Bayesian geometric modeling of indoor scenes

Luca Del Pero Joshua Bowdish Daniel Fried Bonnie Kermgard Emily Hartley Kobus Barnard[‡]
University of Arizona
{delpero, jbowdish, dfried, kermgard, elh}@email.arizona.edu [‡]kobus@sista.arizona.edu

Abstract

We propose a method for understanding the 3D geometry of indoor environments (e.g. bedrooms, kitchens) while simultaneously identifying objects in the scene (e.g. beds, couches, doors). We focus on how modeling the geometry and location of specific objects is helpful for indoor scene understanding. For example, beds are shorter than they are wide, and are more likely to be in the center of the room than cabinets, which are tall and narrow. We use a generative statistical model that integrates a camera model, an enclosing room "box", frames (windows, doors, pictures), and objects (beds, tables, couches, cabinets), each with their own prior on size, relative dimensions, and locations. We fit the parameters of this complex, multi-dimensional statistical model using an MCMC sampling approach that combines discrete changes (e.g, adding a bed), and continuous parameter changes (e.g., making the bed larger). We find that introducing object category leads to state-of-the-art performance on room layout estimation, while also enabling recognition based only on geometry.

1. Introduction

We propose an approach for integrating object recognition with 3D reconstruction from monocular images of indoor scenes, based on 3D reasoning and Bayesian inference. Our goal is to simultaneously estimate the camera, detect and localize in 3D the floor, ceilings, and walls comprising the room "box," and determine the position and identity of the objects (e.g., beds, couches, and tables) and frames (doors, windows, and pictures) in the room. For example, if a room contains a couch, we would like to identify it as such, as well as understand where it is in 3D. Reasoning in terms of specific objects allows us to identify several different object categories, rather than generic bounding boxes or regions of occupied space. This improves the semantic parsing of the scene, as we can now identify objects that are typically found in these environments. In this paper we used couches, beds, tables, cabinets, windows, doors, and picture frames.

We are motivated by the recent advancements in the task of recovering the 3D geometry of indoor scenes [2, 3, 5, 9, 10, 12, 15, 16]. Specifically, there has been much interest in estimating the 3D layout of rooms from single images (position of walls, floor and ceiling), as this provides crucial information on the scene geometric context, which allows to reason about objects [6, 9] and human activities [4]. For example, current approaches can estimate what part of the 3D space is free and what part is occupied by objects, modeled either in terms of clutter [5] or bounding boxes [9, 12]. Also, Hedau et al. [6] identified beds by combining image appearance and the 3D reasoning made possible by the estimate of the room layout. To our knowledge, this was one of first attempts to provide a semantic parsing of the objects in indoor scenes based on 3D reasoning, followed by Gupta et al. [4], who managed to label surfaces and surface configurations in terms of affordances (i.e., the opportunities for human interaction provided by the environment, such as where a person can sit or reach). Further, Hoiem et al. [8] did significant work on combining 3D geometry and semantics in the scope of outdoor scenes.

In this paper we focus on exploiting the geometric properties and typical locations of specific objects for better indoor scene understanding. Simultaneously identifying objects while fitting their geometry and location improves both, and also improves the global room layout and the estimate of the camera. A key intuition is that we can discriminate between many indoor scene object categories using only gross geometry (size, relative dimensions, and position). For example, beds are much wider than they are tall, while wardrobes are the other way around (Figure 2). Similarly, position and size with respect to the room box also provide hints to object identity. For example, the height of a door is usually very close to the height of the room (Figure 3), but this is usually not the case for picture frames.

To integrate all these factors we use a Bayesian generative statistical model for the geometry of indoor scenes and entities within them. We set rough priors on object dimensions and their typical location from a held out image data set and from text in on-line Ikea and Home Depot catalogs (§3.2). These priors are combined with an edge like-

lihood model similar to one we used in previous work [12]. Since we are focused on exploring the use of geometry, we use only edge information and do not consider color or texture. We fit the scene model using an MCMC approach that combines sampling over both discrete and continuous variables, and uses multiple threads to speed up convergence (§4).

We find that the room layout estimation benefits from using specific objects coming from realistic categories, rather than plain bounding boxes [9, 12] or voxel occupation [5]. Using a model where every component has specific semantics associated to it (e.g. beds, windows, etc.) is a key contribution of our approach, and allows us to achieve state-of-art room layout results by using geometric information only, and with minimal training (§5). In addition, our object category recognition results, based on minimal geometric information, are promising.

2. Overview

As in previous work in this domain [9, 12], we model an indoor scene as a collection of right-angled parallelepipeds (blocks), parametrized in terms of the 3D position of their center and size. A single block is used to model the room box. Objects in the scene are also approximated using blocks, which provide reasonable bounding-boxes for most furniture, for example beds and tables. Block objects have to lie on the room floor or be attached to a wall in case of doors and windows, and cannot overlap. This is a suitable scenario for incorporating priors on object size and position in 3D, as all this information is encoded in the model.

