Pano2CAD: Room Layout From A Single Panorama Image

Jiu Xu[1] Björn Stenger[1] Tommi Kerola* Tony Tung[2]*
[1] Rakuten Institute of Technology [2] Facebook

Abstract

This paper presents a method of estimating the geometry of a room and the 3D pose of objects from a single 360° panorama image. Assuming Manhattan World geometry, we formulate the task as an inference problem in which we estimate positions and orientations of walls and objects. The method combines surface normal estimation, 2D object detection and 3D object pose estimation. Quantitative results are presented on a dataset of synthetically generated 3D rooms containing objects, as well as on a subset of hand-labeled images from the public SUN360 dataset.

1. Introduction

3D scene understanding from images has been an active research topic in computer vision, enabling applications in navigation, interaction, and robotics. State-of-the-art techniques allow layout estimation from a single image of an indoor scene [5, 26, 30], which is an underconstrained problem. Most prior work estimates the layout of a room corner only or assumes a simple box-shaped geometry. Since a standard camera lens has a limited field of view, an incremental procedure is usually necessary to recover a whole scene [2]. A simple alternative is to capture panorama images, assuming that objects of interest are visible. For example, the *PanoContext* method [39] recovers the full room layout from one panorama image, while still assuming a box-shaped room and box-shaped objects. Walls and floor are used as context information to recognize object categories and positions.

In this paper, we build on insights from the *PanoContext* work, but no longer assume a box model for the scene and objects. In contrast to the bottom-up object proposals from edges [39], we employ more robust top-down methods for object detection and 3D pose estimation. To accomplish this, we first transform the single panorama image into a set of perspective images from which we estimate per-pixel surface orientations and object detections. From these we obtain a first scene layout up to an unknown scale. Next, objects are detected using a trained detector and initial 3D poses are estimated using a libary of 3D models. Global scale is estimated indirectly by projecting 3D object models of known dimensions into the scene. We sample room hypotheses and evaluate their posterior probability. See Fig. 1 for an example output of our algorithm.

Figure 1: **Example output:** *Indoor scene reconstruction from a single panorama image. Top: input image with 2D object detection bounding boxes. Detection is carried out in perspective images and the bounding box coordinates are projected into the panorama image. Bottom left: estimated surface orientations. Bottom right: reconstructed 3D room geometry and furniture items (top view).*

The contributions of this work are: (1) We relax the box-shape assumption of [39] to a Manhattan World assumption, reconstructing the complete shape of the room. (2) Object location and pose is estimated using top-down object detection and 3D pose estimation using a public library of 3D models, (3) We introduce a context prior for object and wall relationships in order to sample plausible room hypotheses. We evaluate the accuracy of the method on synthetically generated data of 3D rooms as well as results on images from the public SUN360 dataset. Please also see the supplementary video for qualitative results.

*The work was done while the authors were with Rakuten.

2. Related work

We put our work into context by discussing prior work in the areas of surface estimation from images, 3D object models, and context priors.

Geometry estimation. Early seminal work in layout estimation includes surface estimation from an image [16] by learning a mapping from an input image to a coarse geometric description. Similarly, the *Make3D* method estimates a 3D planar patch model of the image, with images and depth maps as training data [27]. More recently much progress has been made estimating pixel-wise normals from images [8, 11]. For indoor scenes, assuming a Manhattan World geometry, vanishing points can be detected and the camera parameters recovered. For example, Lee *et al*. [20] proposed a method to interpret a set of line segments to recover 3D indoor structure, demonstrating that the full image appearance is not necessary to solve this problem. Hedau *et al*. [13] modeled the whole room as a 3D box and learned to classify walls, floor, ceiling, and other objects in a room. Work by Schwing *et al*. [29, 30] estimates a 3D box-shaped room from a single image using integral geometry for efficiently evaluating 3D hypotheses. The work by Wang *et al*. [35] has shown improved accuracy by estimating cluttered areas, including all objects except the room boundaries. In this paper we estimate surface orientations of the whole scene [13, 20] and treat the orientations in object regions separately. Other approaches include Cabral and Furukawa [2], who use multiple input images to apply 3D reconstruction and estimate a piece-wise planar 3D model. Building on the work by Ramalingam and Brand [24], recent work by Yang and Zhang [38] recovers 3D shape from lines and superpixels in a constraint graph. However, it does not supply scene semantics or a structured scene representation. Complementary to these two methods we estimate the 3D room geometry together with 3D objects.

3D Objects. Objects contained within a room have been modeled at different levels of complexity. For example, Lee *et al*. [19] fit 3D cuboid models to image data, demonstrating that including volumetric reasoning improves the estimation of the room geometry. Hedau *et al*. [14, 15] showed that the scene around an object is useful for building good detectors, however it was also limited to cuboid objects. Del Pero *et al*. [5] proposed part-based 3D object models, allowing more accurate modeling of fine structures, such as table legs. Configurations of their detailed models are searched using MCMC sampling. In their 'Box in the Box' paper, Schwing *et al*. [28] used a branch-and-bound method to jointly infer 3D room layout and objects aligned with the dominant orientations. Satkin *et al*. [26] proposed a top-down matching approach to align 3D models from a database with an image. The method employs multiple cues to match 3D models to images. In recent work by Su *et al*., a CNN was trained for pose estimation for 12 object categories (from the PASCAL 3D+ dataset) from rendered 3D models [32, 36]. Tulsiani *et al*. [34] combine object localization and reconstruction from a single image using CNNs for detection and segmentation, and view point estimation. This top-down information is fused with shading cues from the image. While these are viable approaches, the number of categories is limited and our object shapes of interest are typically not represented exactly. We therefore estimate 3D object models from a model database, similar to the recent work in [17].

