3D Object Detection with Latent Support Surfaces

Zhile Ren
Brown University
ren@cs.brown.edu

Erik B. Sudderth
University of California, Irvine
sudderth@uci.edu

Abstract

We develop a 3D object detection algorithm that uses latent support surfaces to capture contextual relationships in indoor scenes. Existing 3D representations for RGB-D images capture the local shape and appearance of object categories, but have limited power to represent objects with different visual styles. The detection of small objects is also challenging because the search space is very large in 3D scenes. However, we observe that much of the shape variation within 3D object categories can be explained by the location of a latent support surface, and smaller objects are often supported by larger objects. Therefore, we explicitly use latent support surfaces to better represent the 3D appearance of large objects, and provide contextual cues to improve the detection of small objects. We evaluate our model with 19 object categories from the SUN RGB-D database, and demonstrate state-of-the-art performance.

1. Introduction

Object detection, typically formalized as the 2D labeling of image pixels, is one of the most widely studied semantic scene understanding problems [14, 33, 19]. Recent advances in direct depth sensing technologies have in turn enabled more accurate algorithms for 3D segmentation [29, 30, 1], reconstruction [6], synthesis [41], autonomous driving [12], and 3D object detection [40, 34].

Given an RGB-D image, the goal of 3D object detection is to recover 3D bounding boxes that capture the cubical space that objects occupy in the scene. Such representations are more powerful than 2D bounding boxes. In indoor scenes, 3D bounding boxes encode the spatial extent of objects, which can help autonomous robots better interact with their environment. In outdoor scenes, 3D bounding boxes also contain information about object orientation that is crucial for autonomous driving applications [4]. Previous work has described 3D scenes via a holistic contextual CRF model [25], or aligned CAD models to point cloud data [39, 17] in the small-scale NYU Depth dataset [37]. The larger-scale SUN RGB-D dataset [38] has enabled more recent methods that use deep neural networks to efficiently propose and categorize objects [40, 8, 23], or more accurately categorize objects via the viewpoint-invariant *cloud of oriented gradient* (COG) descriptor [34].

However, existing 3D detection algorithms suffer some common problems. Given diverse objects in the same category, modeling different visual styles is often very challenging [10], and ground truth annotations of 3D cuboids can vary among different human annotators (see Fig. 1). Moreover, objects with smaller physical size are hard to detect because the search space in the whole scene is very big, and bottom-up proposals typically contain many false positives.

State-of-the-art 3D object features, such as COG [34] and TSDF [40], are calculated for a grid of voxels within each hypothesized 3D cuboid. A major cause of feature inconsistency across different object instances is variation in the location of the supporting surface contained by many indoor objects. We treat the height of the support surface as a latent variable, and use it to distinguish different visual styles of the same object category.

Modeling support surface can also help detect smaller objects like monitors, lamps, TVs, and pillows. Since small objects are typically placed on the supporting surfaces of large objects, we first detect large objects on the ground and predict their support surface location, and then search for small objects on top of support surface areas. The reduced search space for small objects naturally reduces false positives and improves performance.

Building on the cascaded 3D scene understanding framework of Ren *et al.* [34], the contributions of this paper include the introduction of new 3D view features that improve 3D detection systems, the modeling of support surfaces as latent variables capturing intra-class variation for large objects, and the use of support surfaces to more accurately detect small objects. We evaluate our algorithm on the SUN RGB-D dataset [38] and achieve state-of-the-art accuracy in the 3D detection of 19 object categories.

2. Related Work

2D Object detection We highlight some of the most related work in the rich literature on object detection.

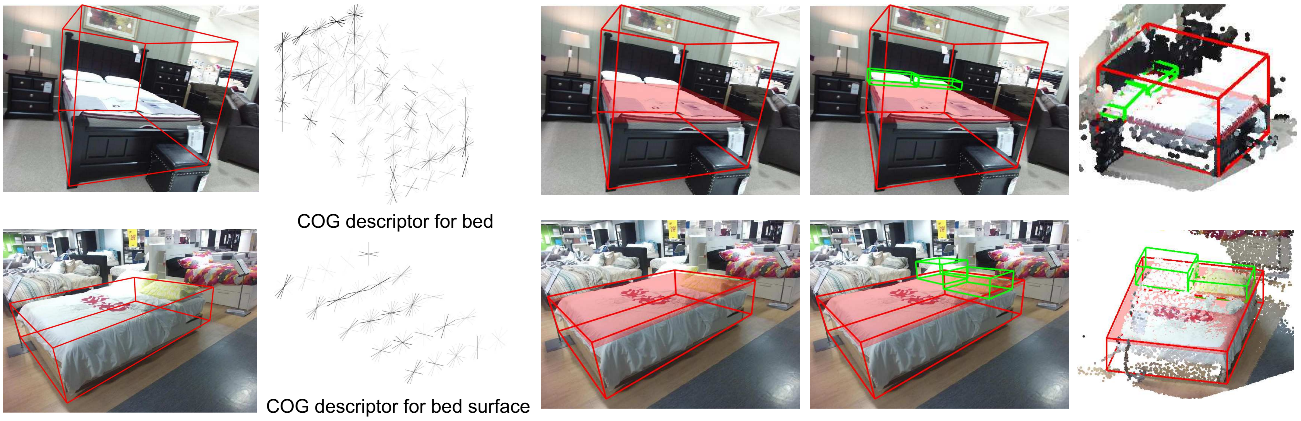


Figure 1. A visualization of 3D object detection system for beds and pillows using latent support surfaces. Given input RGB-D images, we use our learned COG descriptor [34] to localize 3D objects and infer latent support surfaces (shaded) for 3D proposals of beds (red). Then we search for pillows (green) that lie on top of the inferred support surfaces.

