Recovering Free Space of Indoor Scenes from a Single Image

Varsha Hedau
Nokia Research, Palo Alto, CA
varsha.hedau@nokia.com

Derek Hoiem, David Forsyth
University of Illinois at Urbana Champaign
{dhoiem,daf}@cs.illinois.edu

Abstract

In this paper we consider the problem of recovering the free space of an indoor scene from its single image. We show that exploiting the box like geometric structure of furniture and constraints provided by the scene, allows us to recover the extent of major furniture objects in 3D. Our "boxy" detector localizes box shaped objects oriented parallel to the scene across different scales and object types, and thus blocks out the occupied space in the scene. To localize the objects more accurately in 3D we introduce a set of specially designed features that capture the floor contact points of the objects. Image based metrics are not very indicative of performance in 3D. We make the first attempt to evaluate single view based occupancy estimates for 3D errors and propose several task driven performance measures towards it. On our dataset of 592 indoor images marked with full 3D geometry of the scene, we show that: (a) our detector works well using image based metrics; (b) our refinement method produces significant improvements in localization in 3D; and (c) if one evaluates using 3D metrics, our method offers major improvements over other single view based scene geometry estimation methods.

1. Introduction

The ability to estimate the scene's 3D geometry is important for many tasks, such as robot navigation and object placement/manipulation. Consider the image in Fig. 1(a). We humans intuitively interpret the 3D layout of surfaces and objects within the scene. Our spatial reasoning extends well beyond the visible surfaces. For example, we can delineate the bed and the sofa and see that the bed partially occludes the sofa and that there is vacant space between them. Providing computers with a similar level of image-based spatial understanding is one of the grand challenges of computer vision.

There is already good evidence of the potential impact of methods that can recover 3D models of a scene from one image: for example, Gupta et al. [6] show how to use these methods to identify the places where people can sit, reach and so on in a room [6]; and Karsch et al. [10] show how to exploit these methods, and others, to ensure correct lighting for computer graphics objects inserted into room scenes [10]. Recently, there has been much progress in recovering 3D models of a scene from one image [9, 19, 7, 13] and reasoning volumetrically about the objects inside [8, 12, 15, 5, 6]. In this paper, we build on these efforts and take a step further to recover the free space inside a scene. We propose an approach to parse the scene into well-localized 3D object boxes using features that account for scene perspective and object contact points. Existing work tends to evaluate based on pixel error or overlap, but our experiments show that good 2D estimates often do not translate into good 3D estimates(see Fig. 6(a)). Therefore, we propose metrics of 3D occupancy and use them for evaluation.

Background. Our work builds on the success of several single view techniques such as vanishing point estimation [16, 11, 2] and geometric structure estimation [1, 20, 7, 13, 8, 12]. Only a few works have attempted to estimate free space (e.g., obstacle-free ground area or volumes) from a single image. Motivated by the application of long-range path planning, Nabbe et al.[14] use image based appearance models to label an outdoor scene into horizontal, support, and sky surfaces. Our work differs in its focus on indoor scenes and our recovery of detailed 3D object maps. The most closely related works are those by Hedau et al. [7] and Gupta et al. [6]. Hedau et al. [7] recover clutter (object) labels in image and use a simple heuristic to get a corresponding voxel occupancy in 3D. Unlike their approach, however, our method can identify occluded portions of free space, which is useful for path planning with 3D sensors, and also can be used to infer supporting surfaces for placing objects. Gupta et al. [6] recover a volumetric representation of free space by fitting Hedau et al.'s clutter maps with cuboids, in order to predict human action affordances. Their estimates are sensitive to the initial pixel classification, and their evaluation is in pixel space. We evaluate in a 3D space and compare to their geometric estimates in our experiments.

