Holistic Scene Understanding for 3D Object Detection with RGBD cameras

Dahua Lin Sanja Fidler Raquel Urtasun
TTI Chicago
{dhlin,fidler,rurtasun}@ttic.edu

Abstract

In this paper, we tackle the problem of indoor scene understanding using RGBD data. Towards this goal, we propose a holistic approach that exploits 2D segmentation, 3D geometry, as well as contextual relations between scenes and objects. Specifically, we extend the CPMC [3] framework to 3D in order to generate candidate cuboids, and develop a conditional random field to integrate information from different sources to classify the cuboids. With this formulation, scene classification and 3D object recognition are coupled and can be jointly solved through probabilistic inference. We test the effectiveness of our approach on the challenging NYU v2 dataset. The experimental results demonstrate that through effective evidence integration and holistic reasoning, our approach achieves substantial improvement over the state-of-the-art.

1. Introduction

One of the fundamental problems in indoor robotics is to be able to reliably detect objects in 3D. This is important as robots have to be able to navigate the environment as well as interact with it, e.g., accurate 3D localization is key for successful object grasping. Over the last decade a variety of approaches have been developed in order to infer 3D objects from monocular imagery [12, 6, 19]. The most successful approaches extend the popular (2D) deformable part-based model [5] to perform category-level 3D object detection [12, 6, 19].

While 3D detection is extremely difficult when employing still images, the use of additional information such as video or depth sensors is key in order to solve the inherent ambiguities of the monocular setting. Numerous approaches have been developed that model both appearance and depth information to score object detectors in 3D [16, 10, 24, 20, 8] , showing improved performance over the monocular setting.

Objects, however, are not randomly placed in space, but respect certain physical and statistical properties of the 3D world. For example, we are more likely to see a person sitting on top of a camel, then than other way of around. We typically see a bed resting against the wall in bedrooms, and not hanging from the bathroom ceiling. Exploiting physical and contextual relationships between objects and the environment is key to achieve a high precision in semantic tasks such as segmentation [15, 22, 29] and detection [9, 8].

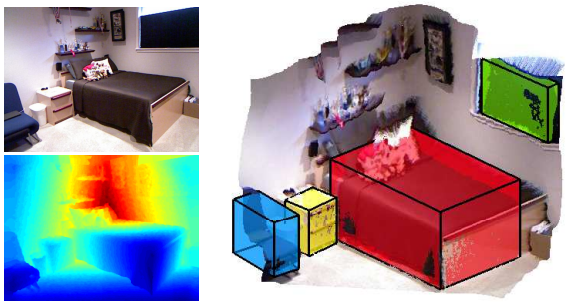

Figure 1. RGBD images provide both appearance and geometric information for indoor scene understanding. We leverage this information as well as contextual relations to detect and recognize objects in indoor scenes. In particular, we first generate candidate cuboids through an extension to CPMC and then use a CRF to assign semantic labels to them.

In this paper we are interested in exploiting RGB-D imagery to perform category level 3D object detection. We represent the objects in the world in terms of 3D cuboids and model the physical and statistical interactions between the objects and the environment (the scene) as well as interactions between objects. Towards this goal, we develop an approach that extends the CPMC framework [3] to generate cuboid hypotheses in point clouds by placing them tightly around bottom-up 3D region candidates. Our region candidates are ranked according to "objectness" in appearance and are encouraged to respect occlusion boundaries in 3D. We formulate the joint detection problem in a conditional random field to model the contextual relationships between objects in 3D. In particular, we encourage the object labels to agree with the scene type, the objects to follow statistical geometric relationships relative to the scene layout, such as proximity to the walls in the room, and absolute size in 3D, as well as inter-object interactions that encourage certain types of support relationships and spatial co-occurances.

We evaluate our approach on the challenging NYUv2 RGB-D dataset [23] and show significant improvements

over existing methods [13], demonstrating the effectiveness of our contextual model over the task-isolated baseline.

2. Related Work

Previous approaches to object recognition in range data [26] typically assumed that the target object is either segmented from the background, or that a detailed 3D model is available [4, 14]. Localizing generic object classes in "the wild" (i.e., highly cluttered scenes) is, however, much more difficult. We focus our review in this domain.

Contextual models incorporating strong priors about the physical world become increasingly popular with the recent release of RGB-D datasets [22] . These models mainly focus on segmentation by enforcing statistical physical constraints. Silberman et al. [23] reason about support surfaces and structural object classes (such as "ground", "furniture" and "props"). [22, 28, 15] do segmentation based on a variety of carefully designed 2D appearance and 3D geometry features. [22] reasons about spatial transitions between superpixels based on RGB and depth information, while [15] enforces statistical coocurrences of 3D spatial relations such as near or on top of. Our approach shares similarities with the physical relationships modeled by these methods. However, we reason at the level of full objects, represented as cuboids, which allows us to better capture the statistical object-object interactions in 3D.

