Three-Dimensional Object Detection and Layout Prediction using Clouds of Oriented Gradients

Zhile Ren and Erik B. Sudderth
Department of Computer Science, Brown University, Providence, RI 02912, USA

Abstract

We develop new representations and algorithms for three-dimensional (3D) object detection and spatial layout prediction in cluttered indoor scenes. RGB-D images are traditionally described by local geometric features of the 3D point cloud. We propose a cloud of oriented gradient (COG) descriptor that links the 2D appearance and 3D pose of object categories, and thus accurately models how perspective projection affects perceived image boundaries. We also propose a "Manhattan voxel" representation which better captures the 3D room layout geometry of common indoor environments. Effective classification rules are learned via a structured prediction framework that accounts for the intersection-over-union overlap of hypothesized 3D cuboids with human annotations, as well as orientation estimation errors. Contextual relationships among categories and layout are captured via a cascade of classifiers, leading to holistic scene hypotheses with improved accuracy. Our model is learned solely from annotated RGB-D images, without the benefit of CAD models, but nevertheless its performance substantially exceeds the state-of-the-art on the SUN RGB-D database. Avoiding CAD models allows easier learning of detectors for many object categories.

1. Introduction

The last decade has seen major advances in algorithms for the semantic understanding of 2D images [6, 29]. Images of indoor (home or office) environments, which are typically highly cluttered and have substantial occlusion, are particularly challenging for existing models. Recent advances in depth sensor technology have greatly reduced the ambiguities present in standard RGB images, enabling breakthroughs in scene layout prediction [22, 13, 41], support surface prediction [34, 8, 10], semantic parsing [11], and object detection [36]. A growing number of annotated RGB-D datasets have been constructed to train and evaluate indoor scene understanding methods [30, 21, 34, 35].

A wide range of semantic 3D scene models have been developed, including approaches based on low-level voxel representations [20]. Generalizing the bounding boxes widely used for 2D detection, the 3D size, position, and orientation of object instances can be described by bounding *cuboids* (convex polyhedra). Several methods fit cuboid models to RGB or RGB-D data [17, 16, 40] but do not have any semantic, high-level scene understanding. Other work has used CRFs to classify cuboids detected by bottom-up grouping [25], or directly detected objects in 3D by matching to known CAD models in "sliding" locations [36].

Several recent papers have used CAD models as additional information for indoor scene understanding, by learning models of object shape [39] or hallucinating alternative viewpoints for appearance-based matching [1, 24, 23]. While 3D models are a potentially powerful information source, there does not exist an abundant supply of models for all categories, and thus these methods have typically focused on a small number of categories (often, just chairs [1]). Moreover, example-based methods [36] may be computationally inefficient due to the need to match each examplar to each test image. It is unclear how many CAD models are needed to faithfully capture an object class.

To model the spatial layout of indoor scenes, many methods assume an orthogonal "Manhattan" structure [4] and aim to infer 2D projections of the 3D structure. Building on [22] and [15], Hedau et al. [12] use a structured model to rerank layout hypotheses, Schwing et al. [33] propose an efficient integral representation to efficiently explore exponentially many layout proposals, and Zhang et al. [41] incorporate depth cues. Jointly modeling objects may improve layout prediction accuracy [13, 32], but previous work has focused on restricted environments (e.g., beds that are nearly always aligned with walls) and may not generalize to more cluttered scenes. Other work has used point cloud data to directly predict 3D layout [25, 35], but can be sensitive to errors in RGB-D depth estimates.

Simple scene parsing algorithms detect each category independently, which can introduce many false positives even after non-maximum suppression. Previous work has used fairly elaborate, manually engineered heuristics to prune false detections [36] or used CAD models and layout cues jointly to model scenes [9]. In this paper we show that a *cascaded classification framework* [14] can be used to learn

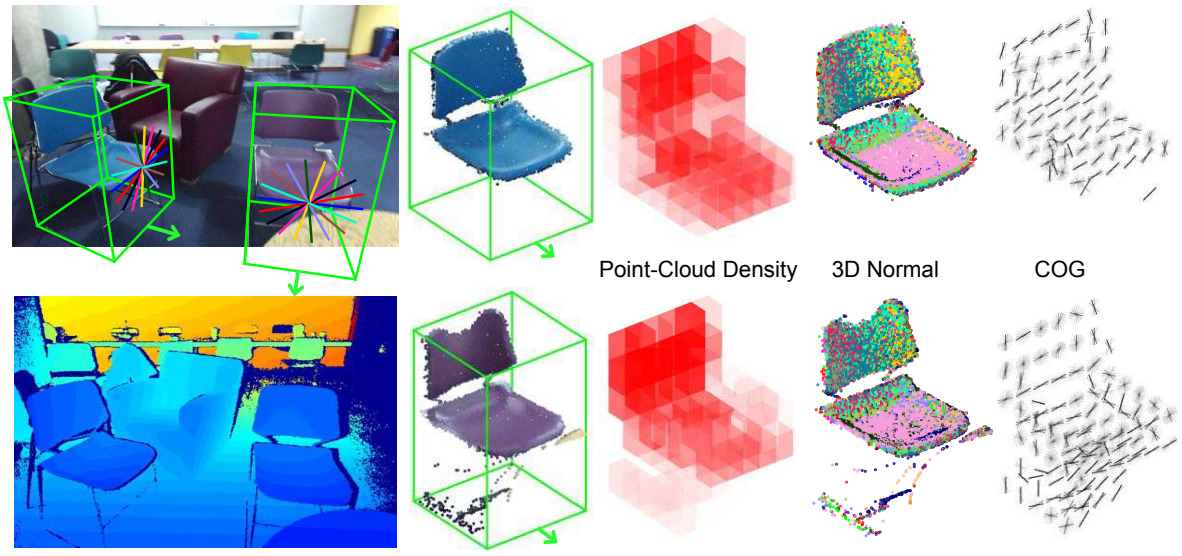


Figure 1. Given input RGB and Depth images (left), we align oriented cuboids and transform observed data into a canonical coordinate frame. For each voxel in a $6 \times 6 \times 6$ grid, we then extract (from left to right) point cloud density features, 3D normal orientation histograms, and our COG model of back-projected image gradient orientations. On the left, COG bins are colored to show alignment between instances. The value of the point cloud density feature is proportional to the voxel intensity, each 3D orientation histogram bin is assigned a distinct color, and COG feature intensities are proportional to the normalized energy in each orientation bin, similarly to HOG descriptors [5].

contextual relationships among object categories and the overall room layout, so that visually distinctive objects lead to holistic scene interpretations of higher quality.