To integrate this prior information with evidence coming from the image data D, we rely on a generative Bayesian framework. We define θ as the model parameters, which comprise block parameters and camera parameters, assuming that the image data is generated by the projection of the blocks in the scene under the given camera. We then introduce the posterior distribution

$$p(\theta) \propto p(D|\theta)\pi(\theta) \quad , \tag{1}$$

where $p(D|\theta)$ is the likelihood function and $\pi(\theta)$ the prior over model parameters. We use category-dependent priors, that inform both where objects in a specific category tend to be, and the size of each dimension (e.g. beds are usually quite short and wide, and against a wall). While we approximate all objects with simple bounding boxes, these category-dependent priors allow us to estimate the most likely class for a given object, based on its position and size. During inference the likelihood and the prior can act as competing forces, as the former will force the objects to change in order to better fit the image data, while the latter will prevent them from changing to positions or sizes that are unlikely for that specific class. Intuitively, a good solution is when an object of the right category fits the image data well and, at the same time, its parameters will be in a region of high probability for the prior distribution for that category.

In our approach, the room box and the objects within it are fit to image data simultaneously for two main reasons: 1) one cannot robustly identify the room box and the camera without adding objects in it, since the layout can be estimated correctly only if we take occlusions into account [9, 12]; and 2) an individual object can be identified more effectively if we take into account the contextual information provided by its position and size with respect to the room box and the other objects in the scene.

3. A geometric model for indoor scenes

The model parameters $\theta = (s, c)$ comprise scene s and camera parameters c. As explained above, the scene consists of a collection of blocks, parametrized in terms of the 3D position of the center, their width, height, length, and the amount of rotation γ around their y axis

$$b_i = (x_i, y_i, z_i, w_i, l_i, \gamma_i) \quad . \tag{2}$$

The room box itself is approximated with one of such blocks

$$r = b_r = (x_r, y_r, z_r, w_r, l_r, \gamma_r) \quad . \tag{3}$$

These structures are also used to model objects inside the room, since they can provide a reasonable approximation for pieces of furniture such as couches and beds (one can think of them as bounding boxes), or for objects on the walls, which we call frames. Windows and doors are an example, and are approximated with very thin blocks. We define each object in the room as

$$o_i = (b_i, t_i) \quad , \tag{4}$$

where t_i defines the object type. The whole scene is then modeled as a room box containing an unknown number of objects n

$$s = (r, o_1, ..., o_n) \quad . \tag{5}$$

We parametrize the camera as in our previous work [12]

$$c = (f, \phi, \psi) \quad , \tag{6}$$

where f, ϕ and ψ are, respectively, the focal length, the pitch and the roll angle. Since we cannot determine absolute position when reconstructing from single images, we can arbitrarily position the camera at the world origin, looking down the negative z axis. This, together with ϕ, ψ and the rotation angle of the room γ_r, fully determine the camera extrinsic parameters [12, 13]. Finally, we assume that the principal point is in the image center, and that there is no skew.

3.1. The image model

Our image model is similar to the one used by Schlecht and Barnard [13]. Specifically, we assume that given an instance of the model parameters $\theta_i = (s_i, c_i)$, image features $D = (f_1, ..., f_s)$ are generated by the projection of the 3D scene s_i under the given camera. We use two features that proved useful in this domain: edges [12] and orientation surfaces [9].

Image edges. We assume image edges to be generated by the blocks in the scene. We measure the quality of a fit by comparing the set of edges E_d detected on the image plane to the set of edges E_m generated by projecting the model. As Schlecht et al. [13], we define a likelihood function $p(E_d|E_m)$, which we approximate using the following intuitions:

- An edge point $e_{dj} \in E_d$ detected in the image plane should be matched to an edge point $e_{mk} \in E_m$ generated by the model. If the match is good the two points should be very close to each other, and the difference in orientation between the two should be minimal. We approximate $p(e_{dj}|e_{mk}) = \mathcal{N}(d_{jk}, 0, \sigma_d)\mathcal{N}(\phi_{jk}, 0, \sigma_\phi)$, where d_{jk} is the distance between the points, and ϕ_{jk} the difference in orientation between the edges.
- We penalize a detected edge point that is not matched to any model edge (noise). We define p_n as the probability of such an event occurring
- We explain points in E_m not matched to any point in E_d as missing detections, and define probabilities p_{hmiss} and p_{smiss}. The former is used for "hard" edges arising from occlusion boundaries, such as the edges that belong to the silhouette of an object. The latter is used for "soft" edges that are less likely to be found by the edge detector, such as the room edges and non-silhouette edges from objects. Notice that the detector missing a "hard" edge is less likely than a "soft" edge. One of the advantages of using a full 3D model, is that we can determine whether edge points in E_m are soft or hard.