Several recent approaches have shown the efficacy of

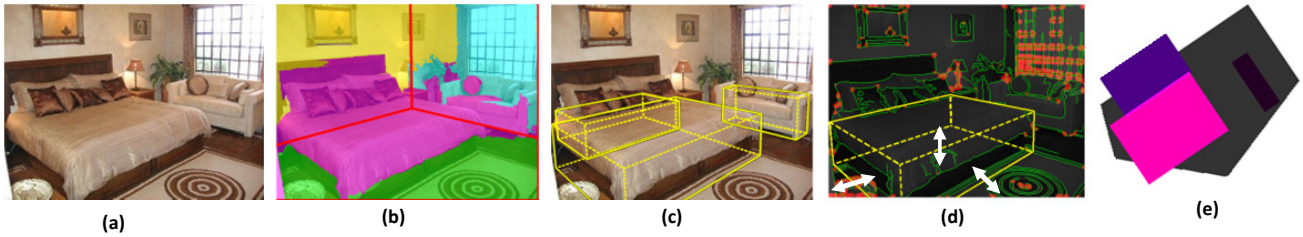


Figure 1. Our goal is to estimate free space in the scene from one image **(a)**. We define free space as the portion of the scene that is not occupied by an object or blocked by a wall. We first estimate wall and floor boundaries (shown in red lines) and clutter labels (shown in pink) **(b)** using Hedau et al.'s method [7]. We extend their 3D bed detector to a generic "boxy" object detector. Example detections of our "boxy" object detector are shown in **(c)**. The localization of detected objects is refined using features specifically designed for better 3D localization. Our features are based on local image contrast cues such as edges and corners shown with green and red **(d)**. Finally, we propose different evaluation measures in 3D. **(e)** shows the floor occupancy map corresponding to our detections. The height of the objects is shown by the hue, and the confidence is shown by intensity. Notice how modeling objects with a constrained cuboid model provides us with their extent in 3D, and the floor map clearly shows the free space between the bed and the sofa which can be used for navigation.

volumetric representations for modeling contextual cues between objects and the scene [8, 12, 5]. Gupta et al. [5] estimate a parse of outdoor images in terms of 3D blocks and reason about mechanical constraints between them. Lee et al. [12] and Hedau et al. [8] model objects as cuboids with sides that are axis-aligned with the walls and floor. We extend the cuboid detector of Hedau et al. to a more general class of boxy objects, and we investigate how to incorporate spatial constraints. Our contextual re-scoring approach is similar to others [3, 18] that encode spatial relations among objects in the image space; we extend the idea to constraints in the 3D space, with features guided by the basic rules of visibility and occupancy.

Contributions. We investigate a very challenging task of estimating 3D free space from one image. One contribution of our work is the 3D-based evaluation of our free space estimates. Standard detection methods evaluate performance in terms of the object's bounding box or pixel error in the image. However, evaluation in the image space is not always sufficient to know how accurately one can recover the object's position and extent in 3D. We have designed many specialised features, including "peg filter" to detect the floor contact points of objects which help them localize more accurately in 3D. We also offer a dataset of 592 indoor images with detailed labels of scene walls, floor and 3D object corners for different objects such as sofas, chairs, tables and beds. Our evaluation on this dataset will provide an effective baseline for future works in single-view 3D interpretation.

2. Overview

Figure 1 shows an overview of our method. To estimate the 3D scene space, we propose to detect and localize generic 3D "boxy objects" (e.g., bed, sofa, chair, and tables). We adopt the spatial layout model of [7] (Fig. 1(b)) comprised of an estimate of floor/wall/ceiling boundaries (shown in red) and a pixel labeling of the individual surfaces and objects (shown in pink). We extend the cuboid object model by Hedau et al. [8] to build a generic class of boxy objects: "boxy detector" (Fig. 1(c)). To model objects like sofas and chairs, we include an optionally present backrest that is attached to a base cuboid. We further incorporate spatial constraints between the objects (e.g., objects cannot occupy the same volume) to obtain improved cuboid estimates. With a goal of making accurate free space predictions in 3D, we also propose a local contrast based refinement step for object cuboids(Fig. 1(d)). We use local image contrast features based on image edges and corners to localize the edges of cuboids more precisely in 3D. Our method searches a small neighborhood around the detected cuboids by locally moving its boundaries in 3D and scoring them using models learned on the training data. We also develop a specialized "peg" feature that captures the thin leg structures of objects such as chairs and tables and helps in accurately localizing their floor contact points, which is necessary for reliable recovery of free space. Finally we compute such floor occupancy maps as in Fig. 1(e), which captures the over head view of the floor, each point on the floor is colored according the height of its occupancy and the intensity is proportional to the confidence. Ground truth floor pixels are gray.

3. Finding 3D Boxy Objects

Many different kinds of objects appear in rooms: tables, chairs, beds, nightstands, sofas, etc. For some purposes, we may want to know the semantic category, but, for many tasks, a geometric representation is sufficient. Most of these objects are boxy, and their edges tend to align with the orientations of the walls and floor. Given simple 3D wireframe annotations of the objects, our approach is to cluster them according to 3D aspect ratios, which roughly divides the objects into beds, sofas, chairs, and tables.

