*). The supporting region is visualized by showing the confidence that a pixel belongs to a supporting region (*white* indicate high confidence). Results on labelme (Hoiem et al. 2006), table-top (Sun et al. 2010b) datasets are shown in rows 1∼3 and 4∼6, respectively. Notice that the modules jointly improve the original detection and enable convincing layout estimation and supporting region segmentation (Color figure online)

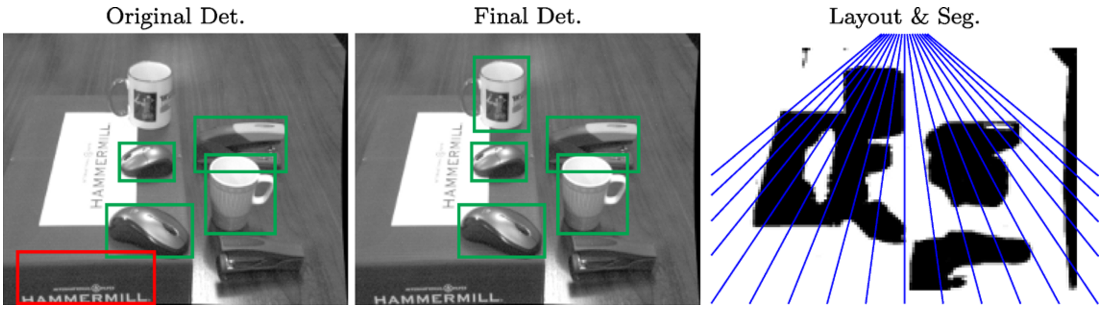


Fig. 13 Typical results of joint object detection (*green*), layout estimation (*blue*), and original false detections (*red*) on images with 2 planes in table-top dataset (Sun et al. 2010b). The supporting region is visualized by showing the confidence that a pixel belongs to a supporting region (*white* indicate high confidence). Notice that we assume multiple planes must be parallel to each other. Therefore, multiple planes will share the same vanishing line (Color figure online)

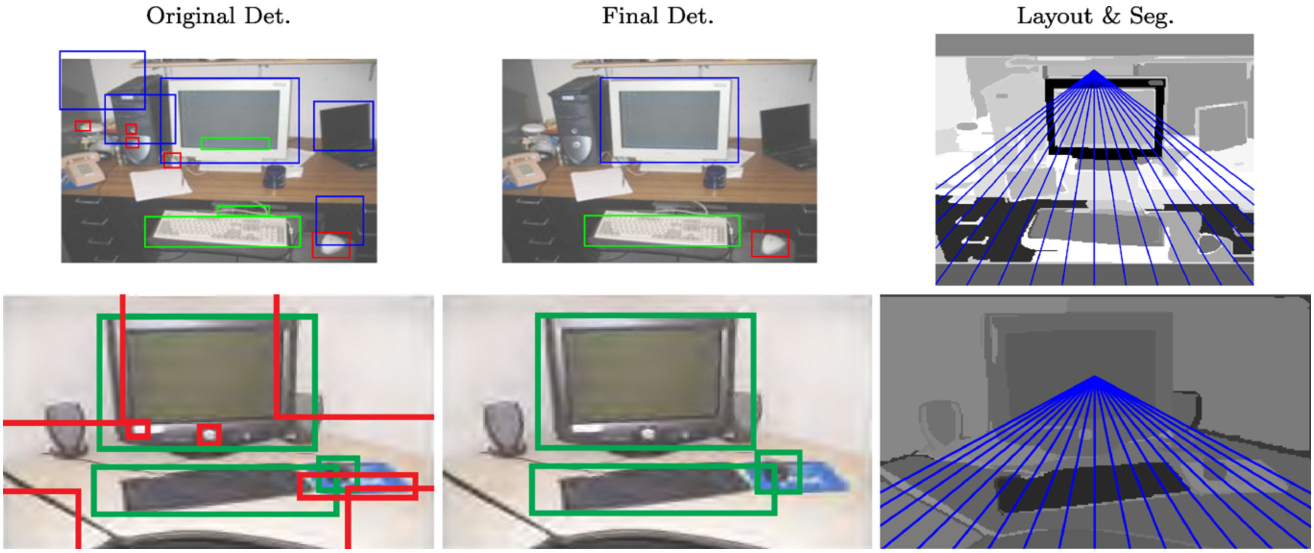


Fig. 14 Typical results of joint object detection (*green*), layout estimation (*blue*), and original false detections (*red*) on office dataset (Sudderth et al. 2008). The supporting region is visualized by showing the confidence that a pixel belongs to a supporting region (*white* indicate high confidence) (Color figure online)

Acknowledgements 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 (Sun et al. 2010b).