Context priors. Pieces of furniture tend not to be uniformly distributed within a room, but follow certain rules that include physical constraints, such as non-intersection, or less rigid functional constraints, such as aligning a bed with one of the walls or leaving some space to access all areas of the room. Such prior knowledge has been employed to improve layout estimation. For example, Del Pero *et al*. [4, 5] introduced constraints to avoid object overlap and to explicitly search for objects that frequently co-occur, such as tables and chairs. In *PanoContext*, Zhang *et al*. [39] show that context evidence of an entire room can be captured from panoramic images. They learn pairwise object displacements to score their bottom-up object hypotheses. However, their box-shaped room model does not take relative orientation or distance to walls into account. Some insight can be gained from the graphics literature where generative models have been used for 3D model search. For example, Fisher and Hanrahan proposed a method for efficient search of 3D scenes [9]. Pairwise relationships were learned from 3D Warehouse scene graphs, but only relative distances, not orientations, were taken into account. Merrel *et al*. [22] proposed a density function for room layout design that encodes numerous design rules, such as respecting clearance distance around objects and the relative alignment of objects with each other. Handa *et al*. [10] used geometric constraints to automatically generate 3D indoor scenes as training data for semantic labeling. The method proposed here scores room layout hypotheses, in a 2D top-down view, with pairwise energy terms, encoding object-to-object and object-to-wall constraints, but allows for more flexibility compared to the generative model in [22].

3. Generative model

Given an indoor scene $S = (W, O)$, defined by a set of walls $W = \{w_i\}_{i=1}^{N_w}$ and a set of objects $O = \{o_j\}_{j=1}^{N_o}$, we formulate the room layout estimation in a Bayesian framework, where the model parameters consist of

$$\Phi = (c, \lambda, p_i^w, \theta_i^w, p_j^o, \theta_j^o), \tag{1}$$

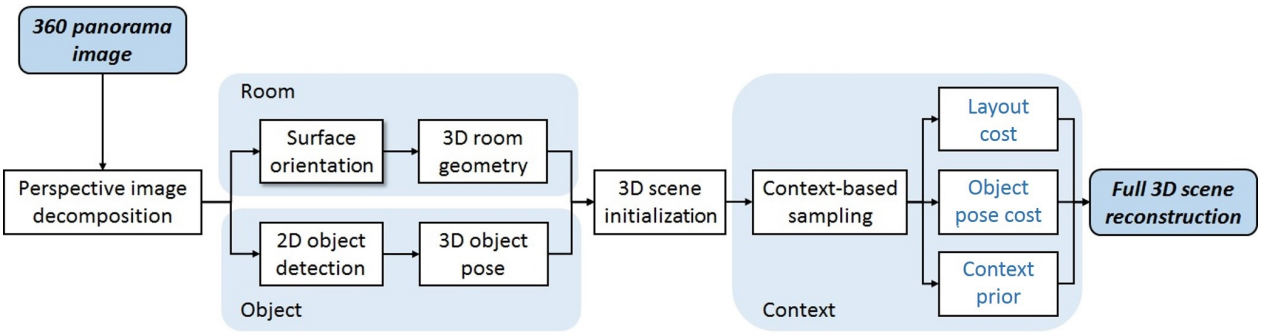


Figure 2: **Algorithm overview.** *From a single panorama image the proposed method estimates initial 3D room geometry and 3D object poses. Subsequently we sample a global posterior distribution which includes terms for room layout, object poses, and a context prior.*

which includes the camera model c, the absolute room scale λ, wall center positions and wall orientations $\{p_i^w, \theta_i^w\}$, as well as center positions $\{p_j^o\}$ and the orientations $\{\theta_j^o\}$ of objects $\{o_j\}$ in the scene S. We formulate the estimation task as maximizing the probability $P(\Phi|\mathcal{I})$ of the model parameters Φ given an input image $\mathcal{I}$ of the scene S. This is equivalent to maximizing the posterior $P(\mathcal{I}|\Phi)\pi(\Phi)/P(\mathcal{I})$ to obtain the set of optimal parameters:

$$\Phi_{\text{MAP}} = \underset{\Phi}{\text{argmax}}\ P(\mathcal{I}|\Phi)\,\pi(\Phi)\,, \tag{2}$$

where $\pi(\Phi)$ is the prior on the model parameters, and $P(\mathcal{I})$ is assumed uniform. In what follows, we give details on the different components of the model and the estimation process. The overall approach is summarized in Fig. 2.

3.1. Room layout likelihood

We decompose the likelihood in Eq. 2 as follows, making a conditional independence assumption:

$$P(\mathcal{I}|\Phi) = P(\mathcal{I}|\lambda, p_j^o, p_i^w, \theta_i^w)\,P(\mathcal{I}|\theta_j^o), \tag{3}$$

obtaining two likelihood terms, one for the scene including object positions, and one for object orientations. Both terms are evaluated by comparing the projected image $\mathcal{D}$ of the predicted 3D scene model (*i.e.*, obtained with estimated model parameters), with the observed orientation image. Our justification for decomposing the likelihood is that we synthesize only the walls of the scene, not the (occluding) objects. Object bounding boxes, which are not always accurate, serve as masks. Therefore we do not account for object orientation when evaluating the first term. Note that like in previous work the camera parameters c are approximated by placing it at the center of the spherical image at a known height, *e.g.* 1.70m for the SUN360 dataset [39].