To model the appearance of boxy objects, we extend the cuboid detector of Hedau et al. [8], which was shown to work well for detecting beds. The object is represented as a 3D cuboid with horizontal and vertical faces that are aligned to room orientations. For chairs and sofas, we add a vertical plane for a backrest, with a latent variable that indicates whether the backrest is present. Object candidates are constructed by sliding a 3D cuboid along the floor. The candidates are scored using perspective-sensitive HOG (histogram of gradient) features, in which the gradients are computed along the directions of the principal vanishing points of the scene. Detected straight lines and object/clutter pixel label confidences are also used as features. Each face of the cuboid is divided into spatial cells, according to its aspect ratio (details in Sec. 5) and scored using a trained SVM classifier, following the same methodology as [8]. The overall score of a cuboid is computed as the sum of scores of each face. The backrest is considered detected if it increases the score. Cuboids of different aspect ratios are proposed and scored separately, and each high-scoring candidate is then considered a potential generic boxy object in our free space reasoning.

Context Features. Object candidates that score high independently may be geometrically implausible. Some candidates may occupy the same 3D volume, and the same edge features may provide evidence for multiple detections. Empty spaces, such between a coffee table and a sofa, for example, are often mistaken for boxy objects. Hard constraints of mutual exclusion for overlapping boxes tends to be a brittle solution. Instead, we rescore each candidate, taking into account the surrounding (and possibly overlapping) detected objects. For rescoring, we design an appropriate set of 3D context features that softly enforce constraints of size, visibility, and mutual exclusion in 3D.

As context features, we compute the ratio of scores, 3D heights, and 3D volume overlap of the highest scoring candidate close to a given cuboid. We include the maximum scores of cuboids that have containment, occlusion, and/or floor overlap spatial relations with the candidate. We also add features based on the object's height, the extent to which objects are contained within the walls, and the distance to the walls, for the three most likely room layouts, as in [8]. Finally, we add the score of a well-engineered 2D detector [4] trained on the same objects. For training we take the top 100 detections from each aspect ratio of cuboid and train a logistic regression model similar to [8]:

$$f(O_i) = 1/(1 + \exp(-w^T \phi_c(O_i, O_J, L))) \qquad (1)$$

where ϕ_c are the context features, O_J is a set of objects interacting with a given object O_i as described above, and, L is box layout of the scene. Using this context feature based model, we rank the detections. This ranking softly incorporates the geometric constraints between different detections. For example, if there is a stronger detection that violates visibility or mutual exclusion constraints as above, the score of the current detection is reduced. Similarly, the presence of high scoring candidates that almost overlap a detection boosts its score. We present the results of our generic boxy object detection in Sec. 5.

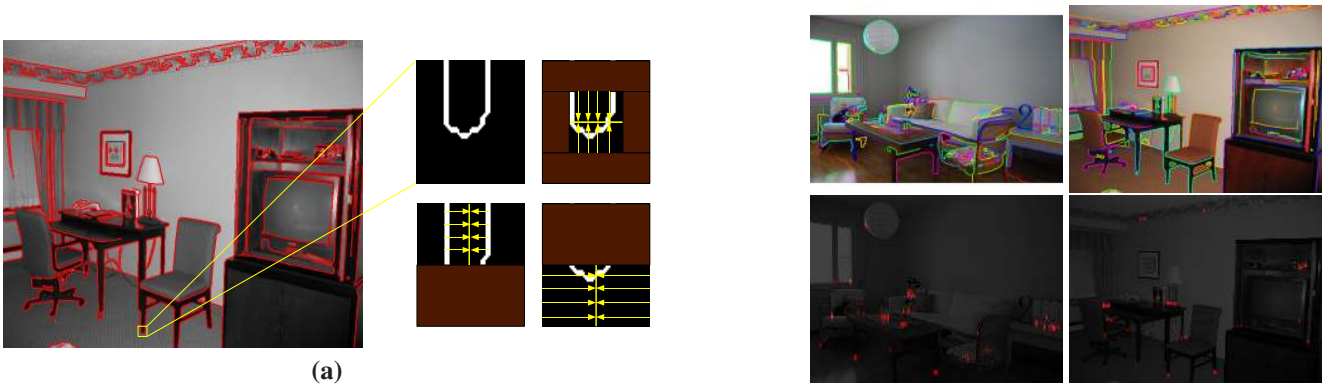
4. Refining Objects

Since the cuboid detector scores the over all gradients across the cuboid faces and due to its sliding window nature the detected cuboids need not be exactly aligned with the actual boundaries of the objects. Figure 5 shows the floor occupancy map for the given cuboid detections. For each image the occupancy map corresponding to the detection on the top is shown on the left. The footprint of the ground truth object is overlaid on the top in green. Notice the bed detection in the first image has good overlap with the true object in the image; however, the overlap of the detection with the object on the floor in 3D is actually very low. Errors measured in the image domain are not very indicative of actual errors in 3D. Small errors in 2D could result in substantial errors in 3D depending on perspective distortion and accuracy of the recovered single-view calibration. Figure 6(a) shows a scatter plot of image overlaps versus 3D floor overlaps of our detections. As seen, some detections with high image overlap have very low or even zero 3D overlap. Towards obtaining improved localization of objects in 3D, we further refine our cuboid detections using local contrast based image features.