Most 3D object detectors make use of both appearance and depth information, but do not exploit the inter-object relationships. Lai et al. [16, 10] extend the popular deformable part based model (DPM) [5] to include disparity and 3D size features. Walk et al. [25] use depth statistics to learn to enforce height constraints in pedestrians. Sun et al. [24] augment the implicit shape model with depth information, by voting with patches in 3D. Scene-object geometry is exploited for indoor scenarios in [8], adopting a sliding window approach that uses appearance as well as 3D features such as normals, height and distance to the ground. In [13], object candidates are found by fitting 3D cuboids in point clouds, but do not reason about the class of the object.

Related work on contextual object detection in 3D has been sparser. Most approaches use monocular imagery to infer the objects [19], and use the context of the problem to parametrize the model. In [12, 6, 17], objects in indoor scenarios are represented as cuboids aligned with the major axes of the room. These approaches use estimated room layouts to rescore object detections in 3D, which softly imposes the constraints that the objects do not penetrate the room, and rest on the floor. Gupta et al. [9] represent the objects in the world as cuboids and model physical constraints among them encoding that objects need to be supported in 3D, and cannot penetrate each other. Their model uses monocular imagery, while in our approach we are interested in exploiting the richer RGBD data. For video, Geiger et al. [7] exploit the strong priors in outdoor scenarios to reason jointly about objects and road intersections.

3. 3D detection with RGBD Imagery

We generate a set of cuboids via candidate 3D "objectness" regions that are encouraged to respect intensity as well as occlusion boundaries in 3D. To generate the regions we build on CPMC [3], which achieved state-of-the-art performance on the very challenging PASCAL segmentation challenge. We propose a simple extension to generate candidate class-independent object regions by exploiting both depth as well as appearance cues.

3.1. Generating bottom-up 3D region candidates

CPMC [3] uses parametric min-cut to generate a wide variety of foreground candidates from equally spaced seeds. The overall objective is to minimize an energy function over pixel labels $\{x_1, \ldots, x_N\}$, with $x_i \in \{0, 1\}$ and N the total number of pixels. In particular, the energy is defined as

$$E^{\lambda}(X) = \sum_{u \in \mathcal{V}} C_{\lambda}(x_u) + \sum_{(u,v) \in \mathcal{E}} V_{uv}(x_u, x_v), \quad (1)$$

with $\lambda \in \Re$, $\mathcal{V}$ the set of all pixels, $\mathcal{E}$ the edges between neighboring pixels, and C_{λ} the unary potentials:

$$C_{\lambda}(x_u) = \begin{cases} 0, & \text{if } x_u = 1,\ u \notin \mathcal{V}_b \\ \infty, & \text{if } x_u = 1,\ u \in \mathcal{V}_b \\ \infty, & \text{if } x_u = 0,\ u \in \mathcal{V}_f \\ f(x_u) + \lambda, & \text{if } x_u = 0,\ u \notin \mathcal{V}_f \end{cases} \quad (2)$$

Here λ is an offset and is used to generate different solutions of the objective function. The function f is defined as $f(x_u) = \ln p_f(x_u) - \ln p_b(x_u)$, where p_f represents the probability distribution of pixel i belonging to foreground. To exploit both appearance and depth, we define

$$p_f(i) = \exp\left(-\gamma \cdot \min_j (\alpha \cdot ||I(i) - I(j)|| + (1-\alpha) \cdot ||D(i) - D(j)||)\right)$$

with D the depth, and I the rgb imagery. Here j indexes the representative pixels in the seed region, selected as centers resulting from a k-means algorithm ($k = 5$), as in [3], and γ a scaling factor.

The pairwise term V_{uv} penalizes assignments of different labels to similar neighboring pixels:

$$V_{uv}(x_u, x_v) = \begin{cases} 0, & \text{if } x_u = x_v \\ g(u, v), & \text{if } x_u \neq x_v \end{cases} \quad (3)$$

The similarity between two adjacent pixels is based on the gPb response [18]: $g(u, v) = \exp\left(-\frac{max(gPb(u), gPb(v))}{\sigma^2}\right)$. To exploit depth information, we use both gPb_{rgb} computed on the original image as well as gPb_{depth} computed on the

depth image, and combine them linearly: $gPb_{rgbd} = \alpha \cdot gPb_{rgb} + (1-\alpha) \cdot gPb_{depth}$, with α set to 0.3. This simple combination has been shown to improve boundary detection in RGBD imagery by roughly 2% [28].

3.2. Fitting the Cuboids

We generate cuboids from the candidate regions. Specifically, we select top K candidate regions ranked by the objectness scores [3], after performing non-maxima suppression (we use 0.5 as max overlap), and then generate candidate cuboids by fitting a 3D cube around each candidate region. A natural idea to accomplish this is to map the pixels in a given region into 3D coordinates, and find the minimal bounding cube around them. However, this approach is sensitive to noise, as a single outlier point may completely change the fitted cube. To improve the robustness, we consider a variant that instead of finding a minimal bounding cube to all the points, returns the minimal cube that contains 95% of the 3D points. In this way, outlier points are allowed to be precluded from the cube. Additionally, as most objects of interest are parallel to the floor, we enforce this constraint, reducing the estimation problem to an optimization problem with only one variable – the orientation along the x-z plane. This can be solved using direct search.