We propose a general framework for learning detectors for multiple object categories using only RGB-D annotations. In Sec. 2, we introduce a novel *cloud of oriented gradients* (COG) feature that robustly links 3D object pose to 2D image boundaries. We also introduce a new *Manhattan voxel* representation of 3D room layout geometry. We then use a structured prediction framework (Sec. 3) to learn an algorithm that aligns 3D cuboid hypotheses to RGB-D data, and a cascaded classifier (Sec. 4) to incorporate contextual cues from other object instances and categories, as well as the overall 3D layout. In Sec. 5 we validate our approach using the large, recently introduced SUN-RGBD dataset [35], where we detect more categories with greater accuracy than a state-of-the-art CAD-model detector [36].

2. Modeling 3D Geometry & Appearance

Our object detectors are learned from 3D oriented cuboid annotations in the SUN-RGBD dataset [35], which contains 10,335 RGB-D images and 19 labeled object categories. We discretize each cuboid into a $6 \times 6 \times 6$ grid of (large) voxels, and extract features for these $6^3 = 216$ cells. Voxel dimensions are scaled to match the size of each instance. We use standard descriptors for the 3D geometry of the observed depth image, and propose a novel *cloud of oriented gradient* (COG) descriptor of RGB appearance. We also propose a *Manhattan voxel* model of 3D room layout geometry.

2.1. Object Geometry: 3D Density and Orientation

Point Cloud Density Conditioned on a 3D cuboid annotation or detection hypothesis i, suppose voxel ℓ contains $N_{i\ell}$ points. We use perspective projection to find the silhouette of each voxel in the image, and compute the area $A_{i\ell}$ of that convex region. The *point cloud density* feature for voxel ℓ then equals $\phi^a_{i\ell} = N_{i\ell}/A_{i\ell}$. Normalization gives robustness to depth variation of the object in the scene. We normalize by the local voxel area, rather than by the total number of points in the cuboid as in some related work [36], to give greater robustness to partial object occlusions.

3D Normal Orientations Various representations, such as spin images [19], have been proposed for the vectors normal to a 3D surface. As in [36], we build a 25-bin histogram of normal orientations within each voxel, and estimate the normal orientation for each 3D point via a plane fit to its 15 nearest neighbors. This feature ϕ^b_i captures the surface shape of cuboid i via patterns of local 3D orientations.

2.2. Clouds of Oriented Gradients (COG)

The *histogram of oriented gradient* (HOG) descriptor [5] forms the basis for many effective object detection methods [6]. Edges are a very natural foundation for indoor scene understanding, due to the strong occluding contours generated by common objects. However, gradient orientations are of course determined by 3D object orientation and perspective projection, so HOG descriptors that are naively extracted in 2D image coordinates generalize poorly.

To address this issue, some previous work has used 3D

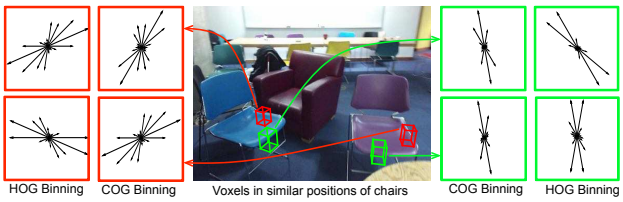


Figure 2. For two corresponding voxels (red and green) on two chairs, we illustrate the orientation histograms that would be computed by a standard HOG descriptor [5] in 2D image coordinates, and our COG descriptor in which perspective geometry is used to align descriptor bins. Even though these object instances are very similar, their 3D pose leads to wildly different HOG descriptors.

CAD models to hallucinate the edges that would be expected from various synthetic viewpoints [23, 1]. Other work has restrictively assumed that parts of objects are near-planar so that image warping may be used for alignment [7], or that all objects have a 3D pose aligned with the global "Manhattan world coordinates" of the room [13]. Some previous 3D extensions of the HOG descriptor [3, 31] assume that either a full 3D model or mesh model is given. In recent independent research [37], 3D cuboid hypotheses were used to aggregate standard 2D features from a deep convolutional neural network, but the relationship between these features and 3D object orientation was not modeled. Our *cloud of oriented gradient* (COG) feature accurately describes the 3D appearance of objects with complex 3D geometry, as captured by RGBD cameras in any orientation.

Gradient Computation We compute gradients by applying filters $[-1, 0, 1]$, $[-1, 0, 1]^T$ to the RGB channels of the unsmoothed 2D image. The maximum responses across color channels are the gradients (dx, dy) in the x and y directions, with corresponding magnitude $\sqrt{dx^2 + dy^2}$.

3D Orientation Bins The standard HOG descriptor [5] uses evenly spaced gradient bins, with 0° being the horizontal image direction. As shown in Fig. 2, this can produce very inconsistent descriptors for objects in distinct poses.

For each cuboid we construct nine 3D orientation bins that are evenly spaced from $0^\circ - 180^\circ$ in the half-disk sitting vertically along its horizontal axis. We then use perspective projection to find corresponding 2D bin boundaries. For each point that lies within a given 3D voxel, we accumulate its unsigned 2D gradient in the corresponding projected 2D orientation bin. To avoid image processing operations that can be unstable for objects with non-planar geometry, we accumulate standard gradients with warped histogram bins, rather than warping images to match fixed orientation bins.

Normalization and Aliasing We bilinearly interpolate gradient magnitudes between neighboring orientation bins [5]. To normalize the histogram $\phi^c_{i\ell}$ for voxel ℓ in cuboid i, we then set $\phi^c_{i\ell} \leftarrow \phi^c_{i\ell} / \sqrt{||\phi^c_{i\ell}||^2 + \epsilon}$ for a small $\epsilon > 0$. Accounting for all orientations and voxels, the dimension of the COG feature is $6^3 \times 9 = 1944$.