Our refinement approach consists of moving the detected cuboid in 3D, projecting it onto the image, and scoring the local contrast evidence for this projection. We move the floor edges and height of the cuboid in 3D. We have five parameters, four corresponding to the rectangular footprint of the cuboid and one for the height of the cuboid. We search in a small local neighborhood around each corner of the cuboid at a finely sampled grid on the floor. For scoring a refined candidate, we compute its local contrast features, and learn a linear SVM classifier on the training set using ground truth cuboid markups of the objects. We next describe our local contrast features.

4.1. Local Contrast Features

We build local contrast features around the visible boundaries and corners of a given cuboid. We use two types of contrast features: edge based features determining how well the visible cuboid boundaries fit the image edges, and corner based features determining how well the cuboid corners fit the image corners. To compute edge based features, we first detect edges, followed by straight line fitting. Given a visible boundary of a cuboid, we look for edge segments which are oriented along the boundary within a thin tube around it.



(a) **(b)**

Figure 2. **(a)** Reliable detection of floor contact points of the objects is required for accurate recovery of floor occupancy. We build a specialized peg filter to detect peg-like structures in the image such as contact points of the legs of a chair or table. This filter localizes the structures with two vertical edges terminating with a horizontal edge. It is designed as a combination of three component filters. The first one counts the number of vertical edge pixels on the top-left and top-right of the filter axis projected on it. The second one subtracts the projected vertical edge pixels from left and right in the bottom part of the filter. Third component counts the edge pixels projected on the middle horizontal axis of the filter. **(b)** We refine the detected objects using local contrast features. We use straight line segments obtained by fitting lines in the linked Canny edges as shown in the first row. The second row shows the response of the "peg" filter that captures peg-like structures in image, such as contact points for legs of chairs or tables.

We use three types of corner based features. As the first feature, we compute the standard Harris cornerness measure as below:

$$harr = (I_x^2 * I_y^2 - (I_x I_y)^2)/(I_x^2 + I_y^2) \tag{2}$$

where I_x, I_y are image gradients in x and y directions, respectively. The second corner feature we use is the difference of color histograms on the two sides of the corner, the object and the background sides. This feature is helpful to delineate the object if it has a color distinct from the background. In addition to the above standard cornerness measures, we also develop a specialized corner filter called "peg." To obtain accurate localization of objects in 3D and to estimate free space on the floor, it is important to accurately recover the floor contact points of the object. For many sofas, chairs, and tables, no edge boundary based evidence exists for the floor contact points, except for thin leg-like structures that we term as "pegs." We thus develop a specialized peg detector, depicted in Fig. 2. A peg is characterized by a vertical edge on its top right and top left portions, a horizontal edge at the center, and no edges below it. To compute the peg response at a point, we take the edge response around it. The top right vertical response f_{tr} is computed as the fraction of the vertical axis above the center point, which has a vertical edge to its right. The left vertical response (f_{tl}) is computed similarly. The horizontal edge response f_h is computed as the fraction of the horizontal axis which has an edge to its top or bottom within a window. Similarly, the bottom edge response f_b is computed as the fraction of the vertical axis below the point that has a vertical edge to its right or left. The final peg response is computed as

$$f_{peg} = min(f_{tr}, f_{tl}) \cdot (1 - f_b) \cdot f_h \tag{3}$$

Example detections of the peg filter are shown in Fig. 2 . We encode the maximum score of the peg filter and Harris score in a small window around the floor corners of the cuboid. We also compute the maximum peg response outside the cuboid to capture the absence of pegs below the object cuboid boundary.

In addition to edge and corner features explained above, we also use surface label based features [7]. We use the gradient of the floor and object confidence in tubes around the floor and vertical boundaries of the cuboid.