4. Indoor Scene Model

Assigning class labels to candidate cuboids is a challenging task. Feature-based approaches usually face difficulties caused by pose variation, object occlusion, and poor illumination. To tackle this problem, we develop a conditional random field (CRF) model, which integrates appearance, geometry, as well as contextual information (*e.g.*, scene-object relations and spatial configurations) to improve the recognition accuracy.

4.1. Contextual model for 3D Object Detection

We employ candidate cuboids obtained by the method described in the previous section as input to our holistic model. We characterize each cuboid by both a 2D bounding box and a 3D bounding cube. This representation makes it convenient to utilize information from both 2D and 3D domains. In this paper, we are interested in simultaneously classifying the scene and assigning semantic labels to candidate cuboids. Formally, we denote the scene variable by $s \in \{1, \ldots, S\}$, and the objects with $y_i \in \{0, 1, \ldots, C\}$, where S and C are respectively the number of scene and object classes. Note that the value of y_i can be 0, which indicates that the cuboid is a false positive.

We define a CRF in terms of these variables, which exploits appearance features, geometric properties as well as semantic relations between objects and the scene (see Fig. 2 for an illustration). In particular, the model consists of multiple potential functions, which can be expressed generally

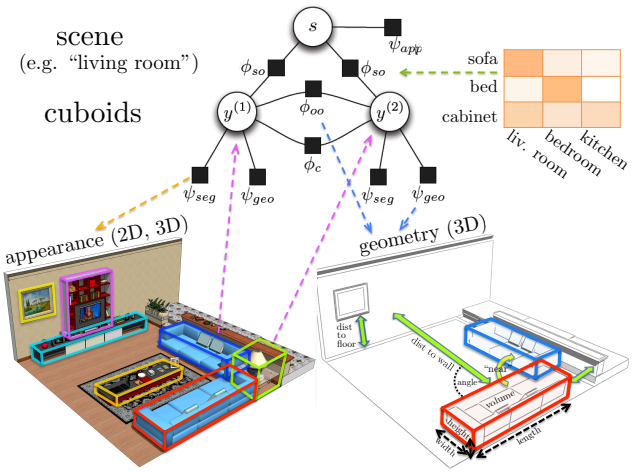

Figure 2. We use a CRF model to integrate various contextual relations. Each cube is associated with unary potentials that characterize both the appearance and geometric properties. In addition, the pairwise potentials between the scene and the objects, as well as between the objects themselves encourage certain configurations.

in the following form:

$$p(\mathbf{y}, s) = \frac{1}{Z(\mathbf{x})} \exp\left(w_s \psi_s(s) + \sum_{t \in \mathcal{U}} w_t \sum_{i=1}^{m} \psi_t(y_i) + \sum_{p \in \mathcal{A}} w_p \sum_{(i,i') \sim \mathcal{P}_p} \phi_p(y_i, y_{i'}) + \sum_{m \in \mathcal{B}} w_m \sum_i \phi_m(s, y_i) \right).$$

There are four categories of potentials: ψ_s, a unary potential associated with the scene label, $\{\psi_t\}_{t \in \mathcal{U}}$, unary potentials defined on object labels, and $\{\phi_p\}_{p \in \mathcal{A}}$, $\{\phi_m\}_{m \in \mathcal{B}}$, binary potentials that capture the contextual relations between the scene and the objects, as well as between the objects themselves. Each potential is associated with a weight shared across cliques, which we learned from training data. We now describe the potentials in more detail.

Scene appearance: In order to incorporate global information about the scene without making hard decisions, we define a unary potential over the scene label s as

$$\psi_s(s = u) = \sigma(t_u),$$

where t_u denotes the classifier score for scene class u and σ is the logistic function. We utilize the scene classification approach by Xiao et al. [27] to obtain the scores t_u.

Ranking potential: We employ the segment ranking framework of Carreira et al. [2] to obtain the cuboid detection scores. Since our cuboid hypotheses were generated via bottom-up region proposals, each cuboid thus has a 3D segment associated with it. We use these segments to train a ranker that, given an input segment, predicts how much it overlaps with a ground-truth cuboid. In particular, we use

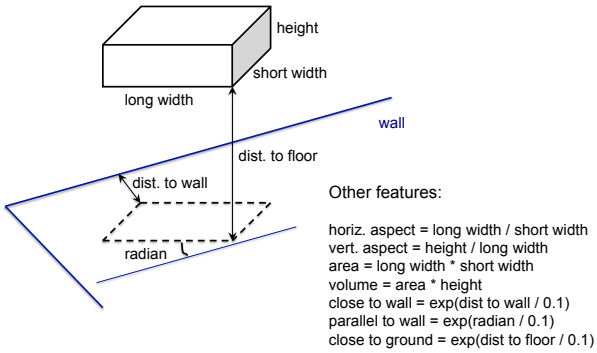

Figure 3. Cube geometric properties: close-to-wall, parallel-to-wall and close-to-ground are computed based on the corresponding distances (using exp), 0.1 is determined empirically.

the code released by [2] to train an SVR predictor. We use the output score as a unary potential in our model:

$$\phi_{rank}(y_i = l) = f(l),$$

where $f(l)$ represents the predicted overlap of object candidate y_i with ground-truth. For the background class, we use a constant threshold.