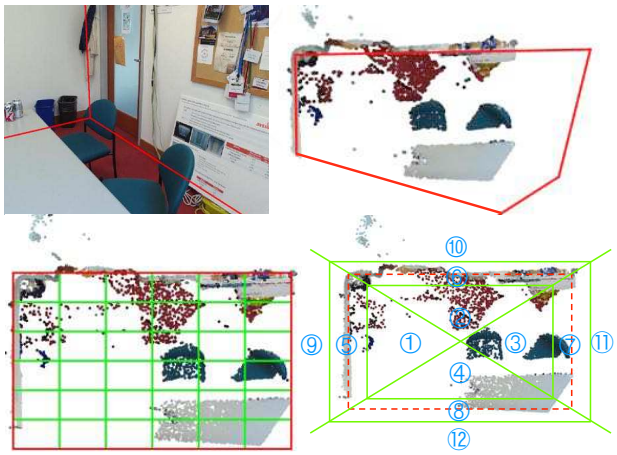


Figure 3. Models for 3D layout geometry. *Top:* Ground truth annotation. *Bottom:* Top-down view of the scene and two voxel-based quantizations. We compare a regular voxel grid (left) to our Manhattan voxels (right; dashed red line is the layout hypothesis).

2.3. Room Layout Geometry: Manhattan Voxels

Given an RGB-D image, scene parsing requires not only object detection, but also room layout (floor, ceiling, wall) prediction [12, 22, 41, 32]. Such "free space" understanding is crucial for applications like robot navigation. Many previous methods treat room layout prediction as a 2D labeling task [2, 33, 41], but small mistakes in 2D can lead to huge errors in 3D layout prediction. Simple RGB-D layout prediction methods [35] work by fitting planes to the observed point cloud data. We propose a more accurate learning-based approach to predicting Manhattan geometries.

The orthogonal walls of a standard room can be represented via a cuboid [27], and we could define geometric features via a standard voxel discretization (Fig. 3, bottom left). However, because corner voxels usually contain the intersection of two walls, they then mix 3D normal vectors with very different orientations. In addition, this discretization ignores points outside of the hypothesized cuboid, and may match subsets of a room that have wall-like structure.

We propose a novel *Manhattan voxel* (Fig. 3, bottom right) discretization for 3D layout prediction. We first discretize the vertical space between floor and ceiling into 6 equal bins. We then use a threshold of $0.15m$ to separate points near the walls from those in the interior or exterior of the hypothesized layout. Further using diagonal lines to split bins at the room corners, the overall space is discretized in $12 \times 6 = 72$ bins. For each vertical layer, regions $R_{1:4}$ model the scene interior whose point cloud distribution varies widely across images. Regions $R_{5:8}$ model points near the assumed Manhattan wall structure: R_5 and R_6 should contain orthogonal planes, while R_5 and R_7 should contain parallel planes. Regions $R_{9:12}$ capture points outside of the predicted layout, as might be produced by depth sensor errors on transparent surfaces.

3. Learning to Detect Cuboids & Layouts

For each voxel ℓ in some cuboid B_i annotated in training image I_i, we have one point cloud density feature $\phi^a_{i\ell}$, 25 surface normal histogram features $\phi^b_{i\ell}$, and 9 COG appearance features $\phi^c_{i\ell}$. Our overall feature-based representation of cuboid i is then $\phi(I_i, B_i) = \{\phi^a_{i\ell}, \phi^b_{i\ell}, \phi^c_{i\ell}\}_{\ell=1}^{216}$. Cuboids are aligned via annotated orientations as illustrated in Fig. 1, using the gravity direction provided in the SUN-RGBD dataset [35]. Similarly, for each of the Manhattan voxels ℓ in layout hypothesis M_i we compute point cloud density and surface normal features, and $\phi(I_i, M_i) = \{\phi^a_{i\ell}, \phi^b_{i\ell}\}_{\ell=1}^{72}$.

3.1. Structured Prediction of Object Cuboids

For each object category c independently, using those images which contain visible instances of that category, our goal is to learn a prediction function $h_c : I \rightarrow B$ that maps an RGB-D image I to a 3D bounding box $B = (L, \theta, S)$. Here, L is the center of the cuboid in 3D, θ is the cuboid orientation, and S is the physical size of the cuboid along the three axes determined by its orientation. We assume objects have a base upon which they are usually supported, and thus θ is a scalar rotation with respect to the ground plane.

Given n training examples of category c, we use an n-slack formulation of the structural support vector machine (SVM) objective [18] with margin rescaling constraints:

$$\min_{w_c, \xi \geq 0} \quad \frac{1}{2} w_c^T w_c + \frac{C}{n} \sum_{i=1}^{n} \xi_i \quad \text{subject to}$$
$$w_c^T [\phi(I_i, B_i) - \phi(I_i, \bar{B}_i)] \geq \Delta(B_i, \bar{B}_i) - \xi_i, \quad \text{for all } \bar{B}_i \in \mathcal{B}_i, i = 1, \ldots, n. \quad (1)$$

Here, $\phi(I_i, B_i)$ are the features for oriented cuboid hypothesis B_i given RGB-D image I_i, B_i is the ground-truth annotated bounding box, and $\mathcal{B}_i$ is the set of possible alternative bounding boxes. For training images with multiple instances, as in previous work on 2D detection [38] we add images multiple times to the training set, each time removing the subset of 3D points contained in other instances.

Given some ground truth cuboid B and estimated cuboid $\bar{B}$, we define the following loss function:

$$\Delta(B, \bar{B}) = 1 - \text{IOU}(B, \bar{B}) \cdot \left(\frac{1 + \cos(\bar{\theta} - \theta)}{2} \right). \quad (2)$$

Here, $\text{IOU}(B, \bar{B})$ is the volume of the 3D intersection of the cuboids, divided by the volume of their 3D union. The loss is bounded between 0 and 1, and is smallest when the $\text{IOU}(B, \bar{B})$ is near 1 *and* the orientation error $\theta - \bar{\theta} \approx 0$. Loss approaches 1 if either position or orientation is wrong.