In the following we describe the processing steps to evaluate the first likelihood term. We first transform the spherical panorama input image $\mathcal{I}$ into a set of K perspective images $\{\mathcal{I}_k\}_{k=1}^K$, which no longer contain strong distortions.

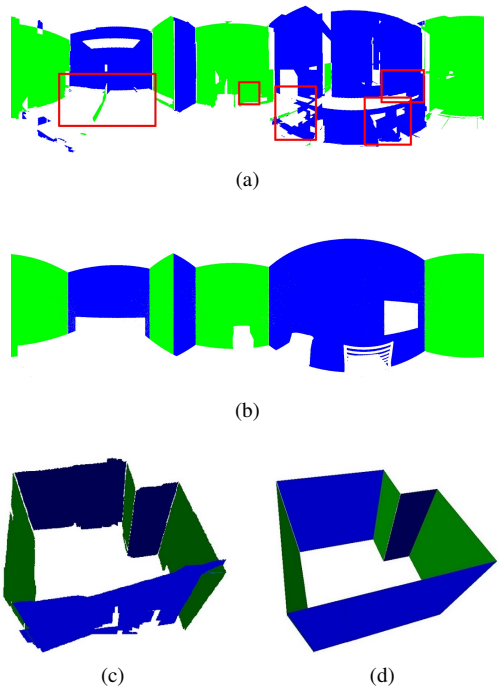


Figure 3: **Processing steps:** *(a) Surface orientations of the observed input image with detected object regions used for masking. (b) Surface orientations of rendered image of predicted models with occluding objects (*i.e.*, silhouettes serve as masks). (c) Room geometry after surface alignment at unknown scale. (d) Room geometry after plane fitting.*

This transformation returns a set of perspective images with overlapping regions, in our case 6 images with a 90° field of view and 30° of overlap between adjacent images. For each image $\mathcal{I}_k$ we estimate surface orientations at each pixel by combining estimates of their Orientation Map (OM) [20] and Geometric Context (GC) [13]. We apply GC to the panorama and combine the OM and GC in the floor region

to obtain wall positions and orientations [39]. We segment the panorama image into regions of three orthogonal surface normal directions (see Fig. 1, bottom left). The orientation surface image for each $\mathcal{I}_k$ is used to recover partial 3D room geometry. The surface normals in each perspective image are converted to 3D points using vanishing points and camera-to-floor distance using the method in [6]. Each image corresponds to a separate 3D point cloud with unknown scale. We apply the constraint that corresponding pixels in overlapping regions in the images have the same depth: We globally align the point clounds by minimizing the sum of 3D point distances of points corresponding to the overlapping image regions. This is followed by greedy plane fitting, starting from the largest segment, using Iterative Closest Points (ICP) [1], resulting in an initial estimate of 3D room geometry, *i.e*. positions and orientations of walls $\{\hat{w}_i\}$, up to scale, see Fig. 3.

We also run object detection in each image $\{\mathcal{I}_k\}_{k=1}^K$ using a *Faster R-CNN* (details in Sect. 4). The coordinates of object locations are reprojected to the panorama image $\mathcal{I}$, and non-maximum suppression is applied to eliminate redundant detections. Since the camera c is oriented toward the center of $\mathcal{I}$, assuming the center position at 0°, the positions $\{p_j\}$ of detected objects $\{o_j\}$ can be derived from the polar coordinates. Absolute distances of objects to the camera remain unknown at this stage. Having estimated a 3D scene model including walls and object positions, we define the likelihood term for the room layout as

$$P(\mathcal{I}|\lambda, \{p_j^o\}, \{p_i^w, \theta_i^w\}) \propto \exp\left[-E_s(\mathcal{I}, \{p_j^o\}, \{p_i^w, \theta_i^w\})\right]. \quad (4)$$

The cost function E_s evaluates a room hypothesis by reprojecting the synthesized 3D scene back into the panoramic view and compare surface normals:

$$E_s(\mathcal{I}, \{p_j^o\}, \{p_i^w, \theta_i^w\}) = 1 - \frac{N_{\mathrm{c}}}{N_{\mathrm{pix}}}, \quad \text{where} \quad (5)$$

$$N_{\mathrm{c}} = \sum_{m \in \mathcal{I}} \mathbb{1}_{l(\mathcal{I}_m)=l(\mathcal{D}_m)}(m). \quad (6)$$

The cost is low when surface orientations of the predicted 3D scene agree with the orientation image $\mathcal{I}$. The terms $l(\mathcal{I}_m)$ and $l(\mathcal{D}_m)$ are the discrete surface orientation labels at pixel location m in the surface orientation maps of images $\mathcal{I}$ and $\mathcal{D}$, respectively, $\mathbb{1}$ is the indicator function, and N_{pix} is the number of pixels in $\mathcal{I}$. In previous work [4, 20], a similar term is used to evaluate wall geometry hypotheses. However, the presence of objects and occluded walls in the scene, which add noise to the estimation, were not considered. Here, we propose to mask detected objects in the surface orientation images as follows: In the observed image, bounding boxes of detected objects serve as masks, while in the predicted images, silhouettes of 3D objects serve as masks, see Fig. 3 (a) and (b). Hence E_s is evaluated in image regions of visible wall regions. Since visible wall areas are directly related to object size, the pixel-wise cost function E_s is sensitive to the global scale λ and object positions $\{p_j\}$. For example, if the estimated room scale λ is smaller than the true scale, objects in the synthetic scene will be placed closer to the camera, thereby occluding larger wall regions, which is penalized by the cost function E_s.