Segmentation potential: Since the number of training examples of each class is limited, classifiers trained to recognize full objects in our experience did not work particularly well. This motivates the use of segmentation potentials as unaries for our cuboid hypotheses. We employ the approach of Ren et al. [28], which trains a classifier on a set of (smaller) superpixels. The approach uses kernel descriptors (KDES), a framework that uses different aspects of similarity (kernel) to derive patch descriptors. Kernel descriptors are aggregated over superpixels and transformed using effcient match kernels (EMK). We used six types of RGB-D kernel descriptors: *gradient*, *color*, *local binary pattern*, *depth gradient*, *spin/surface normal*, and *KPCA/self-similarity*. This approach assigns a score to each superpixel. In order to derive the potential for a cuboid, we project the cuboid onto the image plane and obtain a convex hull. Then, we use the weighted average of the scores as the value of the potential, where the weights are defined to be the intersection areas between superpixels and the projected convex hull. Note that this kind of potential is agnostic to the size of the object. Our model also employs geometric potentials such as absolute size and aspect ratio in 3D, which ensure that the joint model recovers physically valid hypotheses.

Object geometry: Geometric properties of an object are an important source of discriminative information which is complementary to appearance and depth features. For example, a bed is often flat and occupies a large area, while a refrigerator is typically taller. As illustrated in Fig. 3, we capture the geometric properties using a vector with 10 components describing the 3D cuboid: *height*, *longer width*, *shorter width*, *horizontal aspect ratio*, *vertical aspect ratio*, *area*, *volume*, *parallel-to-wall*, *close-to-wall*, and *close-to-ground*. Note that these properties capture not only the intrinsic attributes of an object, but also its position relative to the scene layout. For example, one can roughly reason about the supporting relation between an object and a wall (or floor) from the values of the last two components. This has been proven very useful in the indoor setting [23]. To use geometric properties in our model, we train an SVM with RBF kernel on the geometric features, one for each class, and use the resultant scores r_l as unary potentials for the candidate cubes, i.e., $\phi_{geom}(y_i = l) = r_l$.

Semantic context: Context (*e.g.*, the room that an object resides in or other objects in the same scene) often provides useful information for object recognition. For example, a bed is much more likely to be in a bedroom than in a kitchen. In this framework, we consider two co-occurrence relationships: *scene-object* and *object-object*. The potential values are estimated from the training set by counting the co-occurence frequencies.

Specifically, the *scene-object potential* is defined to be

$$\phi_{so}(s = k, y_i = l) \triangleq \frac{1}{N_{tr}} \sum_{j=1}^{N_{tr}} \sum_{i=1}^{m_j} \mathbb{1}(s_j = k, y_i^{(j)} = l),$$

where $y_i^{(j)}$ is the i-th cuboid on the j-th training example, m_j is the number of objects in the j-th scene, N_{tr} is the number of training images, and $\mathbb{1}(\cdot)$ the indicator function, which equals 1 when the enclosed condition holds.

The *object-object potential* is defined as

$$\phi_{oo}(y_i = l, y_k = l') \triangleq \frac{1}{N_{tr}} \sum_{j=1}^{N_{tr}} \mathbb{1}(\exists i, i' : y_i^{(j)} = l, y_{i'}^{(j)} = l').$$

It is worth noting that in the case where multiple instances of classes l and l' exist in a scene, the co-occurence of l and l' is counted as 1 for this scene. We found empirically that this is more robust than using the number of instance pairs.

Geometric context: We introduce two potentials to exploit the spatial relations between cuboids in 3D: (1) *close-to* relation (*e.g.*, a chair is typically near a desk), and (2) *on-top-of* relations. It is important to note that unlike the *close-to* relation, the *on-top-of* relation is asymmetric. For example, a television can be on top of a chair, while the contrary is very unlikely. As the size and position of each object is available, we can roughly determine spatial relations using some simple geometric rules. More specifically, we consider two objects as being *close to* each other if the

(Hausdorff) distance between them is less than 0.5 meters. Furthermore, we say object A is *on top of* object B if A is higher than B and 80% of A's ground projection is contained within the one of B. Again, the potential values are defined to be the frequencies that specific configurations appear in the training set. Specifically, the *close-to* potential, $\phi_c(l, l')$, is defined to be the normalized frequency that an object of class l is close to that of class l'; while the *on-top-of* potential, $\phi_t(l, l')$ is the normalized frequency that an object of class l' is on top of one of class l.

4.2. Learning and Inference

Our energy is defined in terms of a log-linear model. We learn the weights by employing the primal dual learning framework of [11]. This allow us to minimize both hinge loss and log-loss within the same mathematical framework. This framework is particularly appealing as it does not require to compute neither the partition function, not do loss augmented inference in each step of the gradient computation. Instead, it does block coordinate descent in the approximated primal, alternating between updating the messages and the weights. As it does not require the messages to run to convergence, it results in orders of magnitude speed-ups over classical CRF learning approaches.