Dalal and Triggs [7] introduced the *histogram of oriented gradient* (HOG) descriptor to model 2D object appearance. Building on HOG, Felzenszwalb *et al.* [9] use a discriminately-trained part-based model to represent objects. This method is effective because it explicitly models object parts as latent variables, which implicitly encode object style variations. More recently, many papers have used convolutional neural networks (CNNs) to extract rich features from images [14, 13, 33, 19, 26]. These methods achieve state-of-the-art performance and efficient detection speed [31, 32], but in cluttered indoor scenes, accurate 2D object localization remains a challenging task.

3D Object Detection Increasingly, real-world computer vision systems often incorporate depth data as additional input to increase accuracy and robustness. There have recently been significant advances in methods for 3D object classification [45, 42], point cloud segmentation [29, 30], room layout prediction [24, 35], 3D object context [36, 49], and 3D shape reconstruction [43, 6]. Here, we focus on the related problem of 3D object detection.

In outdoor scenes, object localization with 3D cuboids has become a new standard in the popular KITTI autonomous driving benchmark [12]. 3D detection systems model car shape and occlusion patterns [4, 28, 46] using lidar or stereo inputs, and may also incorporate additional bird's eye view data [5]. However, methods for outdoor 3D detection are usually focused on the identification of vehicles and pedestrians in open scenes, and do not generalize to more challenging detection tasks in cluttered scenes.

In indoor scenes, a larger number of object categories is common, and categories have greater shape and style variations. Because indoor objects are often heavily occluded by their cluttered environments, localizing objects with 3D cuboids [25, 16] instead of 2D bounding boxes can be more useful. Some work aligns 3D CAD models to objects in RGB-D inputs [17, 39], as evaluated on the small-scale NYU Depth dataset [37], but the computational cost is usually expensive. A simple 3D convolutional neural network was designed to detect simple objects in real time [27]. Other work utilizes pretrained 2D detectors or region proposals as priors, and localizes 3D bounding boxes via a separate CNN [40, 8, 23]. Those methods can achieve good performance with great computational speed, but are very sensitive to the accuracy of 2D object proposals. Ren *et al.* [34] introduce the *clouds of oriented gradient* (COG) to represent 3D cuboids and perform holistic scene understanding with a cascaded prediction framework [20]. Although this work achieves state-of-the-art performance on the SUN RGB-D dataset [38] for 10 large object categories, it cannot be directly used to detect smaller objects because it requires exhaustive search in 3D space. In addition, the COG feature does not capture object style variations.

Support Surface Prediction Detecting support surfaces is an essential first step in understanding the geometry of 3D scenes for such tasks as surface normal estimation [44, 11] and shape retrieval [2]. Silberman *et al.* [37] use semantic segmentation to model object support relationships; this work was later extended by Guo *et al.* [15] for support surface prediction. However, support surfaces have not been previously used to enable 3D object detection. In this work, we treat support surfaces as latent variables to capture object style variations, and use them to localize small objects. We demonstrate the effectiveness of our 3D object detection framework on the SUN RGB-D dataset [38].

3. 3D Detection using Clouds of Gradients

Feature extraction is one of the most important steps for object detection algorithms. 2D object detectors typically use either hand-crafted features based on image gradients [7, 9] or learned features from deep neural networks [14, 13, 33, 19, 26]. For 3D object detection systems with additional depth inputs, Gupta *et al*. [18] use horizontal disparity, height above ground, and the angle of pixels local surface normal to encode images as a three channel (HHA) map for input to a convolutional neural network. While such convolutional processing of 2D images may be used to extract features from 2D bounding boxes, it does not directly provide a method for recovering 3D bounding boxes.

Song *et al*. [39] use 3D truncated signed distance function (TSDF) features to encode 3D cuboids, and their subsequent deep sliding shape [40] method aggregates TSDF with standard 2D features from a deep convolutional neural network. However, those features do not explicitly capture 3D orientation. We instead build our 3D detection algorithm on the *cloud of oriented gradients* (COG) descriptor [34]. We briefly review this approach in this section, and introduce simple extensions that improve its performance.

Clouds of Oriented Gradients Given cuboid proposals for multiple instances of some object category, as observed in RGB-D images, they are first transformed into a canonical coordinate frame. Point cloud densities and 3D normal histograms are used to model the geometric features for each voxel in a $5 \times 5 \times 5$ grid. For object appearance features, image gradients are binned in histograms according to perspective geometry in each object proposal. This novel COG feature is a 3D variant of the HOG descriptor [7]. Because image gradients are binned with respect to individual bounding box proposals, this feature is an orientation-invarient representation for 3D objects and can also be used for room layout prediction [34]. The detector of each object category is trained discriminatively using structural SVM [22] with a loss function that penalizes location and orientation errors. When analyzing 3D test scenes, we propose several cuboid sizes using empirical statistics of the training set, and use 3D sliding windows to evaluate the evidence for objects at various 3D poses.

3D objects with locally similar geometric shapes usually confuse object detectors that are run for each object category independently, resulting in many false positive detections. While simple heuristics [39] cannot fully resolve this problem, Ren *et al*. [34] propose to use a cascaded prediction framework [20] to learn the contextual relationship among objects. For overlapping pairs of detected bounding boxes, 3D overlap features and detection score differences are used to train a binary SVM to indicate whether bounding boxes are true or false positives. This second-stage detection score is added to the original detection score, resulting in a holistic scene understanding output [38].