We solve the loss-sensitive objective of Eq. (1) using a cutting-plane method [18]. We also experimented with detectors based on a standard binary SVM with hard negative mining, but found that the loss-sensitive S-SVM classifier is more accurate (see Fig. 5) and also more efficient in handling the large number of negative cuboid hypotheses.

Cuboid Hypotheses We precompute features for candidate cuboids in a sliding-window fashion using discretized 3D world coordinates, with 16 candidate orientations. We discretize cuboid size using empirical statistics of the training bounding boxes: $\{0.1, 0.3, 0.5, 0.7, 0.9\}$ width quantiles, $\{0.25, 0.5, 0.75\}$ depth quantiles, and $\{0.3, 0.5, 0.8\}$ height quantiles. Every combination of voxel size, and 3D location and orientation, is then evaluated.

3.2. Structured Prediction of Manhattan Layouts

We again use the S-SVM formulation of Eq. (1) to predict Manhattan layout cuboids $M = (L, \theta, S)$. The loss function $\Delta(M, \bar{M})$ is as in Eq. (2), except we use the "free-space" definition of IOU from [35], and account for the fact that orientation is only identifiable modulo 90° rotations. Because layout annotations do not necessarily have Manhattan structure, the ground truth layout is taken to be the cuboid hypotheses with largest free-space IOU.

Layout Hypotheses We predict floors and ceilings as the 0.001 and 0.999 quantiles of the 3D points along the gravity direction, and discretize orientation into 18 evenly spaced angles between 0 and 180°. We then propose layout candidates that capture at least 80% of all 3D points, and are bounded by the farthest and closest 3D points. For typical scenes, there are 5,000-20,000 layout hypotheses. See the supplemental material for more details.

4. Cascaded Learning of Spatial Context

If the detectors learned in Sec. 3 are independently applied for each category, there may be many false positives, where a "piece" of a large object is detected as a smaller object (see Fig. 4). Song et al. [36] reduce such errors via a heuristic reduction in confidence scores for small detections on large image segments. To avoid such manual engineering, which must often be tuned to each category, we propose to directly learn the relationships among detections of different categories. As room geometry is also an important cue for object detection, we integrate Manhattan layout hypotheses for *total scene understanding* [35, 25].

Typically, structured prediction of spatial relationships is accomplished via undirected *Markov random fields* (MRFs) [26]. As shown in Fig. 4, this generally leads to a *fully connected* graph [28] because there are relationships among every pair of object categories. An extremely challenging MAP estimation (or energy minimization) problem must then be solved at every training iteration, as well as for each test image, so learning and prediction is costly.

We propose to instead adapt *cascaded classification* [14] to the modeling of contextual relationships in 3D scenes. In this approach, "first-stage" detections as in Sec. 3 become input features to "second-stage" classifiers that estimate confidence in the correctness of cuboid hypotheses.

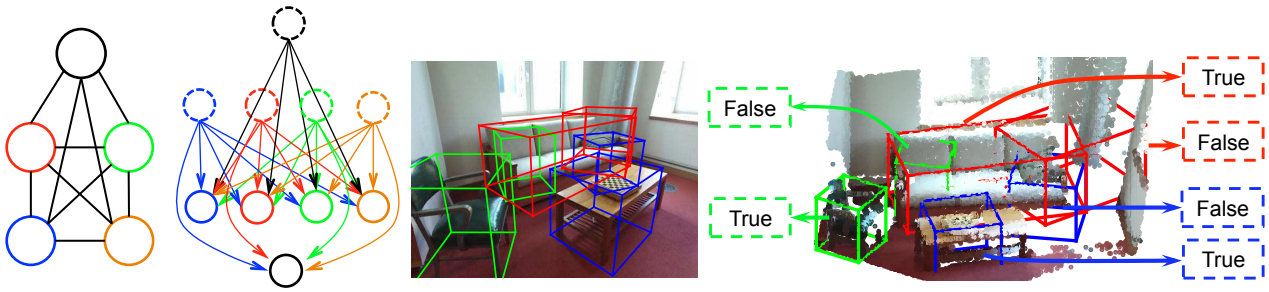


Figure 4. An illustration of how cascaded classification captures contextual relationships among objects. From left to right: (i) A traditional undirected MRF representation of contextual relationships. Colored nodes represent four object categories, and black nodes represent the room layout. (ii) A directed graphical representation of cascaded classification, where the first-stage detectors are hidden variables (dashed) that model contextual relationships among object and layout hypotheses (solid). Marginalizing the hidden nodes recovers the undirected MRF. (iii) First-stage detections independently computed for each category as in Sec. 3. (iv) Second-stage detections (Sec. 4) efficiently computed using our directed representation of context, and capturing contextual relationships between objects and the overall scene layout.

This can be interpreted as a *directed* graphical model with hidden variables. Marginalizing the first-stage variables recovers a standard, fully-connected undirected graph. Crucially however, the cascaded representation is far more efficient: training *decomposes* into independent learning problems for each node (object category), and optimal test classification is possible via a rapid *sequence* of local decisions.

Contextual Features For an overlapping pair of detected bounding boxes B_i and B_j, we denote their volumes as $V(B_i)$ and $V(B_j)$, their volume of their overlap as $O(B_i, B_j)$, and the volume of their union as $U(B_i, B_j)$. We characterize their geometric relationship via three features: $S_1(i,j) = \frac{O(B_i,B_j)}{V(B_i)}$, $S_2(i,j) = \frac{O(B_i,B_j)}{V(B_j)}$, and the IOU $S_3(i,j) = \frac{O(B_i,B_j)}{U(B_i,B_j)}$. To model object-layout context [25], we compute the distance $D(B_i, M)$ and angle $A(B_i, M)$ of cuboid B_i to the closest wall in layout M.

The first-stage detectors provide a most-probable layout hypothesis, as well as a set of detections (following non-maximum suppression) for each category. For a bounding box B_i with confidence score z_i, there may be several overlapping bounding boxes of categories $c \in \{1, \ldots, C\}$. Letting i_c be the instance of category c with maximum confidence z_{i_c}, features ψ_i for bounding box B_i are created via a quadratic function of z_i, $S_{1:3}(i, i_c)$, $A(B_i, M)$, and a radial basis expansion of $D(B_i, M)$. Relationships between second-stage layout candidates and object cuboids are modeled similarly. See the supplemental material for details.