We define loss functions which decompose into unary terms. In particular, we define a 0-1 loss over the scene type, and a 0-1 loss over the cuboid detections.

$$\Delta_{det}(y_i = l, \hat{y}_i) = \begin{cases} 1 & \text{if (IOU} \geq 0.5 \wedge \hat{y}_i = l) \\ 0 & \text{otherwise} \end{cases}$$

with $\hat{y}_i$ the ground truth label. For inference, we compute the MAP estimate by computing

$$\max_{\mathbf{y},s} p(\mathbf{y}, s)$$

This is in general NP-hard for the class of energies we exploit in this paper. We resort to the approximated algorithm of [21], which combines LP relaxations and dual decomposition to obtained a parallel algorithm which is guaranteed to converge. This worked very well in practice, not imposing any restrictions in the types and order of potentials nor on the structure of the graph.

5. Experimental Evaluation

We tested the proposed framework on the NYUv2 [23] RGB-D dataset, and compare it with related approaches. The dataset contains 1449 scenes, each associated with an RGB image and a depth map. The original segmentations of NYUv2 assign pixels to 894 object classes with different names, which is difficult to manipulate in practice. To address this problem, we manually clean up the class list, merging similar classes (*e.g.*, *table* and *desk*) and discarding those that appear sporadically. This results in 21 object classes. Note that we do not consider *floor*, *ceiling*, and *wall*. These classes are "special" as they define the scene layout. We find them through 3D Hough transform [1] instead of object detection.

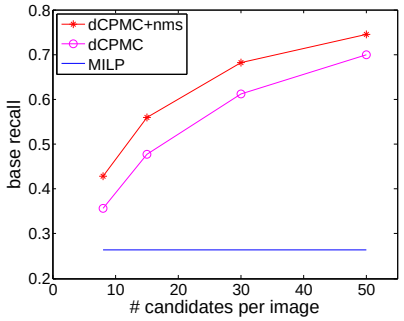

Figure 4. Comparison of different methods in cuboid detection. Performance measured in recall, *i.e.* the fraction of objects which are covered by at least one candidate cube (with overlap > 0.5.)

For each object instance that belongs to one of these 21 classes, we generate a ground-truth cuboid using the fitting algorithm described in Sec. 3. We inspected all these ground-truths and found that most of them (over 95%) fit very well and are good for performance evaluation. We also identified cubes that were poorly fitted due to imperfect segmentation (less than 5%) and ignored them in both training and performance evaluation. In this way, we obtained 6680 ground-truth cubes in total.

We partitioned the dataset into two disjoint subsets, respectively for training and testing, using the same split as [23]. In particular, the training set contains 795 scenes (with 3630 objects), and the testing set contains 645 scenes (with 3050 objects). In what follows, we will first compare the performance of two major components (*cuboid detection* and *scene & object classification*) to state-of-the-art methods, and then examine the overall performance.

Performance of cuboid detection: The primary goal of the cuboid detection stage is to generate a reasonable amount of candidate cuboids, such that a significant portion of the true objects is contained in the candidate set. The performance of a cuboid detector is measured in terms of *base recall*, given a fixed K_c – the maximum number of candidates for each scene. Specifically, an object is said to be *recalled* if there is a candidate cube which overlaps with it more than 0.5 IOU. The *base recall* is defined to be the fraction of ground-truth objects that are recalled by the candidate set.

We compared three approaches here: (1) CPMC (extended to use depth information), (2) CPMC with non-maximal suppression, and (3) the mixed integer programming (MILP) algorithm of [13]. Specifically, we used CPMC to propose 150 cubes for each scene, and selected top K ones as candidates. By varying K, we obtain curves that show how the number of candidates per image influ-

configuration	object	scene
scene appearance only	-	55.20
segmentation only	54.46	-
geometry only	42.85	-
seg. + geo.	59.02	-
app. + scene-obj	55.87	57.49
app. + obj-obj	54.49	55.20
app. + obj-spa	55.61	55.20
unaries + scene-obj	60.00	57.65
unaries + obj-obj	58.92	55.20
unaries + obj-spa	59.41	55.20
unaries + scene-obj + obj-obj	60.13	58.56
unaries + scene-obj + obj-spa	60.33	58.10
unaries + obj-obj + obj-spa	59.28	55.20
all combined	**60.49**	**58.72**

Table 1. Performance of scene & object classification on ground-truth cuboids in terms of the percentage of correct labels. Here, "scene-obj", "obj-obj", and "obj-spa" respectively refers to scene-object co-occurances, object-object co-occurances, and spatial relations between objects. In addition, "app." refers to appearance-based potentials, including segmentation and scene appearance, and "unaries" refers to all unary potentials.

ences the base-recalls (see Fig. 4). Note that the MILP results are directly acquired from the authors of [13], and therefore we were not able to vary the candidate numbers as for our approach.

