Box In the Box: Joint 3D Layout and Object Reasoning from Single Images

Alexander G. Schwing
ETH Zurich
aschwing@inf.ethz.ch

Sanja Fidler
TTI Chicago
fidler@ttic.edu

Marc Pollefeys
ETH Zurich
pomarc@inf.ethz.ch

Raquel Urtasun
TTI Chicago
rurtasun@ttic.edu

Abstract

In this paper we propose an approach to jointly infer the room layout as well as the objects present in the scene. Towards this goal, we propose a branch and bound algorithm which is guaranteed to retrieve the global optimum of the joint problem. The main difficulty resides in taking into account occlusion in order to not over-count the evidence. We introduce a new decomposition method, which generalizes integral geometry to triangular shapes, and allows us to bound the different terms in constant time. We exploit both geometric cues and object detectors as image features and show large improvements in 2D and 3D object detection over state-of-the-art deformable part-based models.

1. Introduction

Despite the fact that our world is three-dimensional, many approaches to object recognition employ sliding window paradigms and rarely include knowledge about the inherent physical constraints. However, endowing computers with spatial reasoning allows prediction of navigable space, one of the main goals in robotic vision.

In the past few years, a variety of approaches have been proposed in order to extract the 3D layout of rooms from single images [12, 13, 31, 19, 22, 25, 26]. Common to all these approaches is the use of the Manhattan world properties of indoor scenes, which assume that the room is aligned with the three dominant and orthogonal directions, defined by the vanishing points. As a consequence a simple parameterization exists, since, given the vanishing points, only 4 degrees of freedom are needed to represent the layout [12, 31]. By exploiting the inherent decomposition of additive energy functions, real-time inference was shown to be possible with this parameterization [25, 26].

Objects, however, populate rooms. The first attempts to incorporate object reasoning into semantic parsing of indoor scenes treated objects as clutter, and focused on removing them from the layout estimation [12, 31]. But objects are more than just clutter. If we could estimate them reliably, we should be able to better predict the room layout. Similarly, if we could estimate the layout we should be able to better parse the objects. For example, we could employ the physical constraints inherent to the problem, as objects are typically fully contained within the room. This strategy is utilized in a variety of approaches [25, 19, 22], where object candidates are employed to score the layout. Alternatively, the layout has been employed to better detect objects. In [14] and [6], a few candidate layouts are utilized in order to re-rank 3D object detections.

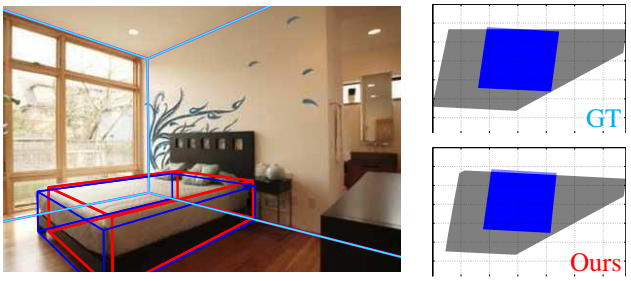
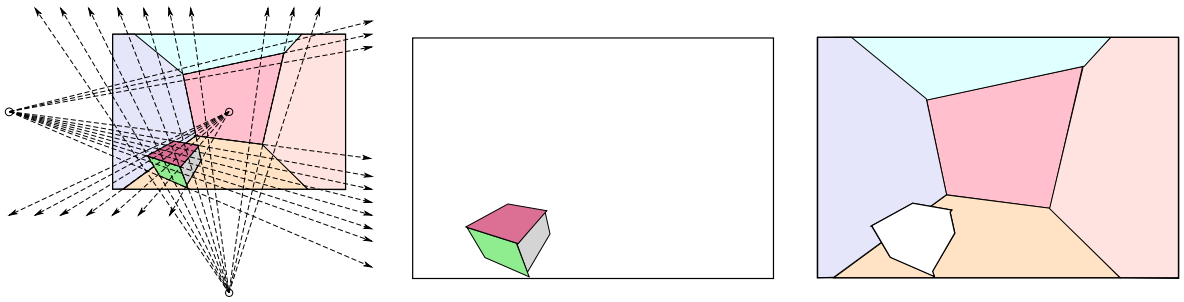


Figure 1. Image with overlayed ground truth (blue, cyan) and our detection result (red, magenta) as well as the ground truth (GT) floor plan (top right) and our prediction (Ours) (bottom right).

Despite these numerous efforts, most approaches trade the complexity of one of the tasks by proposing a small set of candidates. As a consequence, the space of hypotheses is not well explored resulting in sub-optimal solutions. Furthermore, most approaches employ generic cuboids which are typically generated from bottom-up reasoning.

In contrast, in this paper we jointly reason about both the exponentially many layouts as well as the exponentially many object locations and sizes. Our approach makes use of both 2D and 3D object detectors as well as geometric features, and results in very accurate predictions from monocular imagery, as shown in Fig. 1. The inherent difficulty of the joint estimation comes from the fact that we have to handle occlusion in order to not over-count the evidence. Towards this goal, we propose an algorithm based on branch and bound, which is guaranteed to give a globally optimal solution of the joint problem. In order to compute the bounds in constant time and in order to be able to handle occlusion, we generalize the concept of integral geometry [25] to triangular shapes. Furthermore, we also develop a greedy algorithm, which performs inference efficiently.