3.2. Object pose estimation

We define the second factor in the likelihood term in Eq. 3 as:

$$P(\mathcal{I}|\{\theta_j^o\}) \propto \exp\left[-E_o(\mathcal{I}, \{\theta_j^o\})\right], \quad (7)$$

where the cost function E_o evaluates object orientation hypotheses $\{\theta_j^o\}$ by comparing HOG descriptors [3] of detected objects in $\mathcal{I}$ and rendered images from corresponding 3D models. For the initial object pose estimation $\{\hat{\theta}_j\}$ (superscript omitted for clarity in this section), two distinct sources of data are employed: a set of rendered images $\mathcal{R}$ of 3D models with known pose, and a set of visually similar web images $\mathcal{W}$ found by Google Image search. The auxiliary set of images helps to regularize the solution when jointly estimating object pose, as demonstrated in [17] and confirmed in initial experiments. We therefore take the same approach as [17] with two extensions. First, we do not assume images with clean background and therefore extract the object in the input image by automatic grab-cut segmentation, assuming that the image center contains the object and image corners are part of the background. Further, [14] assumes that a very specific category type is known (e.g. Windsor chair). Our approach works with just knowing the abstract category (chair), and uses visual search to find web images similar to the target object. Therefore our approach is more robust against cluttered background and generalizes to a wider range of object categories. The web images are obtained automatically by retrieving the first 400 results of Google Image search for visually similar images within the detected object category. Background is removed from the web images by co-segmentation, since we assume that this image set contains a shared common object [7].

Given object bounding boxes, HOG descriptors are computed for each region in a 4×4 image-grid using unsigned gradients with ℓ_2-normalization, and are concatenated into a global image descriptor. A CRF model is then employed to regularize the pose estimation. Let $\mathcal{T}$ denote the cropped input image (or multiple images if the same object appears more than once in the scene) of objects for which we want to find the 3D pose θ. Each node in the CRF represents an image $I \in \mathcal{T} \cup \mathcal{W}$, and the label space is the quantized pose space sampled uniformly from yaw and pitch angles (360 poses from yaw $\in [0°, 360°]$ and pitch $\in [0°, 45°]$, roll angle is fixed). For image I we search for the K nearest neighbors among a set of rendered images $\mathcal{R}$ in a 3D database.

The unary potential is defined by the number of nearest neighbors in the rendered image set with the same discretized pose:

$$E^{(i)}_{\text{unary}} = \exp\left[-\sum_{\{I_k | I_k \in \mathcal{N}_i^{(K)} \subseteq \mathcal{R}\}} \mathbb{1}_{\theta_i = \theta_k}\right] \quad (8)$$

where $\mathbb{1}$ is the indicator function and $\mathcal{N}_i^{(K)}$ denotes the set of the $K = 6$ nearest neighboring images of I_i in terms of HOG-distance.

The binary potential between two images I_i and I_j in $\mathcal{T} \cup \mathcal{W}$ encourages smoothness between the predicted poses of neighboring images:

$$E^{(i,j)}_{\text{binary}} = d^{\gamma}(\theta_i, \theta_j)\, d^{HOG}(I_i, I_j), \quad (9)$$

where $d^{\gamma}(\theta_i, \theta_j)$ is the an angle distance function defined as

$$d^{\gamma}(\theta_i, \theta_j) = \min(d(\theta_i, \theta_j), \gamma)\ \text{, where} \quad (10)$$

$$d(\theta_i, \theta_j) = |\rho_i - \rho_j| + |\xi_i - \xi_j|, \quad (11)$$

and γ is a threshold, ρ is the yaw angle, and ξ the pitch angle. The energy function for the CRF is then:

$$E_{CRF} = \sum_{I_i \in \mathcal{T} \cup \mathcal{W}} E^{(i)}_{\text{unary}} + \sum_{\{I_i \sim I_j | I_i, I_j \in \mathcal{T} \cup \mathcal{W}\}} E^{(i,j)}_{\text{binary}}, \quad (12)$$

and CRF inference is performed using the TRW-S algorithm [18]. Qualitative results can be seen in Fig. 4. 3D model retrieval is performed by finding the nearest neighbor in HOG space among the images in $\mathcal{R}$. The cost function E_o for object orientation is the Euclidean distance of the descriptors.

3.3. Context prior

The context prior, $\pi(\Phi)$, evaluates the relative positions and orientations of objects and walls in a 2D top-down view of the scene. The object-to-wall cost $E_{o,w}$ measures distance and alignment of an object with its closest wall segment:

$$E_{o,w}(\Phi) = \sum_{j=1}^{N_o} \|p^o_j - p^w_{i^*(j)}\| + \nu_n \sum_{j=1}^{N_o} \|n^{o\top}_j n^w_{i^*(j)}\|, \quad (13)$$

where p^o_j is the position of object o_j, $i^*(j) = \text{argmin}_i\, d(p^o_j, p^w_i)$ is the index of the closest wall segment to object o_j, n^o_j and $n^w_{i^*(j)}$ are the normals of the object and its closest wall, respectively, and ν_n is a weighting factor. The object-to-object cost $E_{o,o}$ is a function penalizing the overlap between objects.

$$E_{o,o}(\Phi) = \sum_{j,k=1}^{N_o} A\left(b(o_j) \cap b(o_k)\right), \quad (14)$$

Figure 4: **3D pose estimation:** *Two pose estimation results for a segmented input image (top left) shown with the five 3D models closest in HOG space.*

where A is the area of intersection between two object bounding boxes, denoted as b. The prior term combines the object-to-wall and object-to-object costs and is defined as

$$\pi(\Phi) = \exp\left[-(E_{o,w}(\Phi) + \mu E_{o,o}(\Phi))\right], \quad (15)$$

where μ is a weighting factor.