Appendix A: Derivation of Object Detect Score

In detail, we define $V(O,x|D)$ as the sum of individual probabilities over all observed images patches at location l_j and for all possible depth $d_j^p \in D$, i.e.,

$$V(O,x|D) = \sum_j \sum_{d_j^p \in D} p\big(O,x,C_j,d_j^p,l_j\big)$$

$$= \sum_j \sum_{d_j^p \in D} p\big(x|O,C_j,d_j^p,l_j\big) p\big(O|C_j,d_j^p,l_j\big) \times p\big(C_j|d_j^p,l_j\big) p\big(d_j^p|l_j\big)$$

where the summation over j aggregates the evidence from individual patch location, and the summation over depth d_j^p marginalizes out the uncertainty of depth corresponding to each image patch location. Since C_j is calculated deterministically from l_j and d_j^p, and assuming O only depending on C_j, we obtain:

$$V(O,x|D) \propto \sum_j \sum_{d_j^p \in D} p\big(x|O,C_j,d_j^p,l_j\big) p\big(O|C_j\big) p\big(d_j^p|l_j\big)$$

We further assign image patches with different depths to different index j. As a result, we can take only the summation over patch index j and obtain (1).

Appendix B: Proof of Three Objects Requirement

Equation (14) admits one or at most two non-trivial solutions of $\{f, n_1, n_2, n_3\}$ if at least three non-aligned observations (u_i, v_i) (i.e. non-collinear in the image) are available. If the observations are collinear, then (14) has infinite number of solutions.

$$\begin{bmatrix} u_1 & v_1 & f \\ u_2 & v_2 & f \\ u_3 & v_3 & f \\ & \vdots & \\ u_N & v_N & f \end{bmatrix} \begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix} = \begin{pmatrix} -\cos\phi_1\sqrt{u_1^2+v_1^2+f^2} \\ -\cos\phi_2\sqrt{u_2^2+v_2^2+f^2} \\ -\cos\phi_3\sqrt{u_3^2+v_3^2+f^2} \\ \vdots \\ -\cos\phi_N\sqrt{u_N^2+v_N^2+f^2} \end{pmatrix} \tag{14}$$

Proof Suppose at least three objects are not collinear in a image, then the rank of the left matrix in the left-hand side of (14) is 3. Therefore (14) provides 3 independent constraints. Recall the unknowns in (14) are n_1, n_2, n_3, f. With these constraints, each of n_1, n_2, n_3 can be expressed as a function of f, i.e. $n_i = n_i(f)$. Because $\|n\| = 1$, we obtain an equation about f:

$$\sum_{i=1,\ldots,3} n_i^2(f) = 1$$

In the above equation, f appears in the order of f^2 and f^4. Therefore, there are at most two real positive solutions of f. Given f, $\{n_1, n_2, n_3\}$ can be computed as $n_i = n_i(f)$.

On the other hand, if all objects are collinear in the image, then infinite number of solutions of (14) exist. If all objects are collinear, the rank of the left matrix in the left-hand side of (14) is 2. Without loss of generality, assume $(u_1, v_1) \neq 0$. In such a case, after using Gaussian elimination, (14) will be in the following form:

$$\begin{bmatrix} \alpha & \beta & f \\ \gamma & \epsilon & 0 \\ 0 & 0 & 0 \\ & \vdots & \end{bmatrix} \begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix} = \begin{matrix} \zeta \\ \eta \\ 0 \\ \vdots \end{matrix} \tag{15}$$

If $\widehat{f}, \widehat{n}_1, \widehat{n}_2, \widehat{n}_3$ is solution, then $\widehat{f}, \widehat{n}_1 + km_1, \widehat{n}_2 + km_2, \widehat{n}_3 + km_3$ is also a solution of (15), where (m_1, m_2, m_3) is the non-trivial solution the following equation:

$$\begin{bmatrix} \alpha & \beta & f \\ \gamma & \epsilon & 0 \end{bmatrix} \begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix} = 0$$

Hence, (14) admits infinite solutions. □

References

Bao, S. Y., Sun, M., & Savarese, S. (2010). Toward coherent object detection and scenelayout understanding. In *CVPR*.

Brostow, G. J., Shotton, J., Fauqueur, J., & Cipolla, R. (2008). Segmentation and recognition using structure from motion point clouds. In *ECCV*.

Cornelis, N., Leibe, B., Cornelis, K., & Van Gool, L. (2006). 3D city modeling using cognitive loops. In *3DPVT*.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *CVPR*.