5. Experiments

To our knowledge there are no current evaluation measures and benchmarks for 3D inference of scene from single view. Most methods evaluate performance in terms of 2D pixel errors which is not very indicative of actual 3D errors. The goal of our experiments is to explicitly evaluate accuracy of our free space estimation algorithm in 3D, we propose different 3D based performance measures. We evaluate our free space estimates in terms of: a) standard 2D precision-recall, b) 3D floor overlap with the ground truth, c) precision-recall over occupied 3D voxels, and d) precision-recall based on distance to closest occupied point on the floor.

We created a dataset of 592 indoor image by appending the dataset of Hedau et al. from [7, 8] with additionally collected images from Flickr and LabelMe [17]. For these images we have labeled ground truth corners of walls, floor

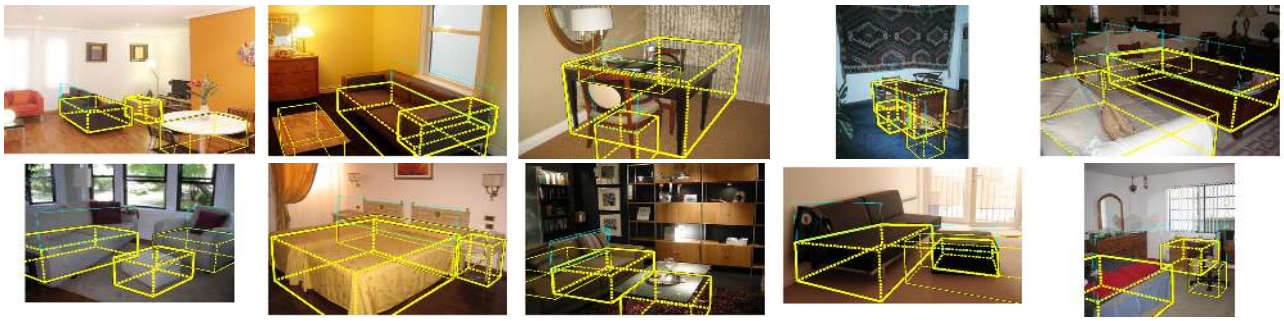


Figure 3. Generic boxy object detections in the indoor images. First three detections are shown for each image. Thickness of the boundary shows the rank of the detection.

and different objects, e.g., sofas, chairs, tables, and dressers. Bed markups are obtained from dataset of [8]. We split this set randomly into 348 training and 244 test images.

Detecting Boxy Objects. We define the aspect ratio for a boxy object by a 4-tuple comprising the ratio of the height, width, and length of its base cuboid and the height of its backrest: $dims = (n_h, n_w, n_l, n_{hres})$. The width of the headrest is the same as that of the base cuboid, and its length is negligible; hence, they are not used. We use an aspect ratio of $dims = (3, 5, 7, 3)$ for beds, $dims = (3, 4, 7, 3)$ for sofas, $dims = (3, 3, 3, 3)$ for chairs, and $dims = (3, 3, 3, 0)$ for tables. For each object, we generate candidates with several different dimensions.

Figure 3 shows the first three detections of our boxy object detector. The boxy detector can detect objects across different size and aspect ratios. Most of the detections occur on objects present in the scene. The false positives are located at high oriented gradient portions of image, e.g., the carpet in the first row, second image. Optionally present backrest (shown in Cyan) helps in localising cuboids by explaining the strong horizontal gradients in the image. Since the backrest is assumed to have negligible width, our backrest estimates does not affect free-space accuracy. Precision-recall curves for boxy object detection are shown in Fig. 4. These are computed for bounding boxes of the object cuboids. In addition to a standard non-maximum suppression, which removes overlapping detections in image in a greedy manner, we also apply a 3D based non-maximum suppression. A detection is added if at least 25% of its 3D volume is unexplained by the previous detections. The average precision (AP) of taking the top 100 detections for each bed, sofa, chair, and table cuboid detector and ranking them according to the cuboid detector score is 0.30. Adding the score of the 2D detector [4] to our cuboid detector results in an increased AP of 0.38. Rescoring based on the context features in addition to the score from 2D detector results in an AP of 0.39.

Improvements due to local refinement. For refinement, we train separate linear SVM classifiers for beds, sofas, chairs, and tables using local constrast features. As positive examples for training, we use the ground truth cuboid markups for the respective class. As negative examples, we sample neighboring cuboids which have less than 50% convex hull overlap with any ground truth. We expect the trained models to reflect the characteristics of the object class, e.g., beds have high contrast at floor contact boundaries, chairs and tables have pegs, and sofas may have high contrast floor boundaries or pegs. During testing, the class of a cuboid is determined by its aspect ratio, and the corresponding classifier is used to refine it. Figure 5 shows qualitative results of our local refinment algorithm. For many examples, high contrast floor edges and high scoring pegs result in improved floor overlap. The last two images, show some failure cases. In the presence of clutter, cuboid edges can latch on the neighboring objects, which results in poor overlap with the original object.