Ren *et al*. [34] augment the COG feature for each cuboid with simple geometric histograms that have limited discriminative power. Here we introduce two novel 3D cuboid features that are suitable for 3D detection systems.

View-to-Camera Feature For single view RGB-D inputs, an object like nightstand may only expose one planer surface to the camera. At test time, features of a 3D cuboid proposal whose orientation is facing backwards resembles those of a correct detection (Fig. 2). This is because voxel features are computed by first rotating the cuboid to a canonical coordinate frame. However, due to the self-occlusions that occur in real objects, the features modeled by the COG descriptor would in fact not be visible when objects are facing away from the camera. Therefore, we add features to represent objects' view to camera, and learn to explicitly distinguish implausible object orientations.

Specifically, we compute the cosine x of the angle between the cuboid orientation and its viewing angle from camera in horizontal direction. Then we define a set of radial basis functions of the form

$$f_j(x) = \exp\left(- \frac{(x - \mu_j)^2}{2\sigma^2} \right),$$

and space the basis function centers μ_j evenly between

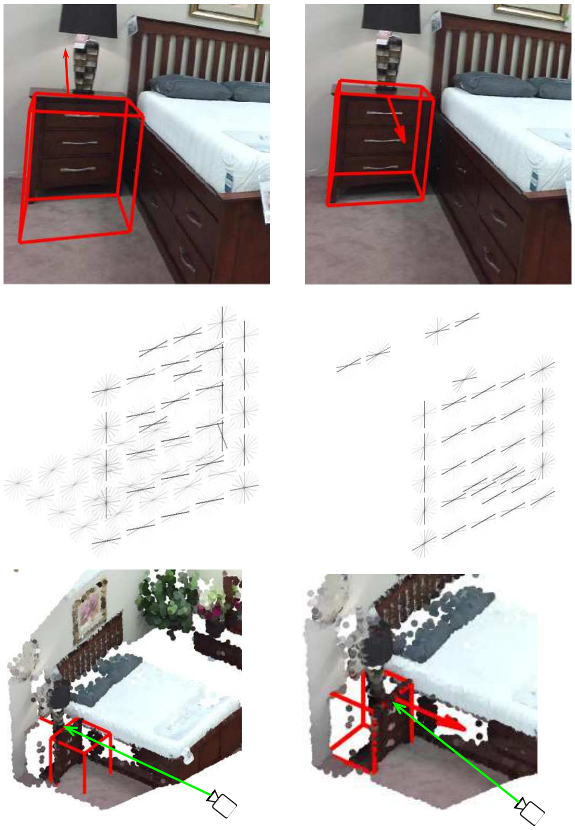


Figure 2. A false positive 3D detection for nightstand without using view-to-camera feature (left). The COG feature is similar to that of a correct detection (right) but the orientation is flipped.

$[-1, 1]$ with step size 0.2. The bandwidth $\sigma = 0.5$ was chosen using validation data. Radial basis expansions are a standard approach to non-linear regression, and can also be seen as a layer of a neural network. We expand the camera angle using this basis representation and refer to the resulting 11-dimensional vector as the *view-to-camera feature*.

Scene Layout Feature The interaction between objects and the scene layout (floor, walls, ceiling) provides important cues for object detection. For example, Song *et al.* [40] propose 3D objects along the predicted walls of the scene. Ren *et al.* [34] introduce a Manhattan voxel discretization to better predict scene layouts and model object-layout interactions. We model object-layout interactions by first computing the distance and angle to the nearest predicted wall using Manhattan voxels [34], then expand the distance-to-wall value using radial basis functions spaced between $[0, 5]$ with step size 0.5. We also expand the absolute cosine value of the angle-to-wall using radial basis functions spaced between $[0, 1]$ with step size 0.2 and $\sigma = 0.5$. Combining these layout features with the view-to-camera feature, we are able to improve detection performance for most object categories (see Table 1).

4. Modeling Latent Support Surfaces

Geometric descriptors and COG descriptors [34] are able to capture local shapes and appearances. However, 3D objects in indoor scenes have widely varying visual styles. Moreover, 3D cuboid annotations are labeled by different people from Mechanical Turk in SUN RGB-D dataset [38], thus objects in the same category may have inconsistent 3D annotations. As a result, features are inevitably noisy and inconsistent across different object instances (see Fig. 3).

To explicitly model different visual styles for each objects, a classical approach is to use part-based models [9, 10] where objects are explained by spatially arranged parts. However, indoor objects have very diverse visual styles, and it is very challenging to design a consistently varying set of latent parts. However, for many object categories, the height of the support surface is the primary cause of style variations (Fig. 3). Therefore, we explicitly model the support surface as a latent part for each object.

By modeling support surfaces we can also constrain the search space for small object detectors. Such detectors are otherwise intractable to learn and perform poorly due to the large set of possible 3D poses [34].

4.1. Latent Structural SVM Learning

Some previous work was specifically designed to predict the area of support surface regions [15], but the predicted support surfaces are not semantically meaningful. Inspired by deformable part-based models for 2D object detection [9], we propose to treat the relative height of the support surface of each object as a latent variable and use latent structural SVMs [47] to learn the detector.