Contextual Learning Due to the directed graphical structure of the cascade, each second-stage detector may be learned independently. The objective is simple binary classification: is the candidate detection a true positive, or a false positive? During training, each detected bounding box for each class is marked as "true" if its intersection-over-union score to a ground truth instance is greater than 0.25, and is the largest among those detections. We train a standard binary SVM with a radial basis function (RBF) kernel

$$K(B_i, B_j) = \exp\left(-\gamma ||\psi_i - \psi_j||^2\right). \quad (3)$$

The bandwidth parameter γ is chosen using validation data. While we use a RBF kernel for all reported experiments, the performance of a linear SVM is only slightly worse, and cascaded classification still provides useful performance gains for that more scalable training objective.

To train the second-stage layout predictor (the bottom node in Fig. 4), we combine the object-layout features with the Manhattan voxel features from Sec. 2.3, and again use S-SVM training to optimize the free-space IOU.

Contextual Prediction During testing, given the set of cuboids found in the first-stage sliding-window search, we apply the second-stage cascaded classifier to each cuboid B_i to get a new contextual confidence score z_i'. The overall confidence score used for precision-recall evaluation is then $z_i + z_i'$, to account for both the original belief from the geometric and COG features and the correcting power of contextual cues. The second-stage layout prediction is directly provided by the second-stage S-SVM classifier.

5. Experiments

We test our cascaded model on the SUN RGB-D dataset [35] and compare with the state-of-the-art *sliding shape* [36] cuboid detector, and the baseline layout predictor from [35]. The older NYU Depth dataset [34] is a subset of SUN RGB-D, but SUN RGB-D has improved annotations and many new images. Since unlike prior work we do not use CAD models, we easily learn and evaluate RGB-D appearance models of 10 object categories, five more than [36]. Object cuboid and 3D layout hypotheses are generated and evaluated as described in previous sections.

We evaluate detection performance via the intersection-over-union with ground-truth cuboid annotations, and consider the predicted box to be correct when the score is above 0.25. To evaluate the layout prediction performance, we calculate the free space intersection-over-union with human annotations. We provide several comparisons to demonstrate the effectiveness of our scene understanding system, and the importance of both appearance and context features.

										
Sliding-Shape [36]	42.95	19.66	20.60	28.21	60.89	-	-	-	-	-
Geom	8.29	15.06	26.20	24.53	1.15	-	-	-	-	-
Geom+COG	52.98	28.64	42.16	45.14	43.00	28.17	7.93	14.25	12.83	47.69
Geom+COG+Context-5	58.72	44.04	42.50	54.81	63.19	-	-	-	-	-
Geom+COG+Context-10	61.29	48.68	49.80	59.03	66.31	44.58	12.97	25.14	30.05	56.78
Geom+COG+Context-10+Layout	**63.67**	**51.29**	**51.02**	**62.17**	**70.07**	**45.19**	**15.47**	**27.36**	**31.80**	**58.26**

Table 1. Average precision scores for all object categories, from left to right: *bed, table, sofa, chair, toilet, desk, dresser, night-stand, bookshelf, bathtub*. Notice that using COG features without second-stage context already outperforms [36], training a second stage classifier with more contextual categories and room layout further boosts performance, and that [36] cannot model categories without CAD models.

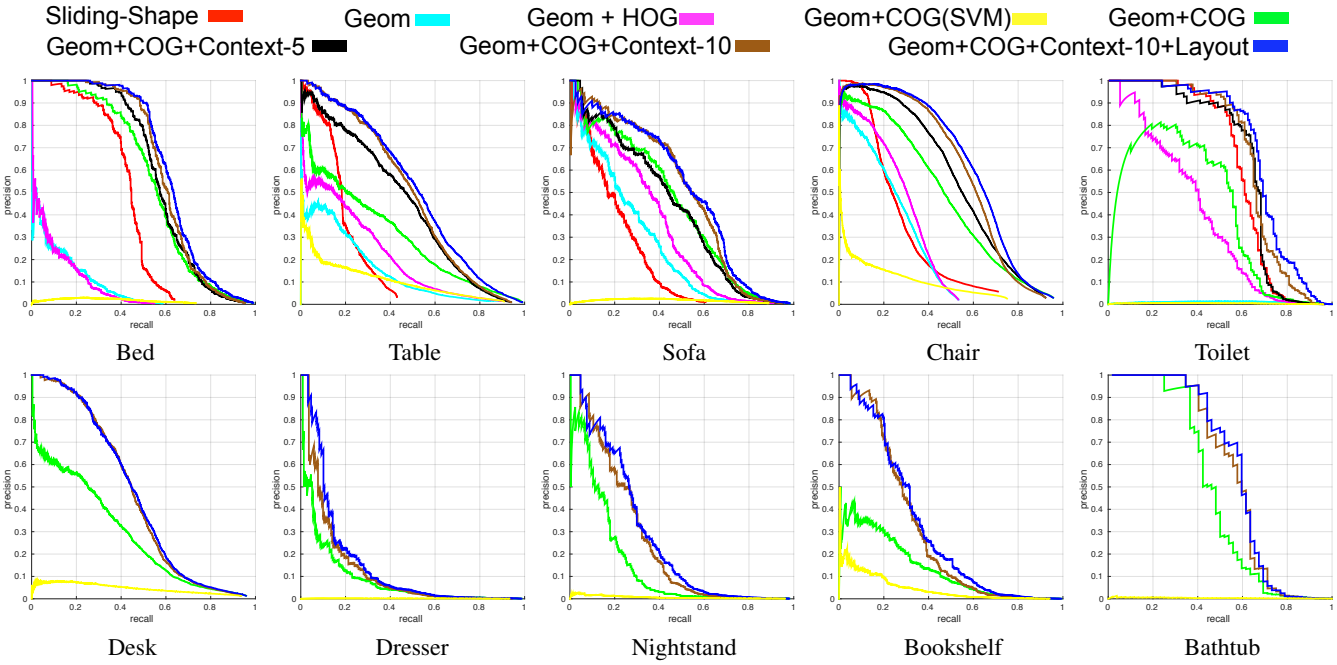


Figure 5. Precision-recall curves for 3D cuboid detection of the 5 object categories considered by [36] (top), and 5 additional categories (bottom). For the first 5 categories, we also test the importance of various features, and the gains from modeling context. See legend at top.