(a) Object + Layout (b) $E_{object}(x, \mathbf{z})$ (c) $E_{layout}(x, \mathbf{y}, \mathbf{z})$

Figure 2. **Jointly inferring room layout and 3D object with occlusion reasoning:** The parameterization is indicated in (a) while the object and the layout evidence are illustrated in (b) and (c) respectively.

We demonstrate the effectiveness of our algorithms on the challenging bedroom data set [13] and show that our approach results in significant performance gains over the state-of-the-art in both 2D and 3D object detection measures. Furthermore, we are able to estimate the free-space very reliably, enabling navigation applications.

2. Related Work

Most 3D scene understanding approaches for outdoor scenes produce mainly qualitative results [10, 15, 23]. Some notable exceptions are [7, 1], which rely on short video sequences or uncalibrated image pairs. While outdoor scenarios remain fairly unexplored, estimating the 3D layout of indoor scenes has experienced increasing popularity. This is mainly attributed to the fact that indoor scenes behave mostly as 'Manhattan worlds,' simplifying the estimation problem. Most monocular approaches approximate the layout of rooms by 3D cuboids [12, 19, 31, 13, 25, 14, 26]. A notable exception is [20], which estimates the 3D layout of corridors by sweeping lines.

Early approaches to 3D layout estimation [12, 19] reduce the complexity of the problem by utilizing a set of candidates. Performance is however limited, as only a small number of hypotheses is considered. Generative models were explored in [22], and inference is performed using Markov Chain Monte Carlo sampling. Wang *et al*. [31] parameterized the problem using a Markov Random Field with only four degrees of freedom. While effective, the employed potentials are high-order involving up to four random variables. As a consequence they used a very crude discretization which limits performance. In [25], the potentials typically employed in the literature were shown to be decomposable into pairwise potentials. As a consequence denser parameterizations were used resulting in significant performance gains. More recently, Schwing and Urtasun [26] showed that the global optimum of typical layout scoring functions is obtained by employing a branch and bound approach. This resulted in provably optimal solutions that are computed in real time on a single core computer.

A wide variety of 3D object detection approaches make use of 2D appearance models from multiple viewpoints [24, 30] to obtain a weak form of 3D information [17, 29, 16, 28]. Alternatively, object centered methods utilize parametric models [8, 2, 4, 27]. Deformable part-based models [5] have also been adapted to predict 3D cuboids [9, 21, 6, 32, 13]. In this paper we make use of 2D and 3D deformable part-based models in order to estimate jointly the layout as well as the objects present in the scene.

Objects and layout were combined in [31, 19, 13], and used in [11] to predict affordances as well as to investigate the interaction between humans and objects [3]. While [31] is more concerned about predicting 'clutter' rather than actual objects, [19] proposes to augment the space of layout candidates by a set of possible objects that are chosen to be either present or absent. Since the dimensionality of the state-space (i.e., the product space of object and layout candidates) increases tremendously, a heuristic optimization with beam search is performed. In [13] the layout prediction is used to guide a 3D object detector.

Unfortunately, most approaches trade the complexity of one of the tasks (*i.e*., object and layout prediction) by proposing a small set of candidates. As a consequence, the space of hypotheses is not well explored resulting in sub-optimal solutions. In contrast, in this paper we propose a provably exact solution to the joint problem, which reasons about the exponentially many layouts as well as the exponentially many object locations and sizes. Since the complexity is at least five orders of magnitude larger than a standard layout task the problem is much more difficult to solve. The challenges are two-fold: finding an efficient parametrization that permits reasonable inference time and dealing with occlusions which arise from object-layout interactions. Towards this goal we make use of object detectors as well as geometric features and show significant improvements over state-of-the-art detectors.

3. Approach

We are interested in predicting the layout of the room as well as the objects present in the scene from monocular imagery. In this paper we mainly focus on predicting a single object, and search over all possible 3D object locations and sizes. Following existing approaches [14, 13, 19, 25],