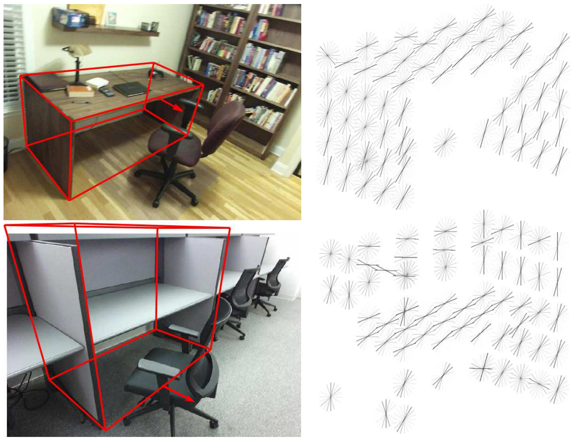


Figure 3. Different surface heights for "desk" in SUN RGB-D dataset [38] lead to inconsistent 3D COG representations [34].

We follow the notation of Ren *et al.* [34] with an updated learning objective. For each category c, our goal is to learn a prediction function $I \to (B, h)$ that maps an RGB-D image I to a 3D bounding box $B = (L, \theta, S, y)$ along with its relative surface height h. L is the center of the cuboid in 3D, θ is the cuboid orientation, S is the physical size of the cuboid along the three axes determined by its orientation, and y is an indicator variable representing the existence of such prediction. The latent variable h is defined as the relative surface height to the bottom of the cuboid. We discretize cuboid height to 7 slices, and thus h localizes the support surface to one of those slices (see Fig. 4).

Given n training examples of category c, we want to solve the following optimization problem:

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

Here B_i is the ground-truth bounding box, $\mathcal{B}_i$ is the set of possible bounding boxes, and $\mathcal{H}$ is the set of possible surface heights. $\phi(I, B, h)$ are the features associated to cuboid B whose relative surface height is indicated by h. We first discretize B into $5 \times 5 \times 5$ voxels and compute geometric features, COG [34], view-to-camera feature, and scene layout feature, as denoted by $\phi_{\text{cuboid}}(I, B)$. Then we discretize B with finer resolutions at the vertical dimension into $5 \times 5 \times 7$ voxels and take the h-th slice from the bottom to represent cuboid feature, as denoted by $\phi_{\text{surface}}(I, B, h)$. Finally we add an indicator vector for support surface height, so that

$$\phi(I, B, h) = [\phi_{\text{cuboid}}(I, B), \phi_{\text{surface}}(I, B, h), 0, ..., 1, ..., 0].$$

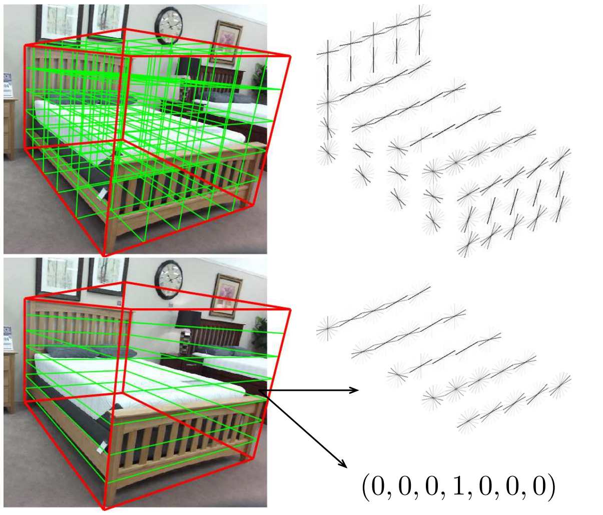


Figure 4. Features of 3D cuboid with support surface. The surface feature is computed at a single slice of the cuboid followed by an indicator vector to represent the relative height.

If the indicator variable y in B is 0, meaning there's no detection, we set the feature vector to be all zeros. A visualization of support surface feature is shown in Fig. 4.

Following Ren *et al*. [34] we define a loss function for cuboid proposals $\bar{B}$: If a scene contain ground truth cuboid B and indicator variable $\bar{y}$ is 1, we compute

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

where $\mathrm{IOU}(B,\bar{B})$ is 3D intersection over union. The scale of this loss function ranges in $[0,1]$. If a scene doesn't contain any ground truth cuboid and the indicator variable $\bar{y}$ is 0 for the cuboid proposal, the loss is set to be 0. We penalize all other cases with a loss of 1.

To train the model with latent support surfaces, we follow Ren *et al*. [34] by pre-training cuboid descriptors (geometric features, COG, view-to-camera, and scene layout feature) without modeling support surface. We then extract the center slice of pre-trained cuboid descriptors and concatenate it to the pre-trained models. Finally, we initialize the support surface height indicator vector randomly in $[0,1]$. We use the CCCP algorithm [48] to solve the resulting latent structural SVM learning problem [47].

4.2. Small Object Detection on Support Surfaces

In indoor scenes, besides large furniture like beds and chairs, many other objects with comparatively small physical size are very hard to detect [40, 34]. Some algorithms are specifically designed to detect small objects in 2D images using multi-scale methods [3, 21], but they cannot be directly applied to 3D object detection.

The biggest issue for detecting small objects is that the search space can be enormous, and thus training and testing with a sliding-windows based approach are usually intractable. But note that small objects, such as pillows and monitors and lamps, are usually placed on top of other objects with support surfaces. If we only search for small objects on predicted support surfaces, the search space will be greatly reduced. As a result, the inference speed will be improved and object proposals contain less false positives. This is another benefit of modeling support surfaces.