3.4. MAP estimation

We use a sampling strategy to find room layouts with a maximum posterior solution, as defined in Eq. 2. From an initial estimate of 3D room geometry and 3D object pose, we use the context prior term to sample locations and orientations of objects, as well as the global scale parameter λ. Scale is sampled uniformly within a fixed interval, while object locations are sampled from a normal distribution that has large variance in the object-camera direction, accounting for distance ambiguity, and small variance normal to this direction, giving high confidence to the location predicted by the detector and lower confidence to the bounding box size. Object orientation is sampled from a normal distribution with a mean of the orientation found in section 3.2. For each of the N_S configuration samples we evaluate the likelihood terms (Eq. 3) and context prior (Eq. 15) and output the hypothesis with the maximum posterior value. Implementation details are given in the next section.

4. Results

The algorithm is validated on a subset of the public SUN360 dataset [37]. It contains panorama images of indoor scenes at high resolution (up to 9K) which we rescale to 2K to reduce computation time. We obtain reasonable initial pose estimations when object detection bounding boxes and segmentation are accurate (see Fig. 5(a)). However, directly applying state-of-the-art techniques is insufficient to obtain correct room layouts, as shown in Fig. 5(b). Even though surface alignments estimated from different perspective views return correct room shape, the absolute scale remains ambiguous. In addition, the initial object pose estimation $\{\hat{\theta}_j^o\}$ (*e.g.*, bed orientation) is not always accurate, and object distances to camera are unknown. In comparison, our proposed method returns more accurate results as shown in Fig. 5(c).

4.1. Quantitative evaluation

To evaluate the accuracy of the proposed method, we created ground truth data by manually annotating object positions and orientations in panorama input images. 34 bedroom images are selected from SUN360 dataset and the results are shown in Table 1. We measure positional error as distance between object centroids projected onto the 2D ground plane and orientation error as the angle between ground truth and estimated pose. As seen in Fig. 6, the estimation error is lower for certain object classes, *e.g.*, TV, where the pose can typically be estimated reliably and the object prior helps by favoring alignment with nearby walls. The error for chairs tends to be higher for several reasons: There is a large variation and symmetry of chair shapes, which can lead to less accurate bounding boxes and pose estimation. The example in Fig. 4 shows that the appearance for chairs can be very similar for rotated versions of the model. Our method does not attempt to estimate the orientation of potted plants since they tend to be rotationally symmetric. In addition, the joint estimation of room layout, scale and object pose allows us to automatically generate a 2D floor map from one panorama image, see Fig. 8(a).

Our method is compared to *PanoContext* [39] on the same image set using the code provided by the authors with default parameters. For each scene, $200,000$ hypotheses per room are generated together for predicting the object types and positions as well as generating the room layout (within a box-shaped room) as described in [39]. The hypothesis with the top-1 score is selected as estimation result. Since *PanoContext* does not contain room scale estimation, results are scaled to match with ground truth (see Fig. 6). As can be seen in Fig. 8(b), both of false positive (FP) and false negative (FN) rates of *PanoContext* are very high: for bed, chair, and TV, the FP rate is 59.1%, 43.2%, and 17.2%, respectively, while the FN rate is 22.7%, 89.2%, and 72.3%, respectively.

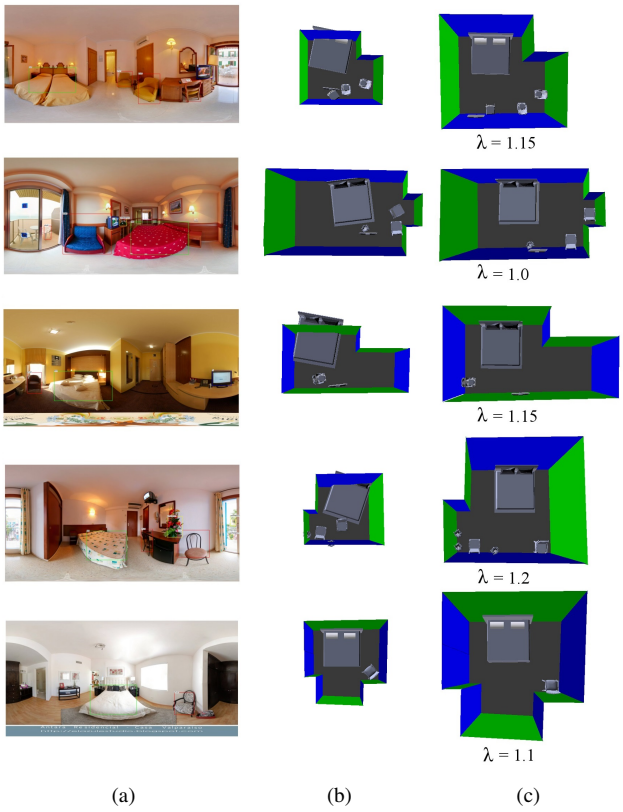


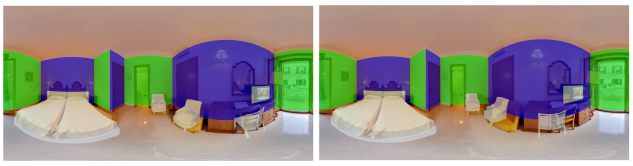
Figure 5: **Example results on SUN360 images:** *(a) Panorama images with detected objects. (b) Initial layout from estimated surface orientations. (c) Optimized layout of our result. Input images from the SUN360 dataset are on the left. The center column shows the initial layout estimation from object detection and orientation surface (with unknown global scale), the right column shows results after use of the context prior. Absolute wall height equals* $2.5m \times \lambda$*. The dimensions of 3D models from public datasets are known and remain fixed.*

In the comparison, only true positive detection results are chosen for position error calculation (see Table 1). Orientation estimation is not available in their method. Note that PanoContext has many more false detections, but that the location accuracy can be higher, since the correct detections in *PanoContext* are based on accurate low-level line features.