We then have

$$\tilde{p}(E_d|E_m) = p_n^{N_n} p_{smiss}^{N_{smiss}} p_{hmiss}^{N_{hmiss}} \prod_{(j,k)\in matches} p(e_{dj}|e_{mk}) \quad , \tag{7}$$

where N_n is the number of edge points labeled as noise, and N_{smiss} (N_{hmiss}) the number of missed soft (hard) edges. We match points in a greedy fashion by finding the closest point e_m to a data edge e_d along the edge gradient, provided that this distance is smaller than 40 pixels. We further adjust this likelihood function, in order to make it independent of the number of edge points, which we found makes it more stable over a larger variety of input data. Specifically, we use

$$p(E_d|E_m) \approx \tilde{p}(E_d|E_m)^{(N_{hmiss}+N_{smiss}+N_n+N_{matches})^{-1}} \quad . \tag{8}$$

Orientation surfaces. Indoor environments typically satisfy the Manhattan World assumption, since most surfaces in the scene are aligned along three orthogonal directions. Thus, most pixels in the scene are generated by a plane aligned with one of these directions, and we can estimate which one using the approach by Lee et. al [10]. We compare the pixel orientation O_d detected from the image plane with the orientation surfaces O_m generated by projecting our model. We approximate $p(O_d|O_m)$ as the ratio between the number of pixels such that the orientation detected on the image plane agrees with the orientation predicted by the model, and the total number of pixels. Notice that this is a number between 0 and 1.

Combining the two features. Assuming independence between the edges and the orientation features, we define our likelihood function

$$p(D|\theta) = p(E_d|E_m)p(O_d|O_m)^\alpha \quad , \tag{9}$$

where α is used to weigh the importance of the orientation likelihood (experiments on the Hedau test set suggest that 6 is a good value for this parameter). In Figure 1 we show that these two features work very well together, as the errors in the edge detection process can be fixed using orientation surfaces, and vice versa. Using edges also helps improving the camera fit when starting from a wrong estimate of the vanishing points, which are detected at the beginning and used to initialize the camera parameters [12]. In fact, since the algorithm for computing orientation maps depends on the initial vanishing point estimation, this feature is compromised by this initial error, whereas edges are not.

3.2. Model priors

A major novelty of our approach is that priors on scene elements help global scene understanding, and are also key for identifying objects based on geometry cues, such as size and location, alone. For example, wardrobe cabinets are tall and narrow and typically against the wall, while beds are short, wide, and often against the wall. Notice that, since we are reconstructing from a single image, we have one overall scale ambiguity, and thus priors on object "size" and camera height are always relative to the overall room box size.

We start by introducing priors over the room box $\pi(r)$, the camera parameters $\pi(c)$, and each object o_i inside the room $\pi(o_i)$. Assuming independence, we compute the prior over the model parameters as

$$\pi(\theta) = \pi(r)\pi(c)\left(\prod_{i=1}^{n}\pi(o_i)\right)^{\frac{e}{n}} \quad , \tag{10}$$

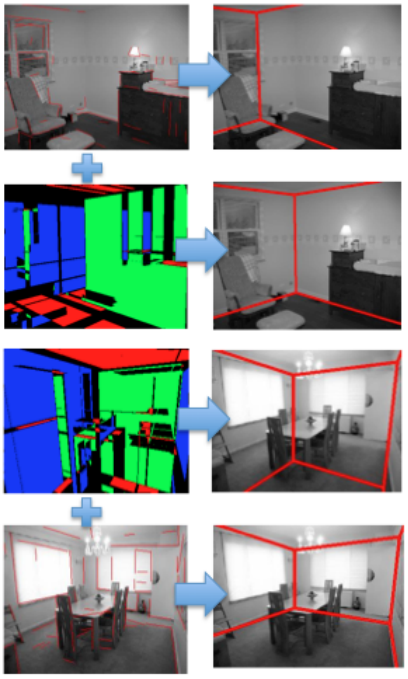

Figure 1. Advantages of integrating edges and surface orientation in the likelihood function. Faint wall edges are often missed by the edge detector (top left), and the edge likelihood alone would provide the wrong solution, by "latching" the wall edge to the window (top right). However, in this case the orientation surfaces help converge to the right solution (second row). Conversely, mistakes in the orientation map estimation can be overcome by relying on edge information (third and fourth rows).

where we take the geometric means of the object priors, so that we can compare models with a different number of objects. e is a stabilizing factor, which we set to 0.3.