On average, MILP generates about 8 candidates per scene. To make a fair comparison, we specifically test the setting with $K = 8$. Under this setting, the base recall of MILP is 0.263, while CPMC yields 0.356 and 0.428 (using non-maximal suppression). As K increases, the base recalls of both CPMC and CPMC+nms increase. We can see that CPMC+nms consistently outperforms CPMC, and attains nearly 75% when $K = 50$. One of the reason that non-maximal suppression helps is that it actively removes redundant cuboids, and consequently the detector is able to cover more objects using fewer candidates.

Performance of classification: To examine the performance of the CRF model in isolation, we test it on the ground-truth cuboids. In this way, the errors introduced in the detection stage are not considered. We measure the performance in terms of *classification accuracy*, that is, the number of the correctly classified objects over the total. We consider various configurations of the CRF model that incorporate different subsets of potentials, so as to analyze their individual effect. Particularly, we test settings using only feature-based potentials to set up the baselines, which are equivalent to feature-based classification. Table 1 compares their performance. We observed: (1) We get accuracies at 54.46% and 55.20% respectively, when the appearance and geometry features are used in isolation. However, the combination of them yields considerably higher accuracy (59.02%). This clearly shows that these two features are complementary. (2) The use of contextual potentials improves performance. For example, using scene-object co-occurances moderately raises the object labeling accuracy from 59.02% to 60.00% and scene classification accuracy from 55.20% to 57.65%. (3) The accuracy increases as more potentials are added to the framework. We reach the highest accuracy at 60.49% when the full CRF model is used. This is a considerable improvement compared to the baseline (54.46%). The scene classification performance also increases (from 55.20% to 58.72%).

Overall Performance: To test the integrated performance, we considered different combination of cuboid detectors and CRF configurations. To put our performance into perspective, we also tested DPM [5], a state-of-the-art 2D detector, on the same set of data[1]. The performance is measured in terms of F1-score, the harmonic mean of recall and precision. Here, an object is said to be *recalled* if there is a cuboid *with the same class label* that overlaps with it by more than 50%. This is different from the *base recall* in detector evaluation, where the cuboids have not been labeled. Table 2 summarizes the results. The poor performance of DPM indicates that 2D object detectors that focuses on objects with regular structures encounter significant difficulties in a general indoor setting. Our approach, which explicitly takes advantage of 3D information and contextual relations, consistently outperforms DPM (achieving over $10x$ higher precision while maintaining a comparable recall). The improvement is also reflected by the substantial gain in F1 scores. Moreover, we can observe that the use of geometric feature, scene-object relations, and other potentials further improves the overall performance. Table 3 depicts the class-specific performances. In an indoor environment, objects of some classes (*e.g.*, cabinets, chair, and shelf) appear much more frequently than others, and thus play a more important role in understanding the scene. To emphasize these classes, we show the re-weighted F1-score of each class, which is defined as $F1_{reweight} \triangleq F1 \cdot m/m_{max}$. Here, m is the number of testing samples in the class of interest, and m_{max} is that of the most frequent class. We can see that the use of object geometry and contextual information leads to notable improvements, especially over frequent classes. Fig. 5 shows example detections of our model and compares them with GT cuboids.

Computational Complexity: With all the cuboids ready, both learning and inference are very efficient. With $K = 15$, the learning of the full CRF model takes about 2 minutes on a workstation with Intel i7 quad-core CPU (using

[1]We applied DPM detectors for different classes respectively, and sorted the detected objects by their decreasing scores of all classes together. We kept only the top K detected objects in each scene for performance evaluation.

	dpm			seg.			seg.+geo.			seg.+geo.+rank			+scene-object			all		
	recall	prec.	F1	recall	prec.	F1	recall	prec.	F1	recall	prec.	F1	recall	prec.	F1	recall	prec.	F1
MILP	-	-	-	31.41	6.75	11.11	14.25	40.90	21.13	15.09	39.90	21.90	15.21	41.00	22.19	15.18	41.92	**22.3**
K = 8	38.22	4.47	8.01	25.76	33.13	28.98	25.53	37.02	30.22	32.14	38.82	35.17	32.79	37.96	35.18	31.67	39.68	**35.23**
K = 15	40.47	3.56	6.54	29.11	27.59	28.33	30.69	28.30	29.44	35.57	34.29	34.92	36.03	33.93	34.95	34.19	37.04	**35.56**
K = 30	44.52	2.62	4.96	32.02	20.25	24.81	33.48	20.70	25.58	38.29	28.30	32.54	39.56	27.67	32.57	30.19	36.68	**33.10**
K = 50	48.71	1.75	3.37	27.59	16.82	20.90	28.74	17.52	21.77	33.83	27.41	**30.28**	35.47	26.33	30.22	32.16	27.80	29.82

Table 2. Performances of the integrated framework. Each row corresponds to a specific detection setting, and each column corresponds to a model configuration. In particular, the first row shows the results obtained using the MILP detector [13], while the other four rows corresponds to the setting where a tunable detector is used to generate $K = 8, 15, 30, 50$ candidates per images. The first column shows the results by DPM, and the other five columns show the results obtained by our framework with different combinations of potentials. The performance are measured in terms of recall, precision, and F1-score. Note that these numbers are percentages.