To further assess the method's accuracy, including scale estimation, we perform evaluations on generated 3D scenes as 3D ground truth. We synthesize 88 rooms of arbitrary shape, based on existing room templates, and size, containing objects. See Fig. 7 for examples. Wall heights are sampled from a normal distribution with mean 2.7m and 0.2m standard deviation. We add an offset to the length of each

Object	Position error (cm)		Orientation error (deg)	
Method	Ours	[39]	Ours	[39]
Bed	25.0 ± 17.4	23.7 ± 21.3	1.0 ± 1.4	n/a
TV	4.7 ± 6.4	5.6 ± 4.8	1.4 ± 1.1	n/a
Chair	52.3 ± 66.0	17.3 ± 14.6	10.7 ± 15.0	n/a
Plant	8.7 ± 12.0	n/a	n/a	n/a

Table 1: **Evaluation on SUN360 images:** *Object position and orientation errors measured against ground truth. [39] does not estimate orientation.*



(a) (b)

Figure 6: **Reprojection of 3D scene into panorama image:** *(a) Ground truth. (b) Our result. Comparison of the estimated layout to ground truth. Surface orientation and 3D objects are overlaid onto the input image. Camera parameter approximations and shape differences between real objects and 3D models can cause slight misalignment.*

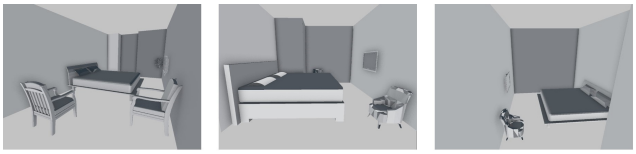


Figure 7: **Synthetic room data.** *Three example rooms out of 88 that were used in our quantitative evaluation.*

wall (only if longer than 0.7m), uniformly sampled from $[-0.3m, 0.3m]$. Objects are placed at random, and their position and orientation are updated by sampling from the context prior. Experimental results are reported in Table 2 and show the contributions of each step separately (object pose, room scale/wall height estimation). Overall estimation using $N_S = 3000$ samples is accurate to 0.8-8° and 2-20cm, depending on object class.

4.2. Implementation details

Object detection. We use the Faster R-CNN from [25] for object detection and recognition. The MS COCO [21] dataset is used for training, since it contains a large number of indoor object categories (*e.g., chair, couch, potted plant, bed, dining table, toilet, tv, laptop, microwave, oven, refrigerator, clock, vase*). The model was trained on the 80,000 image training set for 240,000 iterations with VGG16 networks [31], using top-2000-score Region Proposal Networks (RPN) [25] and Multi-scale Combinatorial Grouping (MCG) [23] as object proposals. The mean average precision (mAP) for all 80 object classes is 49.0%, and 26.5% for the intersection over union (IoU) values of 50% and 95%, respectively. The same metric and validation set was used for the MS COCO 2014 primary challenge. Note that our detection performance is competitive with the current state-of-the-art method by He *et al.* [12], which achieved corresponding mAP scores of 48.4% and 27.2%.

Object pose estimation. We collected a set of 3D models from the 3D Warehouse [33] and rendered each in 360 poses. For hotel rooms, we used 9 beds, 16 chairs, 4 plants, 6 TVs, and crawled 300-350 Internet images per class using Google image search. Note that results in Table 1 are for detected objects only. The number of nearest neighbors K is set to 6. The truncation threshold γ is set to 20°. TRW-S is run for 100 iterations to estimate object pose.

Context prior and sampling. Scale, object location and orientation are sampled by evaluating the context prior. We use 8 sampling epochs of 25 samples each. In every epoch the sample with largest context prior term is used as seed sample for the next epoch. The normal sampling distribution along the camera-to-object-center direction has the obtain location as mean, and a variance of 0.1 times the camera-object distance, and a variance of 0.005 times this distance along the perpendicular direction. Orientation is sampled from a normal with variance 0.1 rad, and scale is sampled uniformly, in terms of wall height, from the interval [2.0m, 3.5m]. The weight ν_n in Eq. 13 is set to 10.0, and μ in Eq. 15 to 0.25.