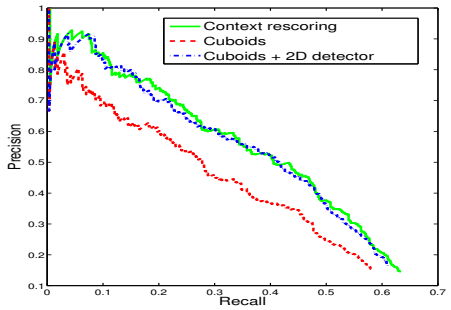


Figure 4. Precision-recall for boxy object detector, computed as bounding box overlap of object cuboids. Red curve shows the performance for concatenating the top few detections from each bed, sofa, chair, and table detectors. Green curve shows the performance for rescoring the cuboid detections using the context features. Blue curve shows the performance for scoring the cuboids by adding the score of the 2D detector from [4].

Free space - floor maps: Table. 1 shows average floor overlap per detection before and after local contrast based reasoning. Overlap is computed as the intersection divided

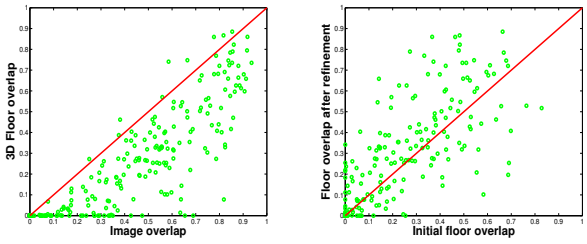
Figure 5. Local contrast based cuboid refinement. For each image we show the initial cuboid detection (first row), refined detection (second row), and corresponding floor occupancy maps (third row). The initial floor map is shown on the left and the refined one is shown on the right. The ground truth object footprint is overlaid on the top in green and the floor is shown in gray. Notice how the presence of strong floor edges help improve the floor overlap for the bed. Similarly, peg features help fix the erroneous footprint of the sofa, chair and table. Reasoning based on local features can sometimes result in wrongly fixating cuboid boundaries of an object on the other strong edges of the neighboring objects or the object itself as in the last two images.

the union of the rectangular footprint of the detection with the closest ground truth object. Average floor overlap over all detections improves by about 1%. Local refinement results in significant improvement for fully visible objects, that are not marked as occluded or cropped in the ground truth annotations. For occluded and cropped objects, the local refinement can lead to erroneous footprints, as shown in the last image of Fig. 5. In Fig. 6(b) we show the scatter plot of the floor overlap of the detections, corresponding to fully visible objects, before and after the local refinement. For most detections, the overlap improves after refinement, while for some cases confusion caused by local clutter can decrease the overlap. We next describe our evaluation for free space.

Average overlap	All	visible	visible+good
Before	19.13%	21.57%	37.47%
After	20.27%	25.98%	41.12%

Table 1. Average floor overlap of a detection before and after local contrast based refinement. First column shows average overlap over all the detections; second column shows average floor overlap for only fully visible ground truths, i.e., objects that are not marked as cropped or occluded. Third column is average floor overlap for non-hard ground truths and good detections, which have good initial floor overlap with the ground truths. Average floor overlap improves after refinement over all detections; floor overlap for unoccluded uncropped objects improves significantly.

Free space - voxel precision-recall: Along the same lines as the 2D image measure, we compute precision recall measure over voxels in 3D. Assuming a fixed camera



(a) **(b)**

Figure 6. (a) Relationship between image overlap and 3D floor overlap of the detected cuboids with the closest ground truth. Some detections with high 2D overlap have very low or zero 3D overlap. However, high 3D overlap always implies good 2D overlap. (b) Floor overlap of the detections before and after local refinement. Local refinement improves overlap in most of the cases and in some cases it does make the overlap worse.

height and using the camera parameters estimated from vanishing point locations, we compute the 3D voxels occupied by the detected cuboids and ground truth markups. We use a voxel grid with 0.25 ft resolution. The voxels are assigned a confidence value equal to that of the maximum scoring detection containing it. Figure 7(a) shows the precision-recall curve computed by varying the detection threshold. Our local feature based refinement step clearly improves the 3D voxel estimates.