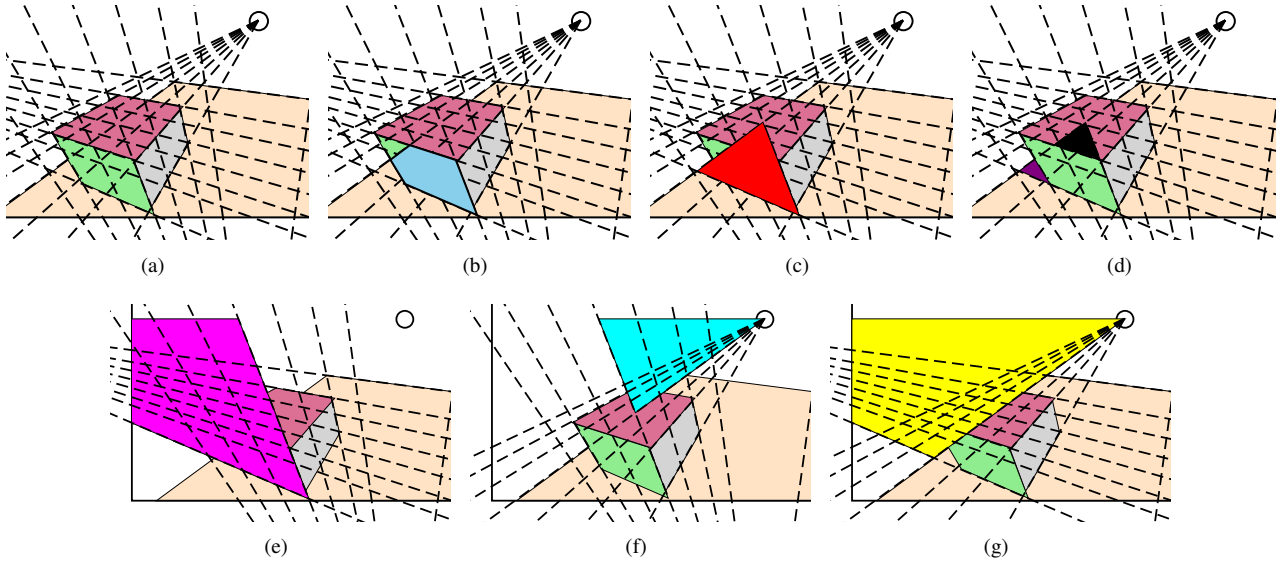


Figure 3. (a) Front face of an object is occluding the floor (blue color in (b)). Decomposition of the occluding area into a larger triangle in (c) and two triangles to be subtracted (d). Decomposition of the triangle in (c) into two positive parts (e) and (f) and a negative part (g) all depending on only two angles illustrating the generalization of integral geometry to triangular shapes, *i.e.*, (c) = (e) + (f) - (g).

we constrain the object to be aligned with the main dominant orientations. We advocate for a joint approach, as we would like to exploit the relationships that exist between the layout and object prediction tasks. The main challenges to solve are dealing with the complexity of the search space as well as handling occlusions properly. Towards this goal, we propose a branch and bound approach, which is guaranteed to find the global optimum of the energy representing the joint problem. We also develop a greedy approach, which produces accurate estimates very fast.

3.1. Joint layout-object problem

More formally, given an image x, we are interested in predicting the layout $\mathbf{y} \in \mathcal{Y}$ as well as the object $\mathbf{z} \in \mathcal{Z}$ present in the scene. As image evidence, we exploit both top-down (class-specific) features in the form of 2D and 3D object detectors, as well as bottom-up (class independent) geometric features. As geometric cues, we employ orientation maps (OM) [20] and geometric context (GC) [12], as they were shown to produce impressive results on the layout task [19, 25, 26]. Given edges detected in the image, OMs estimate a normal orientation for each pixel. Using the vanishing point configuration we can convert these normals into wall estimates, resulting in a five-dimensional feature for each pixel. GCs are six-dimensional features that utilize classifiers to predict the probability of a pixel being part of each wall as well as clutter. Additionally, we consider the 3D object detector of [6], which provides us with four values per pixel representing the likelihood of belonging to one of the four possible object faces. We also extended the deformable part-based model [5] to be supervised in terms of viewpoint, which makes up for one additional feature that represents the probability of a pixel belonging to an object. These are computed via soft-masks estimated from training data for each component.

We define the energy of a joint configuration as the sum of layout and object energies. These energies encode how well the layout and object estimates represent the image evidence. An additional term $E_{pen}(x, \mathbf{y}, \mathbf{z})$ makes sure that objects cannot penetrate walls, and an occam razor term $E_{occam}(x, \mathbf{z})$ encodes the fact that we prefer simple explanations. This is necessary in order to handle rooms that do not contain objects. We thus have

$$\begin{aligned} E_{total}(x, \mathbf{y}, \mathbf{z}) &= E_{lay-occ}(x, \mathbf{y}, \mathbf{z}) + E_{object}(x, \mathbf{z}) \\ &\quad + E_{pen}(x, \mathbf{y}, \mathbf{z}) + E_{occam}(\mathbf{z}). \end{aligned}$$

Note that the energy of the layout depends on the 3D location and size of the object. This is due to the fact that the layout should only explain the image evidence that has not yet been explained by the object, as the object occludes the layout (see Fig. 2). These occlusions make the problem computationally challenging, as the energy depends a priori on a large set of random variables.

We take advantage of the Manhattan world assumption, and let the object and the room be aligned with the three main dominant orientations. We thus first compute vanishing points (VP), and perform joint inference over the remaining degrees of freedom. Hedau *et al.* [12] and Wang *et al.* [31] showed that given the VPs only 4 degrees of freedom are necessary to represent the layout, consisting of four rays originating from two distinct vanishing points. In the case of an object, given the VPs, only 5 degrees of freedom are necessary, consisting of three rays originating from one VP and two rays from another. We refer the reader to Fig. 2(a) for an illustration of the parameterization. We thus define $\mathbf{y} \in \mathcal{Y}$ and $\mathbf{z} \in \mathcal{Z}$ to be product spaces with four

and five factors respectively. We now describe the different terms in the energy.

Object Energy: We define an additive energy which decomposes over the faces of the object, summing the evidence inside each face as illustrated in Fig. 2(b), *i.e.*,

$$E_{object}(x,\mathbf{z}) = \sum_{\gamma=1}^{4} E_{object,\gamma}(x,\mathbf{z}) = \sum_{\gamma=1}^{4} w_{obj,\gamma}^{\top}\phi_{obj,\gamma}(x,\mathbf{z}).$$

Assuming that the object is on the floor, there are only 4 possible faces γ that can be visible (*i.e.*, top, front, left, right). Furthermore, at a given time only a maximum of three faces are actually visible. We define the features for each face to be weighted counts of image cues inside that face, as this will allow us to compute bounds in constant time.