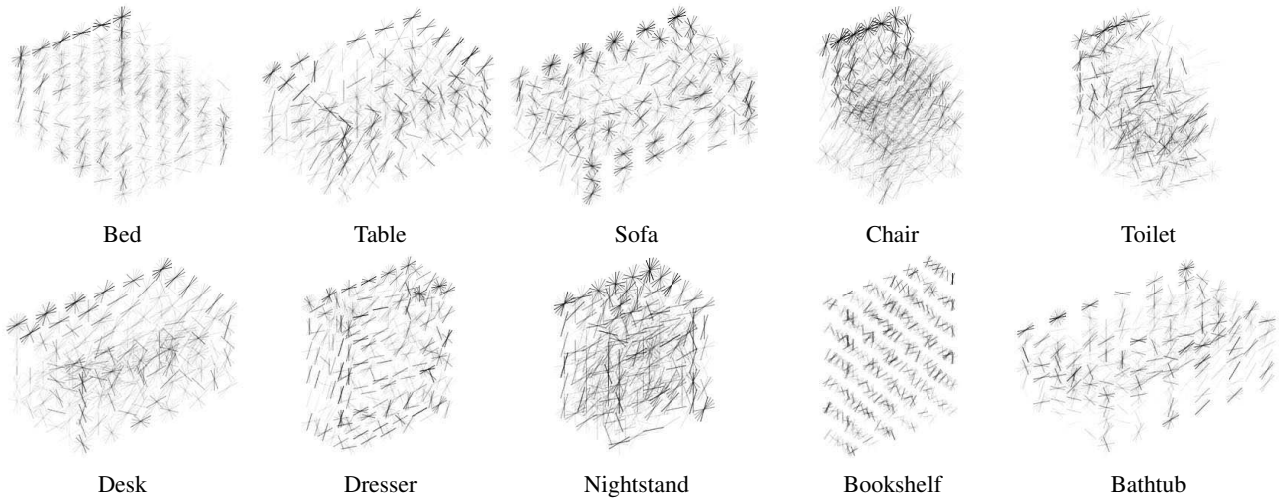


Figure 6. Visualization of the learned 3D COG features for all 10 categories. Reference orientation bins with larger weights are darker, and the 3D visualization is similar to each category's appearance. Cuboid sizes are set to the median of all training instances.

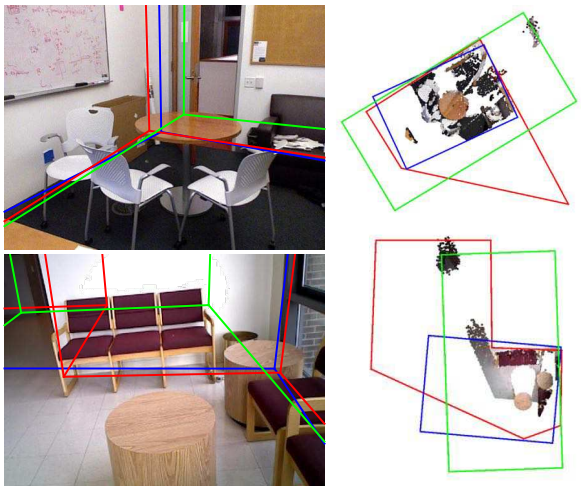


Figure 7. Comparison of our Manhattan voxel 3D layout predictions (blue) to the SUN RGB-D baseline ([35], green) and the ground truth annotations (red). Our learning-based approach is less sensitive to outliers and degrades gracefully in cases where the true scene structure violates the Manhattan world assumption.

The Importance of Appearance We trained our detector with geometric features only (Geom), and with the COG feature added (Geom+COG). There is a very clear improvement in detection accuracy for all object categories (see Table 1 and precision-recall curves in Fig. 5). Object detectors based solely on noisy point clouds are imperfect, and the RGB image contains complementary information.

HOG versus COG To demonstrate the effectiveness of the COG feature, we also use naïve 2D bins to extract HOG features for each 3D cuboid and train a detector (Geom+HOG). Since fixed 2D bins do not align with changes in 3D object pose, this feature is less informative, and detection performance is much worse than when using COG bins corrected for perspective projection.

We visualize the learned COG features for different categories in Fig. 6. We can see many descriptive appearance cues such as the oriented exterior boundaries of each object, and hollow regions for sofa, chair, toilet, and bathtub.

Cubical Voxels versus Manhattan Voxels We use the free-space IOU [35] to evaluate the performance of layout prediction algorithms. Using standard cubical voxels, our performance (**72.33**) is similar to the heuristic SUN RGB-D baseline (**73.4**, [35]). Combining Manhattan voxels with structured learning, performance increases to **78.96**, demonstrating the effectiveness of this improved discretization. Furthermore, if we also incorporate contextual cues from detected objects, the score improves to **80.23**. We provide some layout prediction examples in Fig. 7.

The Importance of Context To show that the cascaded classifier helps to prune false positives, we evaluate detections using the confidence scores from the first-stage classifier, as well as the updated confidence scores from the second-stage classifier (Geom+COG+Context-5). As shown in Table 1 and Fig. 5, adding a contextual cascade clearly boosts performance. Furthermore, when more object categories are modeled (Geom+COG+Context-10), performance increases further. This result demonstrates that even if a small number of objects are of primary interest, building models of the broader scene can be very beneficial.

We show some representative detection results in Fig. 8. In the first image our chair detector is confused and fires on part of the sofa, but with the help of contextual cues of other detected bounding boxes, these false positives are pruned away. For a fixed threshold across all object categories, we have as many true detections as the sliding-shape baseline while producing fewer false positives.