	mantel	counter	toilet	sink	bathtub	bed	headboar	table	shelf	cabinet	sofa	chair	chest	refriger	oven	microwav	blinds	curtain	board	monitor	printer	overall
# samples	12	137	30	55	24	153	32	341	237	566	213	519	135	42	31	39	154	139	53	113	25	3050
seg.	0.0	6.7	1.7	2.7	0.2	10.6	0.9	14.6	12.3	27.8	11.8	27.9	4.1	0.4	0.6	0.9	8.5	7.4	1.6	4.5	0.5	28.22
seg.+geo.	0.0	7.4	1.5	2.9	0.2	11.6	1.0	14.7	13.0	30.7	12.5	29.1	4.0	0.6	0.5	0.7	8.9	8.0	1.8	4.8	0.5	29.25
seg.+geo.+rank.	0.0	11.2	2.6	3.4	0.3	16.6	1.3	19.1	18.2	44.3	18.4	39.4	7.7	1.2	0.6	2.1	14.0	11.7	2.4	8.8	0.0	34.78
+scene-obj.	0.0	9.7	2.4	3.8	0.3	15.6	1.3	19.7	17.6	41.9	17.9	37.2	8.1	1.5	0.9	1.4	13.5	11.0	2.6	8.2	0.7	34.94
all	0.0	10.9	2.6	4.0	0.3	16.7	1.3	20.8	18.7	44.5	18.6	40.3	8.0	1.4	0.9	1.8	13.9	11.5	2.8	9.0	1.0	35.04

Table 3. Class-specific performances obtained using CPMC-nms detector (with $K = 15$) + our CRF model (with four different configs). This is combination yields the best overall performance. Note that the numbers of testing samples in different classes are unbalanced. We show the reweighted F1 scores, which are defined as $F1 \cdot m/m_{max}$, thus emphasizing frequent classes.

4 threads), and the inference for over the entire testing set takes about 10 seconds ($15ms$ per scene). Empirically, the time needed to learn the model parameters or to infer the labels scales up linearly as the number of potentials increases.

6. Conclusion

We have developed an integrated framework to detect and recognize 3D cuboids in indoor scenes, an provided a detailed evaluation on the challenging NYU v2 dataset. Our experiments demonstrate that our approach consistently outperforms state-of-the-art detectors by effectively combining segmentation features, geometric properties, as well as contextual relations between objects. In particular, the combination of CPMC-nms with the full CRF model achieves F1-score at 36%, which is a remarkable improvement over DPM. The framework developed in this work is very flexible. We believe that it can be extended to incorporate information from other sources (e.g., video), thus further improving the performance.

References

[1] D. Borrmann, J. Elseberg, K. Lingemann, , and A. Nchter. The 3d hough transform for plane detection in point clouds - a review and a new accumulator design. *3D Research*, (2), 2011.

[2] J. Carreira, R. Caseiroa, J. Batista, and C. Sminchisescu. Semantic segmentation with second-order pooling. In *ECCV12*, 2012.

[3] J. Carreira and C. Sminchisescu. CPMC: Automatic Object Segmentation Using Constrained Parametric Min-Cuts. *TPAMI*, 2012.

[4] B. Drost, M. Ulrich, N. Navab, and S. Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *CVPR*, 2010.

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

[6] S. Fidler, S. Dickinson, and R. Urtasun. 3D Object Detection and Viewpoint Estimation with a Deformable 3D Cuboid Model. In *NIPS*, 2012.

[7] A. Geiger, C. Wojek, and R. Urtasun. Joint 3d estimation of objects and scene layout. In *NIPS*, 2011.

[8] S. Gould, P. Baumstarck, M. Quigley, A. Y. Ng, and D. Koller. Integrating visual and range data for robotic object detection. In *ECCV w. on S. Fusion Alg. & Appl.*, 2008.

[9] A. Gupta, A. Efros, and M. Hebert. Blocks world revisited: Image understanding using qualitative geometry and mechanics. In *ECCV*, 2010.

[10] H. Hattori, A. Seki, M. Nishiyama, and T. Watanabe. Stereo-based pedestrian detection using multiple patterns. In *BMVC*, 2009.

[11] T. Hazan and R. Urtasun. A primal-dual message-passing algorithm for approximated large scale structured prediction. In *NIPS*, 2010.

[12] V. Hedau, D. Hoiem, and D. Forsyth. Thinking Inside the Box: Using Appearance Models and Context Based on Room Geometry. In *ECCV*, 2010.

[13] H. Jiang and J. Xiao. A Linear Approach to Matching Cuboids in RGBD Images. In *CVPR*, 2013.