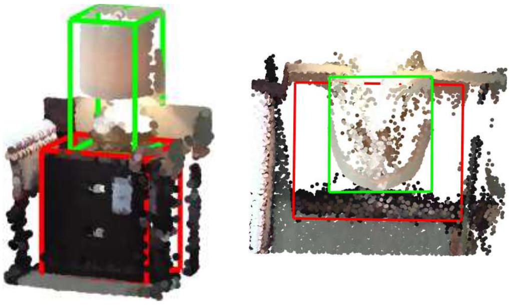


Figure 5. To model contextual relationships between small objects and the large objects supporting them, we compute the 2D overlap between 3D bounding boxes from the top-down view.

In our implementation, we first detect large objects of indoor scenes that are on the ground [34], then we search for smaller objects only on top of the support surfaces of those large objects with positive confidence scores. We reduce the voxel descritization size to be $3\times3\times3$ for lamps and pillows because small cuboids contain less pixels, and $3\times1\times3$ for monitors and TVs because they have thin shapes.

4.3. Spatial Contextual Learning for All Objects

Our object detector is trained discriminatively for each object category. At test time, 3D objects with locally similar shapes can confuse 3D detectors trained for each object category independently. Instead of designing simple heuristics to handle false positives, we follow Ren *et al*. [34] by using an effective cascaded detection framework [20] to model contextual relationships among cuboid proposals.

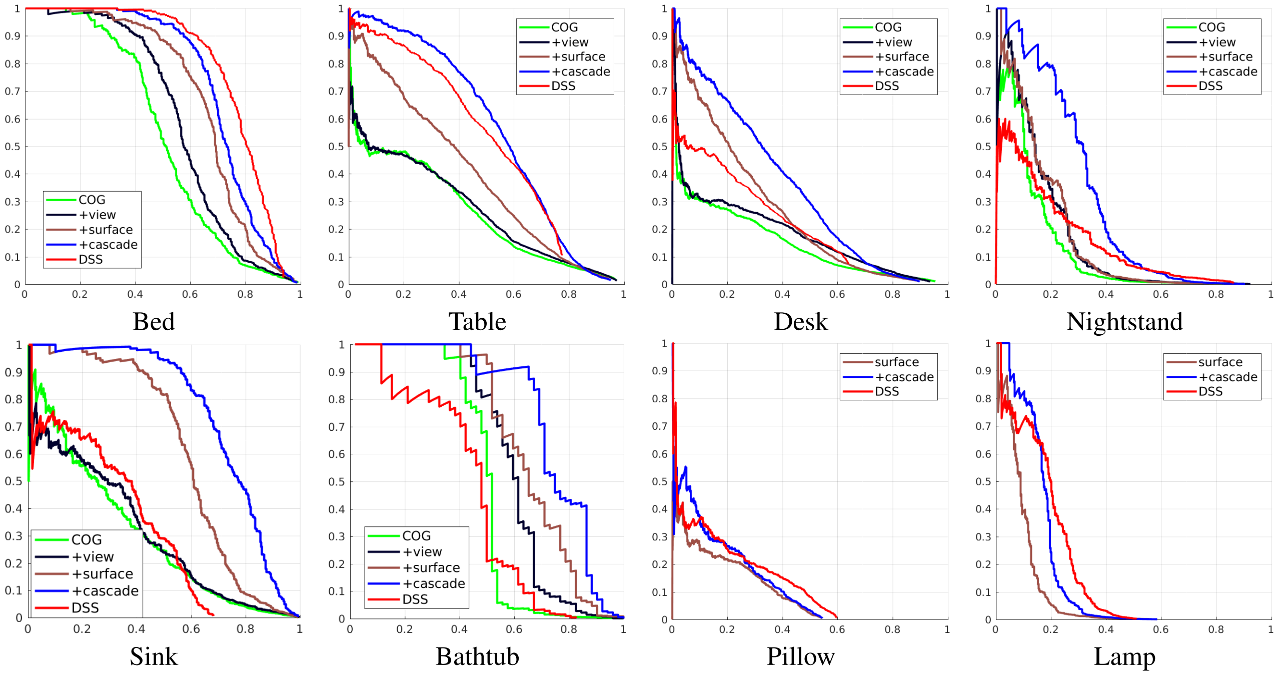
For each 3D cuboid proposal, we encode its contextual relationship with the highest confidence cuboid proposals in all object categories using 3D overlapping features and confidence differences. Using those contextual features we learn a linear SVM to determine whether those 3D object proposals are correct or not, and add this updated confidence score to first-stage detection scores. We refer readers to the supplementary material of Ren *et al*. [34] for a detailed explaination. For small objects that are placed on the support surfaces of large objects, 3D overlap features are noisy. We replace 3D overlap with 2D overlap scores from the top-down view of the scene (Fig. 5). With updated confidence scores that account for both original beliefs and contextual cues, object proposals contain fewer false positives and object detectors have improved performance.