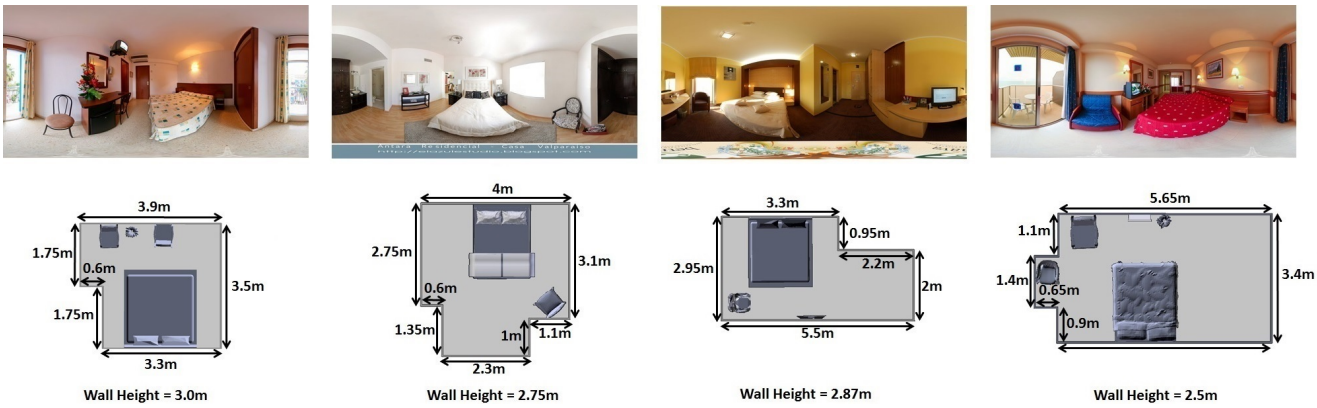
Computation time. The layout estimation pipeline is implemented on a desktop PC with i7 processor and 8GB RAM. The main bottleneck is currently the object pose estimation step using CRF optimization, which takes approximately 1-2min per object class. The object detection method in the pipeline takes 7s on average for 18 perspective images using a GRID K520 GPU. One room layout hypothesis evaluation requires about 30s. In comparison, *PanoContext* [39] requires over 2 hours to complete the overall computation with a Xeon E5-2630 v4 processor.

5. Conclusions

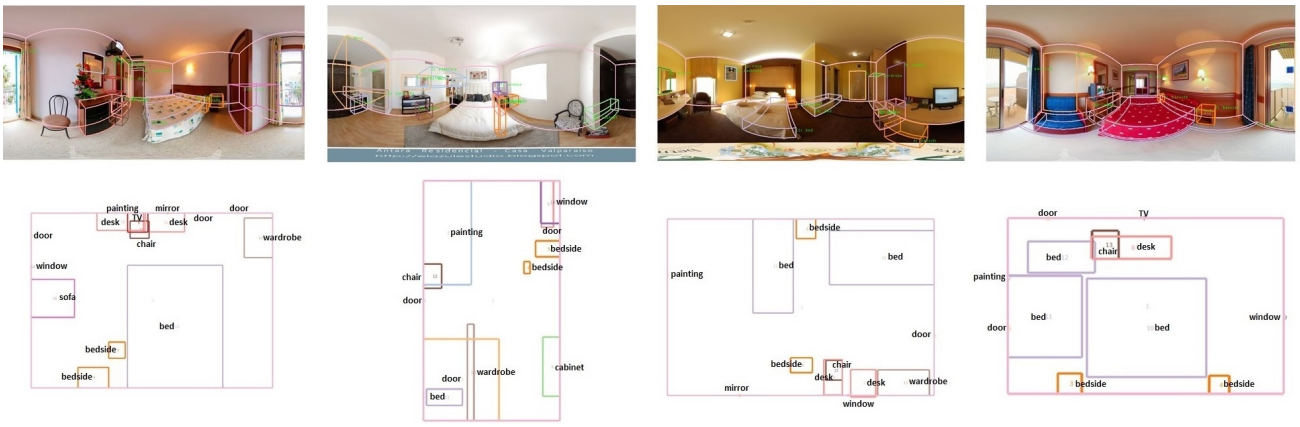
In this paper we presented a formulation for indoor layout estimation. We demonstrated its ability to recover complex room shape with Manhattan World assumption from a single panorama image using detected objects, their pose and their context in the scene. The proposed method does not rely on video, multiple images, or depth sensors as input [2] nor is it limited to box-shaped room or object models as in recent work on panoramic reconstructions [39]. Com-

Average errors	after initialization				with context term			
	bed	chair	TV	plant	bed	chair	TV	plant
$\epsilon_{\text{obj. orient.}}$(deg)	5.2±0.5	4.1±1.8	2.8±1.5	n/a	0.8±0.6	8.0±6.4	0.8±0.7	n/a
$\epsilon_{\text{obj. pos.}}$(cm)	197.6±57.6	186.7±99.6	156.21±73.0	174.7±70.3	21.0±13.0	7.1±7.3	2.0±7.0	6.9±0.5
$\epsilon_{\text{wall height}}$(cm)	n/a (initialized at 2.5m)				4.9±0.1			

Table 2: **Evaluation on synthetic dataset of 88 rooms.** *The table shows the mean error with standard deviation of object orientations, object positions, and wall height. The benefit of the proposed context prior is shown by comparing the results after the initialization stage (left) and after including context-based sampling (right). Average chair orientation error increases slightly. Note that the orientation error of potted plants is omitted, since they do not have a canonical orientation.*



(a)



(b)

Figure 8: **(a) Automatic 2D floor map generated by our method with estimated scale.** *The method estimates the global scale by transferring the scale of known 3D objects to scene objects. Examples shown were generated from SUN360 images.* **(b) Room layout generated by** *PanoContext* **[39].** *Results are obtained using the code provided by the authors with default parameters. 2D floor maps are generated for visual comparison.*

pared to [38], our method produces semantic output, taking the class, location, and pose of objects into account and introduces a context prior to this underconstrained problem. We evaluated the method quantitatively on a synthetic dataset and qualitatively on images from the SUN360 dataset. A limitation of the proposed method is that it currently relies on the output of an object detector. Objects that are not detected are currently not part of the final 3D model. Recent CNN-based methods for predicting depth and semantic labels [8] or 3D object pose [32] from images may be leveraged to improve the results.