Layout Energy: The layout energy is defined as

$$E_{lay-occ}(x,\mathbf{y},\mathbf{z}) = E_{layout}(x,\mathbf{y}) - E_{occ}(x,\mathbf{y},\mathbf{z}),$$

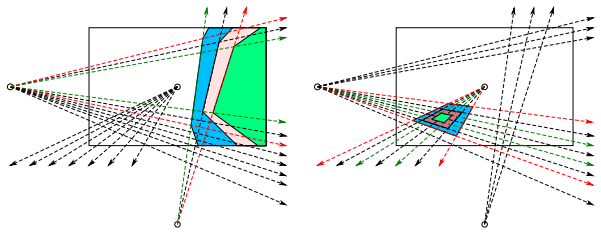
where the last term discounts the image evidence which is already explained by the object, *i.e.*, the pixels for each layout face that are occluded by the object (see Fig. 2(c)). We define features for each face α of the layout and object occlusion as weighted counts

$$\begin{aligned} E_{layout}(x,\mathbf{y}) &= \sum_{\alpha=1}^{5} w_{lay,\alpha}^{\top}\phi_{lay,\alpha}(x,\mathbf{y}), \\ E_{occ}(x,\mathbf{y},\mathbf{z}) &= \sum_{\alpha=1}^{5} w_{lay,\alpha}^{\top}\left(\sum_{\gamma=1}^{4}\phi_{occ,\alpha,\gamma}(x,\mathbf{y},\mathbf{z})\right). \end{aligned}$$

Note that we have shared the weights $w_{lay,\alpha}$ between $E_{layout}(x,\mathbf{y})$ and $E_{occ}(x,\mathbf{y},\mathbf{z})$ to properly represent occlusions. Fig. 3 provides the details for the case of α representing the floor and γ denoting the front face of the object. The area covered by blue color in Fig. 3(b) represents the floor pixels occluded by the object's front face.

Penetration Energy: This energy makes sure that the object cannot penetrate the walls defined by the layout, *i.e.*, it equals 0 whenever the object is inside the layout and is $+\infty$ in the case of penetration.

Occam razor: Given an image, we do not know a priori if there is an object in the scene. To prevent the model to always put an object, we introduce a fixed penalty for solutions that contain an object. In practice we set the penalty to be 10% of the layout energy for the best configuration.



(a) Bounding the wall (b) Bounding object top

Figure 4. We bound $E_{layout}(x,\mathbf{y})$ and $E_{object}(x,\mathbf{z})$ by computing counts over minimal and maximal faces.

3.2. Branch and Bound for exact Inference

During inference we are interested in computing the MAP estimate of the joint problem defined as

$$\min_{\mathbf{y},\mathbf{z}} E_{total}(x,\mathbf{y},\mathbf{z}).$$

Finding a global minimizer of the layout task, *i.e.*, $E_{layout}(x,\mathbf{y})$, is possible using branch and bound [26]. In this paper we generalize this approach to solve the joint layout and object problem with an explicit occlusion reasoning.

We now briefly describe the particular branch and bound algorithm we developed, which is inspired by the object detector of [18]. The algorithm operates on hypothesis sets $\mathcal{A} \subseteq \mathcal{Y} \times \mathcal{Z}$ containing a multiplicity of different object-layout configurations, and starts with a single interval being the full hypothesis set. Then, it proceeds iteratively, where the most promising set on a priority queue is taken at each iteration. If this set contains multiple hypothesis, then the set is divided into two disjoint subsets $\mathcal{A}_1$ and $\mathcal{A}_2$ (*i.e.*, $\mathcal{A}_1 \cap \mathcal{A}_2 = \emptyset$ and $\mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2$). For each one we compute a lower-bound and insert the pair of score and set into the priority queue, ordered by the quality of the bound. The algorithm terminates when the element on top of the priority queue consists of a single hypothesis. Alg. 1 depicts the branch and bound algorithm more formally. In the worst case this algorithm investigates an exponential number of hypotheses, but if the bounds are tight, typically only a small fraction needs to be considered. In order to return a global optimum, the bounds have to be valid for all the elements in the sets, and the bounds have to be exact when a single hypothesis is evaluated. The bounds developed here satisfy these two properties, and thus we retrieve the global optimum of the joint problem.

In order to utilize branch and bound, we need to parametrize sets of hypotheses, and derive bounds which are both efficient to compute and tight. We parameterize sets of hypotheses by intervals of the form $[y_{1,min}, y_{1,max}] \times \cdots \times [z_{5,min}, z_{5,max}]$, as such a parameterization simplifies our bounding functions. To keep the complexity level reasonable, we discretize the possible angles, having on average 18.4 states per layout variable and 28.1 states per object parameter. The variability in the number of states is due to the VP locations.

Algorithm 1 branch and bound (BB) inference

put $(\bar{E}(\mathcal{A}_0), \mathcal{A}_0)$ into queue and set $\mathcal{A} = \mathcal{A}_0 = \mathcal{Y} \times \mathcal{Z}$
repeat
  split $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2$ with $\mathcal{A}_1 \cap \mathcal{A}_2 = \emptyset$
  put pair $(\bar{E}(\mathcal{A}_1), \mathcal{A}_1)$ into queue
  put pair $(\bar{E}(\mathcal{A}_2), \mathcal{A}_2)$ into queue
  retrieve $\mathcal{A}$ having lowest score
until $|\mathcal{A}| = 1$

We now need to define valid bounds. As the energy is a sum of terms, we bound each one separately and compute the final bound by summing the individual ones. It is easy to see that this is a valid bound. While bounding the objects is a straightforward extension of [26], bounding the occlusion term is much more cumbersome. We do not require to bound the penetration energy as we can simply carve the space to consider only objects which are contained within the layout. As far as the occam razor potential is concerned, we equivalently add to the priority queue the best layout configuration found in the absence of any object with bound equal to its energy minus the penalty.