	P_g	R_g	R_r	IoU
Sliding-Shape+Plane-Fitting [35]	37.8	32.3	23.7	66.0
COG+Manhattan Voxel+Context	**47.3**	**36.8**	**35.8**	**72.0**

Table 2. Evaluation of total scene understanding [35]. We choose a threshold for object confidence scores that maximizes P_g, and compute all other metrics. Our highly accurate object and layout predictions also lead to improved overall scene interpretations.

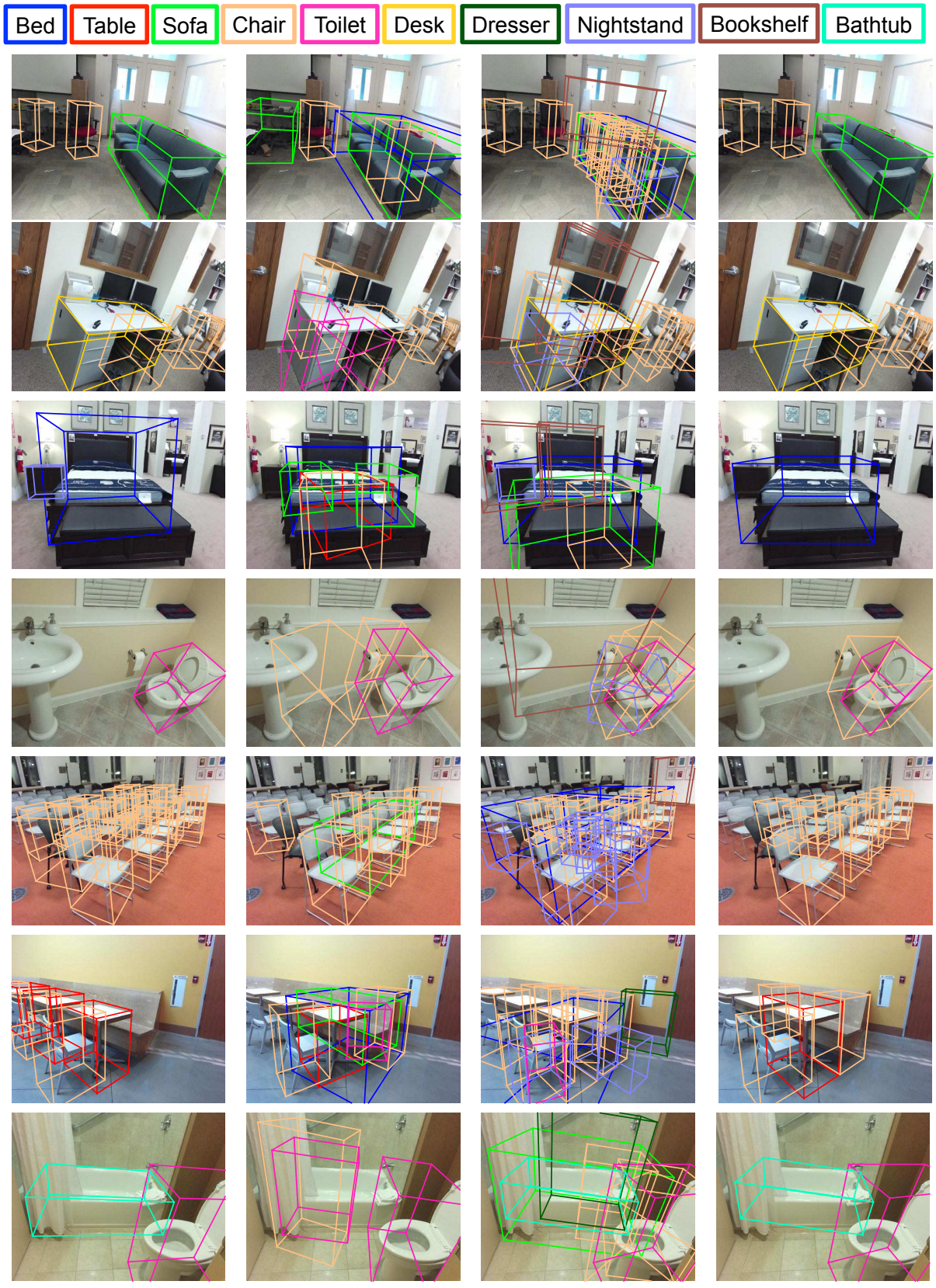
Total Scene Understanding By capturing contextual relationships between pairs of objects, and between objects and the overall 3D room layout, our cascaded classifier enables us to perform the task of total scene understanding [35]. We generate a single global scene hypothesis by applying the same threshold (tuned on validation data) to all second-stage object proposals, and choose the highest-scoring layout prediction. We report the precision, recall, and IOU evaluation metrics defined by [35] in Table 2. In every case, we show clear improvements over baselines.

Computation Speed Our algorithm, implemented in MATLAB, spends most of its running time on feature computation. For a typical indoor image, our algorithm will spend 10 to 30 minutes to compute features for one object category and Manhattan Voxel discretization, and 2 seconds to predict 3D cuboids and layout hypotheses. This speed could be dramatically improved in various ways, such as exploiting integral images for feature computation [36] or using GPU hardware for parallelization.

6. Conclusion

We propose an algorithm for 3D cuboid detection and Manhattan room layout prediction from RGB-D images. Using our novel COG descriptor of 3D appearance, we trained accurate 3D cuboid detectors for ten object categories, as well as a cascaded classifier that learns contextual cues to prune false positives. Our scene representations are learned directly from RGB-D data without external CAD models, and may be generalized to many other categories.

Acknowledgements This research supported in part by ONR Award Number N00014-13-1-0644.



Ground Truth | Sliding Shape [36] | Geom+COG | Geom+COG+Context-10

Figure 8. Detections with confidence scores larger than the same threshold for each algorithm. Notice that using contextual information helps prune away false positives and preserves true positives.

References

[1] M. Aubry, D. Maturana, A. Efros, B. Russell, and J. Sivic. Seeing 3D chairs: Exemplar part-based 2D-3D alignment using a large dataset of CAD models. In *CVPR*, 2014.