	Bathtub	Bed	Bookshelf	Chair	Desk	Dresser	Nightstand	Sofa	Table	Toilet	Box	Door	Counter	Garbage-bin	Sink	Pillow	Monitor	TV	Lamp	mAP (10)	mAP (19)
COG	49.8	53.0	8.5	39.0	14.9	5.5	12.8	52.8	26.0	34.5	11.6	0.9	2.4	20.1	30.3	0.0	0.0	0.0	0.0	29.68	19.1
+view	60.2	59.3	18.4	40.4	18.0	9.5	16.8	53.2	26.8	41.6	11.0	3.2	4.6	21.8	30.6	0.0	0.0	0.0	0.0	34.4	21.9
+surface	66.6	68.0	21.5	42.0	26.0	8.5	17.1	52.8	39.0	45.8	11.3	2.6	4.0	19.7	60.9	9.9	1.6	0.4	9.6	38.7	26.7
+cascade	**76.2**	73.2	**32.9**	60.5	34.5	13.5	30.4	**60.4**	**55.4**	73.7	**19.5**	**5.4**	**10.7**	**34.6**	**75.3**	12.5	**1.6**	**2.1**	16.9	**51.0**	**36.3**
SS [39]	-	43.0	-	28.2	-	-	-	20.6	19.7	60.9	-	-	-	-	-	-	-	-	-	-	-
DSS [40]	44.2	**78.8**	11.9	61.2	20.5	6.4	15.4	53.5	50.3	78.9	1.5	0.0	4.1	20.4	32.3	**13.3**	0.2	0.5	**18.4**	42.1	26.9
Ren [34]	58.3	63.7	31.8	**62.2**	**45.2**	15.5	27.4	51.0	51.3	70.1	-	-	-	-	-	-	-	-	-	47.6	-
Lahoud [23]	43.5	64.5	31.4	48.3	27.9	**25.92**	**41.9**	40.39	37.0	**80.4**	-	-	-	-	-	-	-	-	-	45.1	-

Table 1. Experiment results on SUN RGB-D dataset [38]. Our baseline method uses COG descriptor. Adding extra features to model view-to-camera and scene layout (+view) improves performance, and modeling support surfaces (+surface) not only help detect large objects but also reduce many false positives for small objects (last 4 categories). The final stage cascaded detection framework [34] (+cascade) models object context and help boost the performance to the state-of-the-art over existing methods for the first 10 and all 19 object categories.

	Bathtub	Bed	Bookshelf	Chair	Desk	Dresser	Nightstand	Sofa	Table	Toilet	Box	Door	Counter	Garbage-bin	Sink	Pillow	Monitor	TV	Lamp	mAP (10)	mAP (19)
Whole System	**76.2**	**73.2**	**32.9**	**60.5**	**34.5**	**13.5**	**30.4**	**60.4**	**55.4**	**73.7**	**19.5**	**5.4**	**10.7**	34.6	**75.3**	**12.5**	**1.3**	**2.1**	**16.9**	**51.0**	**36.3**
-view-surface	53.3	63.0	18.7	61.6	29.0	7.5	20.2	58.8	49.1	62.8	17.3	1.1	6.6	**39.1**	60.3	0.0	0.0	0.0	0.0	42.4	28.9

Table 2. We compare our holistic scene understanding system with cascaded detection on SUN-RGBD dataset [38]. Although cascaded detection is powerful, there is still a drop in performance without modeling view-to-camera feature, scene layout and support surfaces.



Bed Table Desk Nightstand

Sink Bathtub Pillow Lamp

Figure 6. Precision-Recall curves for several object categories including small objects (pillow and lamp) on SUN RGB-D dataset [38].

5. Experiments

We train our 3D object detection algorithm solely on the SUN RGB-D dataset [38] with 5285 training images, and report performance on 5050 test images for all 19 object categories (Table 1). The NYU Depth dataset [37] has 3D cuboid labels for 1449 images, but annotations are noisy and inconsistent. Some previous work has only evaluated detection performance on this small dataset [17], or defined their own annotations for 3D cuboids [8]. Because the SUN RGB-D dataset contains all images from NYU Depth dataset with more accurate annotations, we do not evaluate on the NYU Depth dataset in this paper.

Baseline Algorithm using COG We implement a baseline detector using only COG features [34] and local geometric features of the point cloud. This method is denoted by "COG" in the first row of Table 1. Note that this detectors' performances is slightly different from the first stage detection scores in Ren *et al*. [34] because we are using a coarser $5 \times 5 \times 5$ discretization (versus $6 \times 6 \times 6$) for each cuboid. With reduced feature size, our algorithm is more computationally efficient but has similar accuracy.

Effectiveness of View-based Features By adding extra view-to-camera features and scene layout features, denoted by "+view" in the second row of Table 1, we witness notable improvements on detecting small objects with a layered shape such as dressers and nightstands. Those objects usually expose only one side to the camera, and the view-to-camera feature is helpful in distinguishing correct predictions. With scene layout features, we also witness improvements for objects whose orientations strongly correlate with directions of walls, such as beds and bookshelves.

Modeling Latent Support Surfaces For objects such as beds, tables, and desks, modeling support surface as a latent variable help capture the intra-class style variations within each cuboid and we witness great performance gains in the third row of Table 1. We visualize examples of inferred support surfaces in Figure 7. For objects that do not have explicit "support surfaces", such as bathtub, bookshelf, and sink, our model can be viewed as a single part-based model and is also effective for 3D object detection. Note that the goal of this work is to model latent support surface in order to help 3D detection, not to predict accurate support surface area of the scene. We do not use any annotations of support surfaces when training, and also do not evaluate our performance on surface prediction benchmarks [15].

Small Object Detection Detecting small objects is a challenging task and is still an open problem. Without modeling support surfaces, our baseline method fails to detect small objects because the search space is large and 3D object proposals contain many false positives. Using simple heuristics to check support relationships in the SUN-RGBD annotations, we find more than 95% of lamps/pillows/monitors/TVs are placed on the surface of night-stands/tables/beds/desks/dressers. Searching on the predicted surface region enables our algorithm to discover small objects with higher precision. See row 3 in Table 1.

Comparison to Other Methods By modeling latent support surface, our algorithm already outperforms the state-of-the-art method of Ren *et al*. [34] for 10 large object categories, and can also detect some smaller objects. In this paper our main goal is to demonstrate the effectiveness of modeling latent support surface, and we think the current system already shows great potential.