Layout bounds: For the layout, we utilize the lower bounds of [26], which are obtained by dividing the layout scoring function into two parts, one containing positive weights and one containing negative weights:

$$E_{layout}(x, \mathbf{y}) = w_{lay}^{+\top} \phi_{lay}^{+}(x, \mathbf{y}) + w_{lay}^{-\top} \phi_{lay}^{-}(x, \mathbf{y}),$$

where $\phi_{lay}^{+}(x, \mathbf{y})$ are $\phi_{lay}^{-}(x, \mathbf{y})$ are the concatenation of features with positive and negative weights respectively. Lower bounds are then easily estimated by summing the smallest possible face for the positive features and the biggest possible face for the negative ones. Note that we have inverted the bounds with respect to [26] as they maximize a score (defined as the negative energy) while we minimize the energy. The bound for the right layout face is illustrated in Fig. 4(a), where the leftmost ray is depicted in green and the rightmost one in red, and the maximally possible right face area is colored in blue while the minimally possible right wall is highlighted in green. Computing the content of maximal and minimal faces depends on the four intervals for the front face and on three intervals for all other walls, floor and ceiling. Using integral geometry [25] we decompose those functions into sums of accumulators that depend on at most two random variables. This allows computation of bounds in constant time while being memory efficient as well.

Object bounds: Object faces are amenable to a similar strategy. We split the energy into negative and positive components, and bound the counts using the minimally and maximally possible faces. This is illustrated for the object's top face in Fig. 4(b) with green and blue rays denoting the leftmost and rightmost rays. All object faces naïvely depend on four intervals but, as for the layout bounds, we can utilize integral geometry [25], and, by decomposing the faces into sums of pairwise accumulators, we compute the bounds in constant time being memory efficient.

		Top	Side	Hull	BB
loc	[5]	-	-	56.12	57.14
	[6]	30.61	35.71	53.06	66.33
	Sup. DPM	-	-	61.22	63.27
	Ours	**35.05**	**39.18**	**68.04**	**74.23**

Table 1. Comparison to the state-of-the-art.

Occlusion bounds: To effectively compute bounds for $E_{occ}(x, \mathbf{y}, \mathbf{z})$ we decompose the energy into sums over triangular faces. This is illustrated in Fig. 3 for the case of the front face of an object occluding the floor. The occlusion is highlight with blue color in Fig. 3(b). We decompose the occlusion region, into the sum over three triangular shapes, *i.e.*, from the red triangle in Fig. 3(c) we subtract the black and purple triangles in Fig. 3(d). More generally, a fourth positively counted triangle with its cathetus intersecting at an existing upper left corner is additionally required. While we have illustrated this decomposition with an example, all overlaps between object faces and layout walls are decomposed and computed in a similar manner. Furthermore, for each triangle, we compute the counts inside by again decomposing the computation into the sum of three accumulators. This is demonstrated in Fig. 3(e) – (g), where the pink and cyan areas are counted positively while the yellow area is subtracted, *i.e.*, (c) = (e) + (f) - (g). These accumulators are pairwise potentials, as each of the highlighted areas depends on only two angles. We then split the potentials into negative and positive and bound each with either its maximal or minimal face depending on the sign. This procedure again provides bounds computable in constant time.

3.3. Improving speed

Carving: Dividing an interval imposes new constraints that can be used to improve efficiency, *e.g.*, the ray describing the top edge of the front face is required to be above the ray describing the bottom edge of that face. To avoid those intervals we carve out spaces that are physically impossible.

Greedy approach: We derive a greedy strategy that speeds up computation by reducing the search space. To this end we first optimize E_{total} w.r.t. $\mathbf{y} \in \mathcal{Y}$ while fixing the object $\mathbf{z}$ to remain outside the image. Intuitively we explain the scene without considering objects. In a second step we fix the previously obtained layout prediction $\hat{\mathbf{y}}$ and optimize E_{total} w.r.t. $\mathbf{z} \in \mathcal{Z}$. While this is not guaranteed to yield a global optimum, our experiments show that it results

		Intersection over union								Labeling measures			
		joint				greedy				joint		greedy	
		Top	Side	Hull	BB	Top	Side	Hull	BB	9L	5L	9L	5L
loc	Geo	25.51	19.39	48.98	64.29	26.53	24.49	50.00	63.27	26.16	22.00	26.62	22.70
	Geo+2D	33.67	27.55	60.20	65.31	33.67	27.55	60.20	65.31	24.34	21.44	24.46	21.45
	Geo+3D	**37.76**	38.78	60.20	71.43	**35.71**	37.76	60.20	69.39	23.20	20.43	23.95	**21.03**
	Geo+2D+3D	35.05	**39.18**	**68.04**	**74.23**	34.69	**38.78**	**65.31**	**74.49**	**22.65**	**20.30**	**23.81**	21.22
det	Geo	36.30	32.59	51.11	54.07	36.30	34.07	49.63	51.11	27.84	23.81	26.95	23.05
	Geo+2D	42.22	38.52	62.22	66.67	43.70	40.74	62.96	65.93	25.77	22.94	24.50	21.64
	Geo+3D	**44.44**	43.70	58.52	60.74	42.96	43.70	57.78	60.00	**24.45**	**21.64**	**24.28**	**21.37**
	Geo+2D+3D	42.96	**47.41**	**66.67**	**69.63**	**45.19**	**48.89**	**65.93**	**70.37**	24.66	21.67	24.57	21.73

Table 2. Importance of the features: note that every feature we add generally improves detection. We refer to OM+GC features via *Geo*, the 2D detector via *2D*, and the 3D detector via *3D*.