References

[1] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *TPAMI*, 14(2):239–256. 4

[2] R. Cabral and Y. Furukawa. Piecewise Planar and Compact Floorplan Reconstruction from Images. In *CVPR*, 2014. 1, 2, 7

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005. 4

[4] L. Del Pero, J. Bowdish, D. Fried, B. Kermgard, E. Hartley, and K. Barnard. Bayesian geometric modeling of indoor scenes. In *CVPR*, 2012. 2, 4

[5] L. Del Pero, J. Bowdish, B. Kermgard, E. Hartley, and K. Barnard. Understanding Bayesian rooms using composite 3D object models. In *CVPR*, 2013. 1, 2

[6] E. Delage, H. Lee, and A. Ng. Automatic single-image 3D reconstructions of indoor manhattan world scenes, 2005. 4

[7] X. Dong, J. Shen, L. Shao, and M.-H. Yang. Interactive Cosegmentation Using Global and Local Energy Optimization. *Trans. Image Processing*, 2015. 4

[8] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015. 2, 8

[9] M. Fisher and P. Hanrahan. Context-based search for 3d models. In *ACM Transactions on Graphics (TOG)*, volume 29, page 182, 2010. 2

[10] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent, and R. Cipolla. SceneNet: Understanding Real World Indoor Scenes With Synthetic Data. In *arXiv:1511.07041v2*, 2015. 2

[11] C. Häne, L. Ladický, and M. Pollefeys. Direction matters: Depth estimation with a surface normal classifier. In *CVPR*, June 2015. 2

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 7

[13] V. Hedau, D. Hoiem, and D. Forsyth. Recovering the spatial layout of cluttered rooms. In *ICCV*, 2009. 2, 3

[14] V. Hedau, D. Hoiem, and D. Forsyth. Thinking inside the box: Using appearance models and context based on room geometry. In *ECCV*, 2010. 2

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

[16] D. Hoiem, A. A. Efros, and M. Hebert. Recovering Surface Layout from an Image. *IJCV*, 75(1):151–172, 2007. 2

[17] Q. Huang, H. Wang, and V. Koltun. Single-View Reconstruction via Joint Analysis of Image and Shape Collections. *ACM Transactions on Graphics*, 34(4), 2015. 2, 4

[18] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *TPAMI*, 28(10):1568–1583, 2006. 5

[19] 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. 2

[20] D. C. Lee, M. Hebert, and T. Kanade. Geometric Reasoning for Single Image Structure Recovery. In *CVPR*, 2009. 2, 3, 4

[21] T. Lin, M. Maire, S. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common Objects in Context. *arXiv:1405.0312v3*, 2015. 7

[22] P. Merrell, E. Schkufza, Z. Li, M. Agrawala, and V. Koltun. Interactive Furniture Layout Using Interior Design Guidelines. *ACM Transactions on Graphics*, 30(4), 2011. 2

[23] J. Pont-Tuset, P. Arbeláez, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping for image segmentation and object proposal generation. In *arXiv:1503.00848*, 2015. 7

[24] S. Ramalingam and M. Brand. Lifting 3d manhattan lines from a single image. In *ICCV*, December 2013. 2

[25] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NIPS*, 2015. 7

[26] S. Satkin, M. Rashid, J. Lin, and M. Hebert. 3DNN: 3D Nearest Neighbor. Data-Driven Geometric Scene Understanding Using 3D Models. *IJCV*, 111(1):69–97, 2015. 1, 2

[27] A. Saxena, M. Sun, and A. Y. Ng. Make3D: Learning 3D scene structure from a single still image. *TPAMI*, 31(5):824–840, 2009. 2

[28] A. G. Schwing, S. Fidler, M. Pollefeys, and R. Urtasun. Box In the Box: Joint 3D Layout and Object Reasoning from Single Images. In *Proc. ICCV*, 2013. 2

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

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

[31] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *arXiv:1409.1556*, 2015. 7

[32] H. Su, C. R. Qi, Y. Li, and L. J. Guibas. Render for CNN: Viewpoint Estimation in Images Using CNNs Trained with Rendered 3D Model Views. In *ICCV*, 2015. 2, 8

[33] Trimble. 3D Warehouse. https://3dwarehouse.sketchup.com/, 2016. [Online; last accessed 15-Mar-2016]. 7

[34] S. Tulsiani, A. Kar, J. Carreira, and J. Malik. Learning category-specific deformable 3D models for object reconstruction. *TPAMI*, 2016. 2

[35] H. Wang, S. Gould, and D. Koller. Discriminative learning with latent variables for cluttered indoor scene understanding. *Communications of the ACM*, 56(4):92–99, 2013. 2

[36] Y. Xiang, R. Mottaghi, and S. Savarese. Beyond PASCAL: A Benchmark for 3D Object Detection in the Wild. In *WACV*, 2014. 2

[37] J. Xiao, K. A. Ehinger, A. Oliva, and A. Torralba. Recognizing Scene Viewpoint using Panoramic Place Representation. In *CVPR*, 2012. 6

[38] H. Yang and H. Zhang. Efficient 3D room shape recovery from a single panorama. In *CVPR*, 2016. 2, 8

[39] Y. Zhang, S. Song, P. Tany, and J. Xiao. PanoContext: A Whole-room 3D Context Model for Panoramic Scene Understanding. In *ECCV*, 2014. 1, 2, 3, 4, 6, 7, 8