[2] J. Bai, Q. Song, O. Veksler, and X. Wu. Fast dynamic programming for labeling problems with ordering constraints. In *CVPR*, pages 1728–1735. IEEE, 2012.

[3] N. Buch, J. Orwell, and S. A. Velastin. 3D extended histogram of oriented gradients (3dhog) for classification of road users in urban scenes. In *BMVC*, 2009.

[4] J. M. Coughlan and A. L. Yuille. Manhattan world: Compass direction from a single image by Bayesian inference. In *ICCV*, volume 2, pages 941–947. IEEE, 1999.

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, volume 1, pages 886–893. IEEE, 2005.

[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results.

[7] S. Fidler, S. Dickinson, and R. Urtasun. 3D object detection and viewpoint estimation with a deformable 3D cuboid model. In *NIPS*, pages 611–619, 2012.

[8] D. F. Fouhey, A. Gupta, and M. Hebert. Unfolding an indoor origami world. In *ECCV*, pages 687–702. Springer, 2014.

[9] A. Geiger and C. Wang. Joint 3D object and layout inference from a single RGB-D image. In *German Conference on Pattern Recognition (GCPR)*, 2015.

[10] R. Guo and D. Hoiem. Support surface prediction in indoor scenes. In *ICCV*, pages 2144–2151. IEEE, 2013.

[11] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik. Learning rich features from RGB-D images for object detection and segmentation. In *ECCV*, pages 345–360. Springer, 2014.

[12] V. Hedau, D. Hoiem, and D. Forsyth. Recovering the spatial layout of cluttered rooms. In *CVPR*, pages 1849–1856. IEEE, 2009.

[13] V. Hedau, D. Hoiem, and D. Forsyth. Thinking inside the box: Using appearance models and context based on room geometry. In *ECCV*, pages 224–237. Springer, 2010.

[14] G. Heitz, S. Gould, A. Saxena, and D. Koller. Cascaded classification models: Combining models for holistic scene understanding. In *NIPS*, pages 641–648, 2009.

[15] D. Hoiem, A. Efros, M. Hebert, et al. Geometric context from a single image. In *CVPR*, volume 1, pages 654–661. IEEE, 2005.

[16] Z. Jia, A. Gallagher, A. Saxena, and T. Chen. 3D-based reasoning with blocks, support, and stability. In *CVPR*, pages 1–8. IEEE, 2013.

[17] H. Jiang and J. Xiao. A linear approach to matching cuboids in RGBD images. In *CVPR*, 2013.

[18] T. Joachims, T. Finley, and C.-N. J. Yu. Cutting-plane training of structural SVMs. *Machine Learning*, 77(1):27–59, 2009.

[19] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *PAMI*, 21(5):433–449, 1999.

[20] B.-s. Kim, P. K. Kohli, and S. Savarese. 3D scene understanding by voxel-CRF. In *ICCV*, 2013.

[21] K. Lai, L. Bo, X. Ren, and D. Fox. A large-scale hierarchical multi-view RGB-D object dataset. In *ICRA*, pages 1817–1824. IEEE, 2011.

[22] D. C. Lee, M. Hebert, and T. Kanade. Geometric reasoning for single image structure recovery. In *CVPR*, pages 2136–2143. IEEE, 2009.

[23] J. J. Lim, A. Khosla, and A. Torralba. FPM: Fine pose parts-based model with 3D CAD models. In *ECCV*, pages 478–493. Springer, 2014.

[24] J. J. Lim, H. Pirsiavash, and A. Torralba. Parsing IKEA objects: Fine pose estimation. In *ICCV*, 2013.

[25] D. Lin, S. Fidler, and R. Urtasun. Holistic scene understanding for 3D object detection with RGBD cameras. In *ICCV*, pages 1417–1424. IEEE, 2013.

[26] S. Nowozin and C. H. Lampert. Structured learning and prediction in computer vision. *Foundations and Trends in Computer Graphics and Vision*, 6(3–4):185–365, 2011.

[27] L. D. Pero, J. Guan, E. Brau, J. Schlecht, and K. Barnard. Sampling bedrooms. In *CVPR*, pages 2009–2016. IEEE, 2011.

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

[29] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015.

[30] B. C. Russell and A. Torralba. Building a database of 3D scenes from user annotations. In *CVPR*, pages 2711–2718. IEEE, 2009.

[31] M. Scherer, M. Walter, and T. Schreck. Histograms of oriented gradients for 3D object retrieval. In *Europe on Computer Graphics, Visualization and Computer Vision*, 2010.

[32] A. G. Schwing, S. Fidler, M. Pollefeys, and R. Urtasun. Box in the box: Joint 3D layout and object reasoning from single images. In *ICCV*, pages 353–360. IEEE, 2013.

[33] A. G. Schwing, T. Hazan, M. Pollefeys, and R. Urtasun. Efficient structured prediction for 3D indoor scene understanding. In *CVPR*, pages 2815–2822. IEEE, 2012.

[34] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from RGBD images. In *ECCV*, pages 746–760. Springer, 2012.

[35] S. Song, L. Samuel, and J. Xiao. SUN RGB-D: A RGB-D scene understanding benchmark suite. In *CVPR*. IEEE, 2015.

[36] S. Song and J. Xiao. Sliding shapes for 3D object detection in depth images. In *ECCV*, pages 634–651. Springer, 2014.

[37] S. Song and J. Xiao. Deep sliding shapes for amodal 3D object detection in RGB-D images. In *CVPR*, 2016.

[38] A. Vedaldi and A. Zisserman. Structured output regression for detection with partial occulsion. In *NIPS*, 2009.

[39] Z. Wu, S. Song, A. Khosla, X. Tang, and J. Xiao. 3D shapenets for 2.5D object recognition and next-best-view prediction. *arXiv preprint arXiv:1406.5670*, 2014.

[40] J. Xiao, B. C. Russell, and A. Torralba. Localizing 3d cuboids in single-view images. In *NIPS*, 2012.

[41] J. Zhang, C. Kan, A. G. Schwing, and R. Urtasun. Estimating the 3D layout of indoor scenes and its clutter from depth sensors. In *ICCV*, pages 1273–1280. IEEE, 2013.