Comparing with other algorithms that use CNN features [40, 23] pretrained on external datasets, the performance of our algorithm is comparable even without the cascaded prediction step. Conventional CNNs for 3D detection [40, 23] are trained to produce weighted confidence scores for each of multiple object categories, while our first-stage detector algorithm is instead tuned to discriminatively localize individual categories in 3D. Our subsequent cascaded prediction [20] of contextual relationships between object detections has structural similarities to a multi-stage neural network, but it is trained using (convex) structural SVM loss functions and designed to have a more interpretable, graphical structure. Interestingly, our cascaded approach is comparable to or more accurate than standard 3D CNNs [40, 23] in the detection of both 10 and 19 object categories.

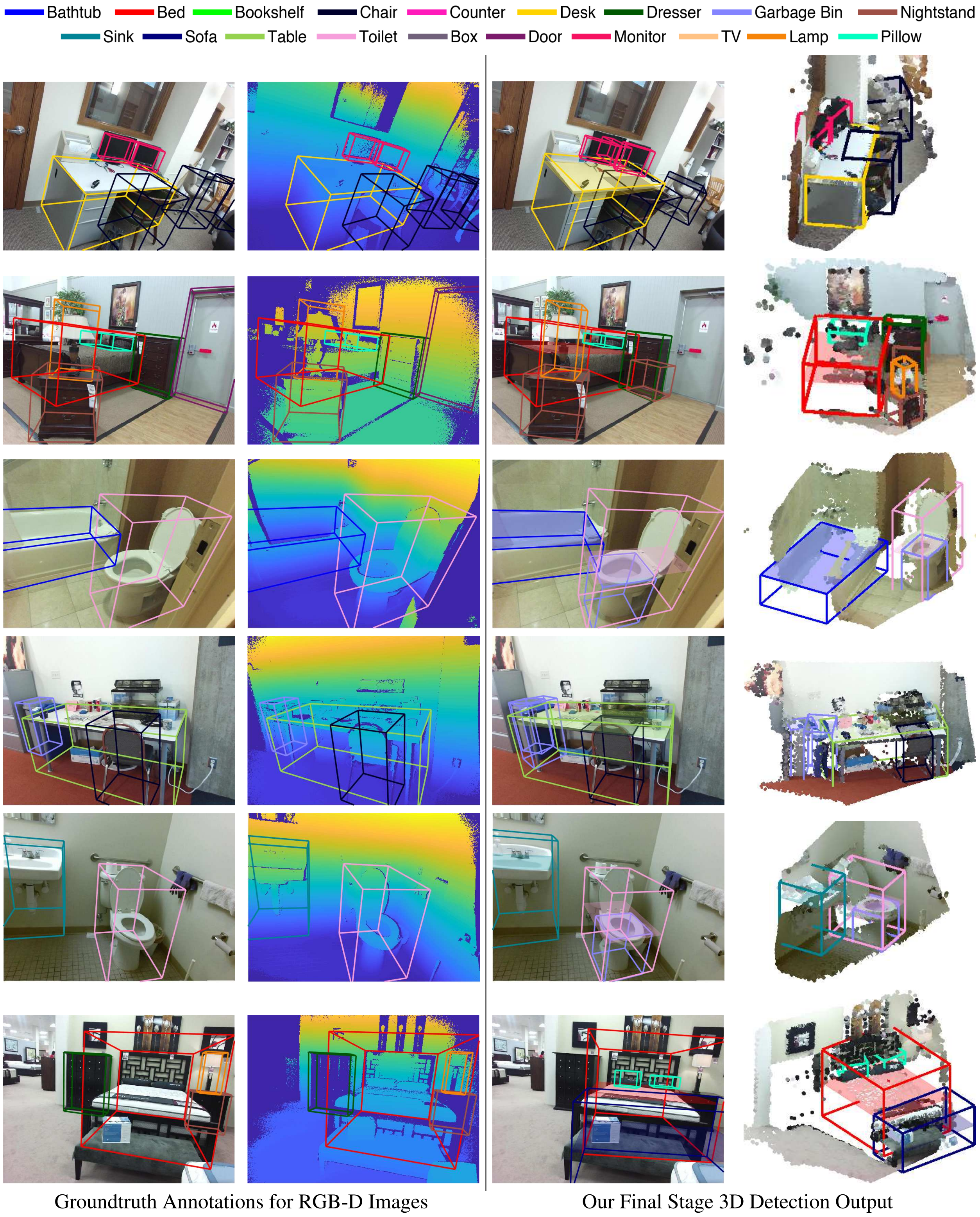
Computational Speed We implemented our algorithm using MATLAB in a 2.5GHz single core CPU. The computational speed of our algorithm is 10-30min per image, which is slightly better than the reported speed in Ren *et al*. [34]. The most time-consuming part is the feature computation step, which could be improved by using parallel computing with multi-core CPUs or GPUs. With precomputed cuboid features for each RGB-D image, the inference time is 2sec for each object category. With precomputed contextual features among all objects, the cascaded prediction framework takes less than 0.5sec on average to run.

Failure Cases An typical failure case for our algorithm is shown in the last row of Fig. 7, where missing depth values cause objects to be missed. While it is true that some small objects will be missed when we fail to detect their supporting surface, given the extreme difficulty of detecting small objects in highly cluttered indoor scenes, there are still substantial net benefits to exploiting support relationships for 3D detection. Some previous work was specifically designed to solve this issue [8, 23] by using CNN features in RGB images, and we believe incorporating a similar approach in our cascaded prediction framework might also help resolve this failure case.