Dance, C., Willamowski, J., Fan, L., Bray, C., & Csurka, G. (2004). Visual categorization with bags of keypoints. In *ECCV workshop on statistical learning in computer vision*.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2007). *The PASCAL visual object classes challenge 2007 (VOC2007) results*.

Fei-Fei, L., Fergus, R., & Perona, P. (2003). A Bayesian approach to unsupervised one-shot learning of object categories. In *ICCV*.

Felzenszwalb, P. F., & Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. In *IJCV*.

Felzenszwalb, P. F., & Huttenlocher, D. P. (2005). Pictorial structures for object recognition. In *IJCV*.

Fergus, R., Perona, P., & Zisserman, A. (2005). A sparse object category model for efficient learning and exhaustive recognition. In *CVPR*.

Gonfaus, J. M., Boix, X., van de Weijer, J., Bagdanov, A. D., Serrat, J., & Gonzàlez, J. (2010). Harmony potentials for joint classification and segmentation. In *CVPR*.

Gould, S., Fulton, R., & Koller, D. (2009). Decomposing a scene into geometric and semantically consistent regions. In *ICCV*.

Grauman, K., & Darrell, T. (2005). The pyramid match kernel: discriminative classification with sets of image features. In *ICCV*.

Gupta, A., & Davis, L. S. (2008). Beyond nouns: exploiting prepositions and comparative adjectives for learning visual classifiers. In *ECCV*.

Hedau, V., Hoiem, D., & Forsyth, D. (2009). Recovering the spatial layout of cluttered rooms. In *ICCV*.

Heitz, G., Gould, S., Saxena, A., & Koller, D. (2008). Cascaded classification models: combining models for holistic scene understanding. In *NIPS*.

Hoiem, D., Efros, A. A., & Hebert, M. (2005). Geometric context from a single image. In *ICCV*.

Hoiem, D., Efros, A. A., & Hebert, M. (2006). Putting objects in perspective. In *CVPR*.

Hoiem, D., Efros, A., & Hebert, M. (2007). Recovering surface layout from an image. In *IJCV*.

Hoiem, D., Efros, A. A., & Hebert, M. (2008). Closing the loop on scene interpretation. In *CVPR*.

Ladicky, L., Russell, C., Kohli, P., & Torr, P. (2010). Graph cut based inference with co-occurrence statistics. In *ECCV*.

Leibe, B., Leonardis, A., & Schiele, B. (2004). Combined object categorization and segmentation with an implicit shape model. In *ECCV workshop on statistical learning in computer vision*.

Li, C., Kowdle, A., Saxena, A., & Chen, T. (2010). Towards holistic scene understanding: feedback enabled cascaded classification models. In *NIPS*.

Li, L. J., & Fei-Fei, L. (2007). What, where and who? classifying event by scene and object recognition. In *ICCV*.

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

Liebelt, J., & Schmid, C. (2010). Multi-view object class detection with a 3D geometric model. In *CVPR*.

Payet, N., & Todorovic, S. (2011). Scene shape from textures of objects. In *CVPR*.

Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., & Belongie, S. (2007). Objects in context. In *ICCV*.

Russell, B. C., Torralba, A., Murphy, K. P., & Freeman, W. T. (2008). Labelme: A database and web-based tool for image annotation. In *IJCV*.

Savarese, S., & Fei-Fei, L. (2007). 3D generic object categorization, localization and pose estimation. In *CVPR*.

Saxena, A., Sun, M., & Ng, A. Y. (2009). Make3D: learning 3D scene structure from a single still image. In *PAMI*.

Su, H., Sun, M., Fei-Fei, L., & Savarese, S. (2009). Learning a dense multi-view representation for detection, viewpoint classification, and synthesis of object categories. In *ICCV*.

Sudderth, E. B., Torralba, A., Freeman, W. T., & Willsky, A. S. (2008). Describing visual scenes using transformed objects and parts. In *IJCV*.

Sun, M., Su, H., Savarese, S., & Fei-Fei, L. (2009). A multi-view probabilistic model for 3D object classes. In *CVPR*.

Sun, M., Bao, S. Y., & Savarese, S. (2010a). Object detection with geometrical context feedback loop. In *BMVC*.

Sun, M., Bradski, G., Xu, B. X., & Savarese, S. (2010b). Depth-encoded hough voting for coherent object detection, pose estimation, and shape recovery. In *ECCV*.

Thomas, A., Ferrari, V., Leibe, B., Tuytelaars, T., Schiele, B., & Van Gool, L. (2006). Towards multi-view object class detection. In *CVPR*.

Torralba, A., Murphy, K. P., Freeman, W. T., & Rubin, M. A. (2003). Context-based vision system for place and object recognition. In *ICCV*.

Viola, P., & Jones, M. (2002). Robust real-time object detection. In *IJCV*.