Free space - distance to first obstacle: For some tasks such as navigation, the height of the occupied area above the ground is not important. Distance to the closest occupied points on the floor are sufficient to determine open

space on the floor.We propose distance based measures for floor occupancy that capture the detection characteristics similar to precision-recall measures (Fig. 7(b)). First, a point on the floor in 3D is assigned a confidence equal to the maximum scoring detection that contains it. Detected floor points greater than a confidence threshold are chosen. A measure δ_{prec} is computed as the distance of the closest ground truth point for each detection, averaged over all detections(Eq. 4). This is similar to $precision$, since it measures how close are the detections to ground truths. Similarly, a measure δ_{rec} is computed as the distance of closest detection for each ground truth point, averaged over groundtruths. This is similar to $recall$, since it measures how well are the groundtruth points detected.

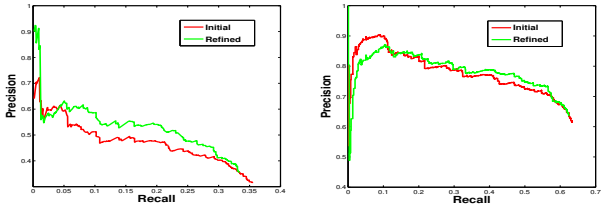
$$\begin{aligned} \delta_{prec} &= 1 - \frac{\sum_{j=1}^{m} \min_{1 \le i \le n} \delta(g_i, d_j)}{m} \\ \delta_{rec} &= 1 - \frac{\sum_{i=1}^{n} \min_{1 \le j \le m} \delta(g_i, d_j)}{n} \end{aligned} \quad (4)$$

Here m, n are number of detected floor points and number of ground truth occupied points, $\delta(g_i, d_j)$ is the normalised distance between i^{th} ground truth point and j^{th} detected point computed as $1 - e^{-||g_i - d_j||/2\gamma}$. We use $\gamma = 1$ foot. Note however that $(\delta_{prec}, \delta_{rec})$ pair measures the precision-recall characteristics in a soft manner, since individual detections are not assigned binary values of correct or incorrect detections. In Fig. 7(b), we plot $1-\delta_{rec}$ versus $1-\delta_{prec}$. We show some qualitative results for our free space estimation in Fig. 8.

In Table. 2 we show comparison of our free space estimates with the object geometry estimates of Gupta et al. [6]. We compute the 3D voxel and distance based precision-recall for both the methods on their testset. Since their method does not give us a rank list of objects in the scene we can compute only one value of precision-recall. We compare our precision value to their precision value at the same recall rate.

Precision (at recall)	Floor occupancy	3D voxels
Gupta et al. [6]	0.48 (0.48)	0.08 (0.25)
Ours	0.74 (0.48)	0.49 (0.25)

Table 2. Our free space estimates are more accurate both in terms of predicting floor occupancy and 3D voxel occupancy in the rooms as compared to Gupta et al. Our method is more robust since we score the cuboids using the overall gradients and object label confidence across cuboid's faces compared to their method of greedily fitting cuboids in object clutter labels of [6]. We additionally gain in performance by localising our cuboids more precisely using specially designed features.



(a) **(b)**

Figure 7. (a) Precision-recall for 3D voxel occupancy. (b) Precision-recall curve for floor occupancy using distances on floor (see text for details). The curves for original cuboids are shown in red and those for refined cuboids are shown in green.

6. Conclusion

We have proposed an approach to obtain free space inside the scene from its single image. Our method localizes object boxes and estimates horizontal and vertical surfaces of objects in 3D. The key to our approach is parsing the free space in terms of constrained boxy geometries which are recovered robustly using global perspective features. These provide good starting point for more detailed location refining using local image cues at a later stage. We have proposed 3D based performance measures to evaluate the estimated free space qualitatively and quantitatively. Our free space outputs can be used for applications such as robot navigation, inserting new objects, or animation in the scene. Future extensions include building improved models of 3D interactions between objects, possibly including object types. Our notion of free space can also be extended to objects supported by walls and the ceiling.

Acknowledgements This work was supported in part by NSF Award 09-16014.

References

[1] O. Barinova, V. Konushin, A. Yakubenko, K. Lee, H. Lim, and A. Konushin. Fast automatic single-view 3-d reconstruction of urban scenes. In *Proceedings of ECCV*, pages 100–113, 2008.

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

[3] C. Desai, D. Ramanan, C. Fowlkes, and U. C. Irvine. Discriminative models for multi-class object layout. In *Proceedings of ICCV*, pages 229–236, 2009.

[4] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 32(9):1627–1645, 2009.

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

[6] A. Gupta, S. Satkin, A. A. Efros, and M. Hebert. From 3-D scene geometry to human workspace. In *Proceedings of CVPR*, pages 1961–1968, 2011.