6. Conclusions

We designed a 3D object detection system using latent support surfaces. Modeling the height of the support surface as a latent variable leads to improved detection performance for large objects, and contrains the search space for small object detectors. Via a cascaded prediction framework our detector achieves state-of-the-art performance on the SUN RGB-D dataset, demonstrating the effectiveness of modeling support surfaces in 3D object detection.

Acknowledgements This research supported in part by ONR Award Number N00014-17-1-2094.



Groundtruth Annotations for RGB-D Images

Our Final Stage 3D Detection Output

Figure 7. Visualizing our final stage 3D detections for objects with high confidence scores. Support surfaces are depicted with faded colors inside each large object. We show one failure case at the bottom: our algorithm failed to detect a dresser and a nightstand due to missing depth inputs (dark blue). As a result, the lamps supported by those objects are missed as well.

References

[1] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese. 3D semantic parsing of large-scale indoor spaces. In *CVPR*, pages 1534–1543, 2016.

[2] A. Bansal, B. Russell, and A. Gupta. Marr revisited: 2D-3D alignment via surface normal prediction. In *CVPR*, pages 5965–5974, 2016.

[3] C. Chen, M.-Y. Liu, O. Tuzel, and J. Xiao. R-CNN for small object detection. In *ACCV*, pages 214–230. Springer, 2016.

[4] X. Chen, K. Kundu, Y. Zhu, H. Ma, S. Fidler, and R. Urtasun. 3D object proposals using stereo imagery for accurate object class detection. In *TPAMI*, 2017.

[5] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia. Multi-view 3D object detection network for autonomous driving. In *CVPR*, 2017.

[6] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3D reconstructions of indoor scenes. In *CVPR*, 2017.

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

[8] Z. Deng and L. J. Latecki. Amodal detection of 3D objects: Inferring 3D bounding boxes from 2d ones in rgb-depth images. In *CVPR*, 2017.

[9] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 32(9):1627–1645, 2010.

[10] 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.

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

[12] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012.

[13] R. Girshick. Fast R-CNN. In *ICCV*, 2015.

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

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

[16] A. Gupta, A. A. Efros, and M. Hebert. Blocks world revisited: Image understanding using qualitative geometry and mechanics. In *ECCV*, pages 482–496. Springer, 2010.

[17] S. Gupta, P. A. Arbeláez, R. B. Girshick, and J. Malik. Aligning 3D models to RGB-D images of cluttered scenes. In *CVPR*, 2015.

[18] 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.

[19] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

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

[21] P. Hu and D. Ramanan. Finding tiny faces. *CVPR*, 2017.

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

[23] J. Lahoud and B. Ghanem. 2D-driven 3D object detection in rgb-d images. In *ICCV*, Oct 2017.

[24] C.-Y. Lee, V. Badrinarayanan, T. Malisiewicz, and A. Rabinovich. Roomnet: End-to-end room layout estimation. *ICCV*, 2017.

[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] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollr. Focal loss for dense object detection. In *ICCV*, 2017.

[27] D. Maturana and S. Scherer. Voxnet: A 3D convolutional neural network for real-time object recognition. In *IROS*, pages 922–928. IEEE, 2015.

[28] A. Mousavian, D. Anguelov, J. Flynn, and J. Košecká. 3D bounding box estimation using deep learning and geometry. In *CVPR*, pages 5632–5640. IEEE, 2017.

[29] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3D classification and segmentation. *CVPR*, 2017.

[30] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *NIPS*, 2017.

[31] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, pages 779–788, 2016.

[32] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. *arXiv preprint arXiv:1612.08242*, 2016.

[33] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[34] Z. Ren and E. B. Sudderth. Three-dimensional object detection and layout prediction using clouds of oriented gradients. In *CVPR*, pages 1525–1533, 2016.

[35] 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.

[36] T. Shao, A. Monszpart, Y. Zheng, B. Koo, W. Xu, K. Zhou, and N. J. Mitra. Imagining the unseen: Stability-based cuboid arrangements for scene understanding. *ACM Transactions on Graphics*, 33(6), 2014.

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

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

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

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

[41] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. *CVPR*, 2017.

[42] H. Su, S. Maji, E. Kalogerakis, and E. G. Learned-Miller. Multi-view convolutional neural networks for 3D shape recognition. In *ICCV*, 2015.

[43] S. Tulsiani, A. Kar, J. Carreira, and J. Malik. Learning category-specific deformable 3D models for object reconstruction. *TPAMI*, 2016.

[44] X. Wang, D. Fouhey, and A. Gupta. Designing deep networks for surface normal estimation. In *CVPR*, 2015.

[45] Z. Wu, S. Song, A. Khosla, X. Tang, and J. Xiao. 3D shapenets for 2.5D object recognition and next-best-view prediction. In *CVPR*, 2015.

[46] Y. Xiang, W. Choi, Y. Lin, and S. Savarese. Data-driven 3D voxel patterns for object category recognition. In *CVPR*, pages 1903–1911, 2015.

[47] C.-N. J. Yu and T. Joachims. Learning structural svms with latent variables. In *ICML*, pages 1169–1176. ACM, 2009.

[48] A. Yuille and A. Rangarajan. The concave-convex procedure. *Neural Computation*, 15(4):915–936, 2003.

[49] Y. Zhang, M. Bai, P. Kohli, S. Izadi, and J. Xiao. Deepcontext: Context-encoding neural pathways for 3D holistic scene understanding. In *ICCV*, Oct 2017.