			Intersection over union								Labeling measures			
			joint				greedy				joint		greedy	
			Top	Side	Hull	BB	Top	Side	Hull	BB	9L	5L	9L	5L
Sparse	loc	Oracle 9L	79.59	80.61	86.73	88.78	79.59	82.65	86.73	89.80	7.82	6.27	8.22	6.82
		Oracle 5L	79.59	79.59	85.71	88.78	76.53	78.57	79.59	85.71	7.68	6.10	10.89	7.75
	det	Oracle 9L	81.48	80.74	85.93	85.93	81.48	82.96	85.93	85.19	10.28	7.51	8.65	6.94
		Oracle 5L	80.74	80.00	84.44	85.19	79.26	80.00	80.74	82.96	11.77	6.96	11.94	7.48
Dense	loc	Oracle 9L	87.76	86.73	90.82	89.80	87.76	86.73	91.84	90.82	5.79	5.54	4.52	4.37
		Oracle 5L	82.65	87.76	88.78	89.80	75.51	80.61	81.63	82.65	5.78	4.60	9.80	6.88
	det	Oracle 9L	85.19	84.44	87.41	85.19	85.19	84.44	88.15	85.93	7.06	6.83	5.04	4.87
		Oracle 5L	83.70	88.15	87.41	88.15	76.30	80.74	80.74	80.00	8.24	5.43	10.09	6.63

Table 3. Comparison of F1 scores and labeling error for the sparse and dense parameterization using oracle features.

in performance, similar to the joint model when employing object detectors. This is expected as those make the two tasks more independent. In the case of only employing GCs and OMs, our features contain only 5 labels and are hence ambiguous (both problems are tightly coupled). Thus, the greedy approach is significantly worst in this setting.

Parameter Learning: The energy of the joint problem depends linearly on the parameters w. To learn the weights, we therefore designed a parallel cutting plane structured SVM algorithm which exploits multiple machines. Note that the loss decomposes just like the features.

4. Experimental Evaluation

We perform our experiments on the bedroom data set [13], which contains 309 labeled images. The data is split into a training and test set of size 181 and 128 respectively. We employ the vanishing point detector of [12]. We measure the performance via a pixel-wise error metric that counts the percentage of pixels that disagree with ground truth and evaluate on 9-label and 5-label metrics. Whereas the latter captures the performance on estimating orientation, irrespective of being part of the layout or the object, the 9-label metric takes into account this distinction, making it significantly harder. To evaluate the object detection performance we use the F1 measure computed for detections with intersection over union (IOU) higher than 0.5, as utilized in Pascal VOC challenge. We report this measure to detect the top face (Top), all the side faces jointly (Side), the convex hull of the object as well as a 2D bounding box (BB). Moreover, we follow [14] and also measure the free-space being a true 3D error metric.

We perform two tasks. First we look into the problem of localization where we know about the existence of an object in the scene, and the goal is to find it in 3D. For the second task, we performed 3D detection, where we have no knowledge about whether an object is present. We refer to these tasks via "loc" and "det" respectively.

Comparison to state-of-the-art: We begin our experimentation by comparing our approach to the 3D detector of [6] and the deformable part-based model [5]. We utilize the occam-razor energy only in the "det" case, since we do not know if the image contains an object. We provide the results in Tab. 1 and observe that our approach significantly improves over both baselines in all metrics.

Feature Importance: Tab. 2 shows our results when employing different types of features. We observe that each source of information, *i.e.*, geometric features (OM and GC), 2D and 3D detectors increases performance.

Oracle: To illustrate the best achievable performance of our approach, we investigate its performance when providing ground truth features. We do so in two scenarios, by using 5-label and 9-label features referred to as "Oracle 5L" and "Oracle 9L" respectively. The former mimics the case where only geometric features are provided, while the latter represents the case where one has access to perfect detec-