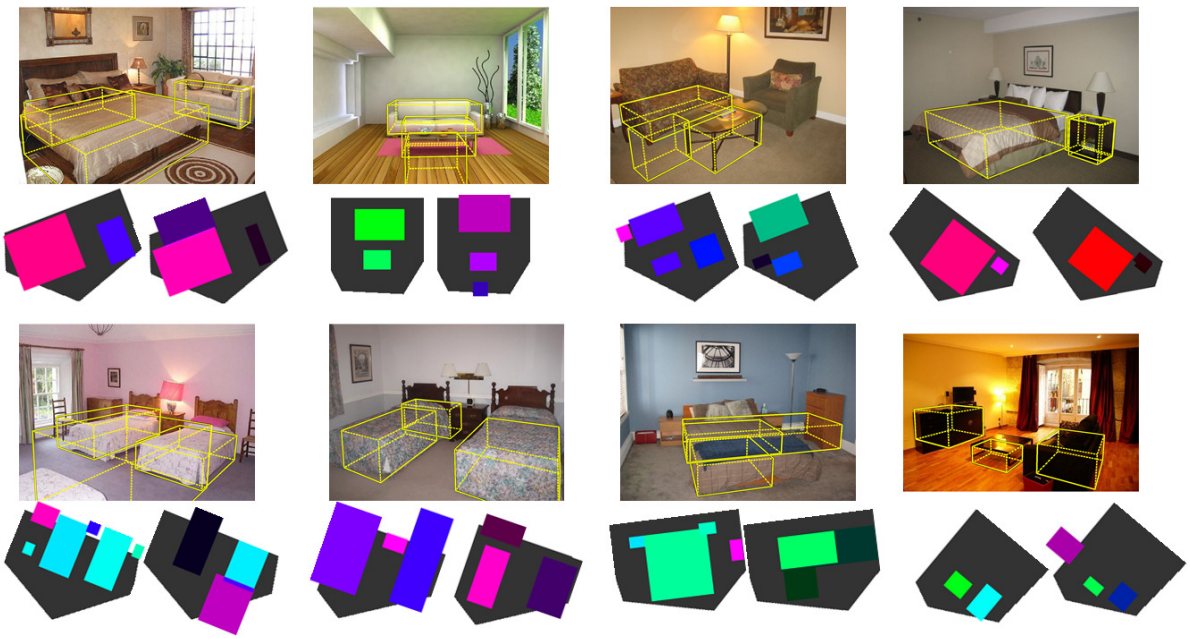


Figure 8. Free space. For each image we show top three detections (first row) and corresponding floor occupancy maps (second row). The ground truth occupancy map is shown on the left and occupancy map corresponsing to our detections is shown on the right. Color of the occupancy map shows the relative height of the detections and intensity shows their confidence. Groundtruth floor is shown in gray.

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

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

[9] D. Hoiem, A. Efros, and M. Hebert. Recovering surface layout from an image. *IJCV*, 75(1):151–172, 2007.

[10] K. Karsch, V. Hedau, D. Forsyth, and D. Hoiem. Rendering synthetic objects into legacy photographs. In *Proceedings of ACM Siggraph Asia*, volume 30, 2011.

[11] J. Kosecka and W. Zhang. Video compass. In *Proceedings of ECCV*. Springer-Verlag, 2002.

[12] D. Lee, A. Gupta, M. Hebert, and T. Kanade. Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. In *Proceedings of NIPS*, volume 24, pages 1288–1296, 2010.

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

[14] B. Nabbe, D. Hoiem, A. Efros, and M. Hebert. Opportunistic use of vision to push back the path-planning horizon. In *Proceedings of IROS*, pages 2388 – 2393, 2006.

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

[16] C. Rother. A new approach to vanishing point detection in architectural environments. *IVC*, 20(10):647–655, 2002.

[17] B. Russell, A. Torralba, K. Murphy, and W. Freeman. LabelMe: A database and web-based tool for image annotation. *IJCV*, 77(1):157–173, 2008.

[18] M. A. Sadeghi and A. Farhadi. Recognition using visual phrases. In *Proceedings of CVPR*, pages 1745–1752, 2011.

[19] A. Saxena, S. Chung, and A. Y. Ng. Learning depth from single monocular images. In *Proceedings of NIPS*, pages 1161–1168, 2006.

[20] S. Yu, H. Zhang, and J. Malik. Inferring spatial layout from a single image via depth-ordered grouping. In *IEEE Workshop on POCV*, pages 1–7, 2008.