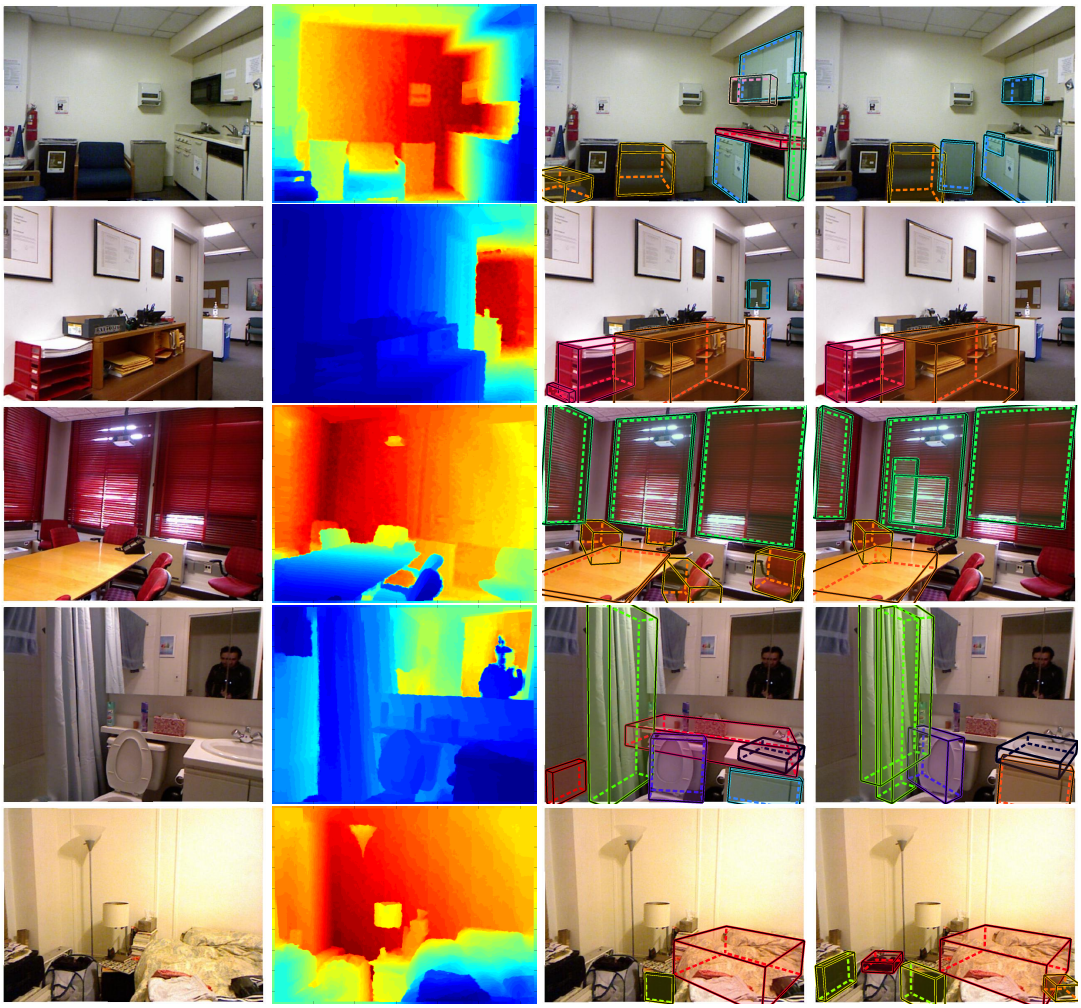

Figure 5. A few examples for our 3D detections. From left to right: image, depth, GT, our approach.

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

[15] H. Koppula, A. Anand, T. Joachims, and A. Saxena. Semantic labeling of 3d point clouds for indoor scenes. In *NIPS*, 2011.

[16] K. Lai, L. Bo, X. Ren, and D. Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *ICRA*, 2011.

[17] D. C. Lee, A. Gupta, M. Hebert, and T. Kanade. Estimating Spatial Layout of Rooms using Volumetric Reasoning about Objects and Surfaces. In *NIPS*, 2010.

[18] P.Arbelaez, M.Maire, C.Fowlkes, and J.Malik. Contour detection and hierarchical image segmentation. In *TPAMI'11*.

[19] B. Pepik, M. Stark, P. Gehler, and B. Schiele. Teaching 3d geometry to deformable part models. In *CVPR*, 2012.

[20] K. Saenko, S. Karayev, Y. Jia, A. Shyr, A. Janoch, J. Long, M. Fritz, and T. Darrell. Practical 3-D Object Detection Using Category and Instance-level Appearance Models. In *IROS*, 2011.

[21] A. Schwing, T. Hazan, M. Pollefeys, and R. Urtasun. Distributed message passing for large scale graphical models. In *CVPR*, 2011.

[22] N. Silberman and R. Fergus. Indoor scene segmentation using a structured light sensor. In *wrk. 3DRR*, 2011.

[23] N. Silberman, P. Kohli, D. Hoiem, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

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

[25] S. Walk, K. Schindler, and B. Schiele. Disparity statistics for pedestrian detection: Combining appearance, motion and stereo. In *ECCV*, 2010.

[26] K. Wu and L. M. Recovering parametrics geons from multiview range data. In *CVPR*, 1994.

[27] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010.

[28] L. B. Xiaofeng Ren and D. Fox. Rgb-(d) scene labeling: Features and algorithms. In *CVPR*, 2012.

[29] Y. Yao, S. Fidler, and R. Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *CVPR*, 2012.