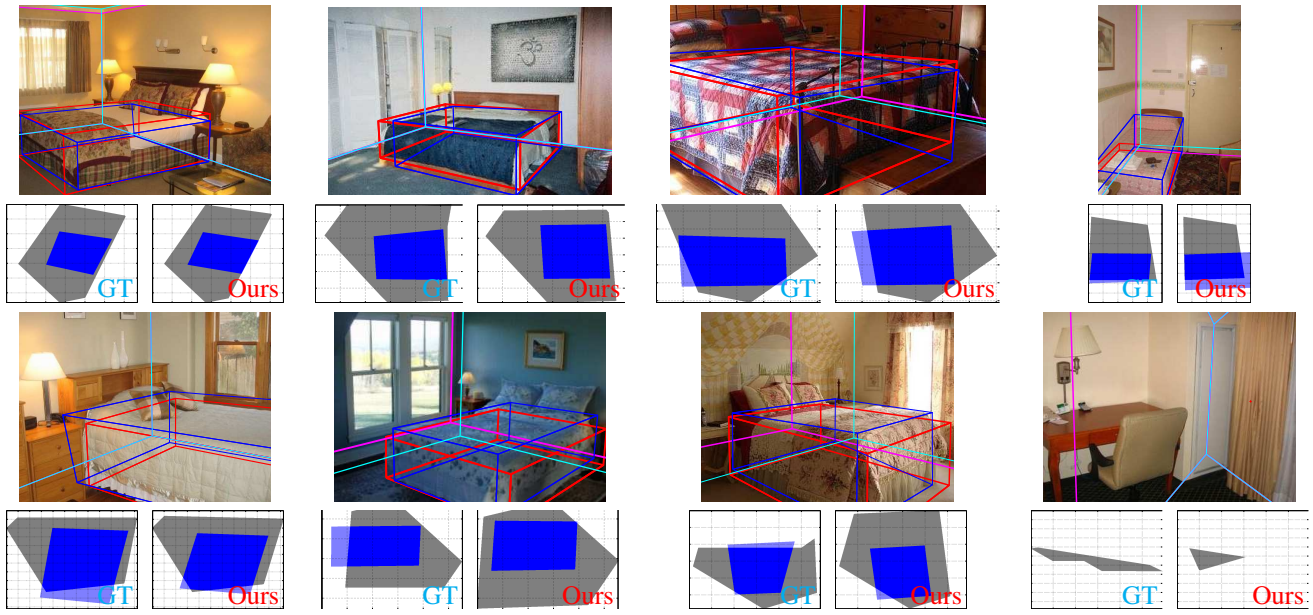


Figure 5. Illustration of prediction results (red, magenta) and best found ground truth state (blue, cyan) given vanishing points for joint object and layout inference overlaying the image. Below each image we provide visible annotation floor plan (gray) and object on the left while corresponding prediction result on the right. A failure case due to wrong vanishing points is illustrated in bottom right figure.

tions and geometry estimates. Tab. 3 shows that while the performance of the greedy approach is more or less identical when providing 9-label information, joint inference outperforms the greedy approach in the 5-label case. This is the same phenomena observed with real features.

Density of the Parameterization: The major failures of our approach are wrong vanishing points as well as discretization artifacts. To illustrate the performance gain when increasing the discretization, we almost double the average number of states per layout variable from 18.4 to 34.6 and similarly increase the state space for object parameters from 28.1 for the sparse approach to 52.9. As illustrated in Tab. 3 for oracle features, the performance increases significantly for some of the measures. On real features, however, we observe almost no gain, which is mainly due to the captured noise in the features.

Computational Complexity: Tab. 4 shows the average inference time for both the greedy and joint approach when employing different types of features. As expected, an increasing amount of features results in slower inference. The greedy approach is about two orders of magnitude faster for oracle features and three orders of magnitude faster for real features. High quality features yield faster inference.

Estimating free space: Following [11, 14] we compute the average F1 score using IOU ≥ 0.5 as well as the standard average of our estimation for the floor, the ground face of the object and the free space in Tab. 5. We observe average free-space estimation accuracies of up to 40%, improving over [14].

	joint	greedy
Oracle 9L	12.88s	0.07s
Oracle 5L	6.95s	0.07s
Geo	331.43s	0.37s
Geo+2D	230.68s	0.30s
Geo+3D	583.18s	0.43s
Geo+2D+3D	3333.09s	1.58s

Table 4. Average inference time in seconds for the joint and greedy approaches given different features

	Pascal			Average		
	Floor	Object	Free	Floor	Object	Free
Oracle 9L	89.76	62.22	77.95	77.22	62.83	64.64
Oracle 5L	90.55	60.00	77.95	78.37	60.81	64.88
Geo	63.78	29.63	35.43	57.21	35.07	40.47
Geo+2D	71.65	29.63	39.37	59.24	37.76	42.40
Geo+3D	68.50	37.78	40.94	58.36	40.95	43.33
Geo+2D+3D	70.63	37.04	38.89	58.64	41.92	42.05

Table 5. Computation of average F1 score for intersection over union of floor, object footprint and free-space for joint inference with indicated features. We provide both the mean across scores as well as PASCAL score with only counts the scores with IOU more than 0.5 as true positives.

Qualitative results: Qualitative results are illustrated in Fig. 5. In general, our approach does a great job at estimating both the layout and object. The main failures illustrated in the bottom right corner of Fig. 5 are due to VPs not being detected properly and noisy features.

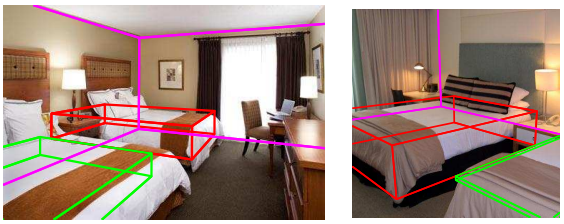


Figure 6. After jointly inferring layout (magenta) and object (red), we re-apply the object part to obtain a second object (green).

Estimating multiple objects: We extend our approach to detect multiple objects in a greedy fashion, by fixing the layout and the previously detected object and solving for the next object. Fig. 6 shows examples, with the layout, the first and second objects depicted in magenta, red and green.

5. Conclusion

We have presented an approach to joint 3D room layout and object reasoning that predicts the optimal box within a box. To this end we carefully modeled the occlusions and phrased the problem as a structured prediction task that permits exact inference via a novel branch and bound algorithm. The main technical difficulty resides in the development of occlusion bounds which require the generalization of integral geometry to triangular shapes. We plan to extend our algorithms to utilize depth sensors when RGB-D imagery is available.

References

[1] S. Bao and S. Savarese. Semantic Structure from Motion. In *Proc. CVPR*, 2011. 2

[2] R. A. Brooks. Model-based three-dimensional interpretations of two-dimensional images. *PAMI*, 1983. 2

[3] V. Delaitre, D. F. Fouhey, I. Laptev, J. Sivic, A. Gupta, and A. A. Efros. Scene semantics from long-term observation of people. In *Proc. ECCV*, 2012. 2

[4] S. J. Dickinson, A. P. Pentland, and A. Rosenfeld. 3-d shape recovery using distributed aspect matching. *PAMI*, 1992. 2

[5] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object Detection with Discriminatively Trained Part Based Models. *PAMI*, 2010. 2, 3, 5, 6

[6] S. Fidler, S. Dickinson, and R. Urtasun. 3D Object Detection and Viewpoint Estimation with a Deformable 3D Cuboid Model. In *Proc. NIPS*, 2012. 1, 2, 3, 5, 6

[7] A. Geiger, C. Wojek, and R. Urtasun. Joint 3D Estimation of Objects and Scene Layout. In *Proc. NIPS*, 2011. 2

[8] D. Glasner, M. Galun, S. Alpert, R. Basri, and G. Shakhnarovich. Viewpoint-aware object detection and pose estimation. In *Proc. ICCV*, 2011. 2

[9] C. Gu and X. Ren. Discriminative mixture-of-templates for viewpoint classification. In *Proc. ECCV*, 2010. 2

[10] A. Gupta, A. A. Efros, and M. Hebert. Blocks World Revisited: Image Understanding Using Qualitative Geometry and Mechanics. In *Proc. ECCV*, 2010. 2

[11] A. Gupta, S. Satkin, A. A. Efros, and M. Hebert. From 3D Scene Geometry to Human Workspace. In *Proc. CVPR*, 2011. 2, 7

[12] V. Hedau, D. Hoiem, and D. Forsyth. Recovering the Spatial Layout of Cluttered Rooms. In *Proc. ICCV*, 2009. 1, 2, 3, 6

[13] V. Hedau, D. Hoiem, and D. Forsyth. Thinking Inside the Box: Using Appearance Models and Context Based on Room Geometry. In *Proc. ECCV*, 2010. 1, 2, 6

[14] V. Hedau, D. Hoiem, and D. Forsyth. Recovering Free Space of Indoor Scenes from a Single Image. In *Proc. CVPR*, 2012. 1, 2, 6, 7

[15] D. Hoiem, A. A. Efros, and M. Hebert. Putting Objects in Perspective. *IJCV*, 2008. 2

[16] D. Hoiem, C. Rother, and J. Winn. 3D LayoutCRF for Multi-View Object Class Recognition and Segmentation. In *Proc. CVPR*, 2007. 2

[17] A. Kushal, C. Schmid, and J. Ponce. Flexible object models for category-level 3d object recognition. In *Proc. CVPR*, 2007. 2

[18] C. H. Lampert, M. B. Blaschko, and T. Hofmann. Efficient Subwindow Search: A Branch and Bound Framework for Object Localization. *PAMI*, 2009. 4

[19] D. C. Lee, A. Gupta, M. Hebert, and T. Kanade. Estimating Spatial Layout of Rooms using Volumetric Reasoning about Objects and Surfaces. In *Proc. NIPS*, 2010. 1, 2, 3

[20] D. C. Lee, M. Hebert, and T. Kanade. Geometric Reasoning for Single Image Structure Recovery. In *Proc. CVPR*, 2009. 2, 3

[21] B. Pepik, M. Stark, P. Gehler, and B. Schiele. Teaching 3d geometry to deformable part models. In *Proc. CVPR*, 2012. 2

[22] L. Pero, J. Bowdish, D. Fried, B. Kermgard, E. Hartley, and K. Barnard. Bayesian geometric modeling of indoor scenes. In *Proc. CVPR*, 2012. 1, 2

[23] A. Saxena, S. Chung, and A. Y. Ng. 3-D Depth Reconstruction from a Single Still Image. *IJCV*, 2008. 2

[24] H. Schneiderman and T. Kanade. A statistical method for 3d object detection applied to faces and cars. In *Proc. CVPR*, 2000. 2

[25] A. G. Schwing, T. Hazan, M. Pollefeys, and R. Urtasun. Efficient Structured Prediction for 3D Indoor Scene Understanding. In *Proc. CVPR*, 2012. 1, 2, 3, 5

[26] A. G. Schwing and R. Urtasun. Efficient Exact Inference for 3D Indoor Scene Understanding. In *Proc. ECCV*, 2012. 1, 2, 3, 4, 5

[27] M. Sun, G. Bradski, B. X. Xu, and S. Savarese. Depth-encoded hough voting for coherent object detection, pose estimation, and shape recovery. In *Proc. ECCV*, 2010. 2

[28] M. Sun, H. Su, S. Savarese, and L. Fei-Fei. A multi-view Probabilistic Model for 3D Object Classes. In *Proc. CVPR*, 2009. 2

[29] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiele, and L. van Gool. Toward multi-view Object Class Detection. In *Proc. CVPR*, 2006. 2

[30] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing visual features for multiclass and multiview object detection. *PAMI*, 2007. 2

[31] H. Wang, S. Gould, and D. Koller. Discriminative Learning with Latent Variables for Cluttered Indoor Scene Understanding. In *Proc. ECCV*, 2010. 1, 2, 3

[32] Y. Xiang and S. Savarese. Estimating the aspect layout of object categories. In *Proc. CVPR*, 2012. 2