In what follows, we describe each of these components, followed by how we set their parameters from training data.

3.2.1 Prior on room box

The room box is defined in terms of the center position in 3D (x_r, y_r, z_r) and its width, height and length (w_r, h_r, l_r). First, we define a prior over the ratio between width and length

$$r_{r1} = \frac{max(w_r, l_r)}{min(w_r, l_r)} \quad . \tag{11}$$

We use this formulation since we do not know in advance which dimension is the largest. We are also interested in the average ratio between room width (length) and height

$$r_{r2} = \frac{max(w_r, l_r)}{h_r} \quad . \tag{12}$$

As explained in further detailed in the Section 3.3, these two quantities have very little impact on the quality of the final solution. However, they reduce time spent in regions of the sampling space with low probability, especially during the early stages of the inference process. Further, these two components prevent the sides of the room that are not visible from expanding arbitrarily, and this also makes the inference more efficient. We then have

$$\pi(r_b) = \mathcal{N}(r_{r1}, \mu_{r1}, \sigma_{r1})\mathcal{N}(r_{r2}, \mu_{r2}, \sigma_{r2}) \quad , \tag{13}$$

where we assumed that the two quantities are normally distributed and independent.

3.2.2 Prior on camera parameters

We found that the camera height from the floor c_h is a particularly discriminative feature in indoor scenes. Small variations in this quantity result in major changes in the image plane. For this reason, we introduce a prior on the ratio between camera height and room height (again, we cannot use absolute sizes)

$$\pi(c) = \mathcal{N}(c_h, \mu_{ch}, \sigma_{ch}) \quad . \tag{14}$$

3.2.3 Prior on objects.

Several categories of furniture and frames have a very distinctive size (Figure 2). In this section, we introduce a general formulation for a prior for a specific object category τ that exploits this intuition. Given an object o_i defined in terms of its size (w_i, h_i, l_i), and a room with dimensions (w_r, h_r, l_r), we are interested in the following quantities

- ratio between object height and largest dimension $r_{i1} = h_i/max(w_i, l_i)$ (Figure 2)
- ratio between object width and length $r_{i2} = max(w_i, l_i)/min(w_i, l_i)$ (Figure 2)
- ratio between room height and object height $r_{i3} = h_r/h_i$ (Figure 3)

The first two carry information on the object structure, and do not depend on the scene. As shown in Figure 2, both features can help distinguish between different object classes. Notice that we do not use the second component for frames, since these objects are very thin blocks attached to a wall, and it would thus not make sense to use this measure.

The third quantity encodes information on the relative size of an object with respect to the room. Intuitively, the height of a bed is quite small with respect to the room height, whereas the height of a wardrobe or of a door is quite large (Figure 3). Assuming that these quantities are normally distributed, we introduce prior distributions

$$\pi_j(o_i|t_i = \tau) = \mathcal{N}(r_{ij}; \mu_{\tau j}, \sigma_{\tau j}) \quad , \tag{15}$$

for $j = 1, 2, 3$. Each category τ has different $(\mu_{\tau j}, \sigma_{\tau j})$. For object o_i, we use the prior distribution for the category it belongs to, denoted by t_i. Notice that from now on we will use the shorthand $\pi_j(o_i)$ for $\pi_j(o_i|t_i = \tau)$.

Last, we introduce a fourth component that relates the position of an object to that of the room box. We use a discrete variable d_i that takes two possible values depending on whether o_i is against a wall or not, based on the intuition that some objects tend to be against a wall (e.g. beds) more than others (tables). For frames, we use the position with respect to the floor instead. For example, doors touch the floor, while windows typically do not. For each category, we introduce distribution $p_\tau(d_i)$ over these two possible values.

Last, given an object o_i, we combine the components of its prior probability as follows

$$\pi(o_i) = p_{t_i}(d_i) \prod_{j=1}^{3} \pi_j(o_i) \quad . \tag{16}$$

3.3. Setting prior probabilities from data

Setting prior probabilities for objects categories. As mentioned above, the first two components of the object prior are independent of the scene. For each category τ, we set $(\mu_{\tau 1}, \sigma_{\tau 1}, \mu_{\tau 2}, \sigma_{\tau 2})$ using fifty random examples selected from online furniture and appliances catalogs. We recorded their dimensions, provided in the text description, and the means and variances of the relevant ratios. We used the Ikea catalog [1] for beds, couches, cabinets and tables, and the Home Depot catalog [2] for windows, doors and picture frames.

Setting the parameters for the remaining two priors is more challenging, since they relate the size of an object category to that of the room, and this information is not available in furniture catalogs. In this case, we rely on image data, and set (μ_{c3}, σ_{c3}) as explained in Figure 3. We also use images to set $p_\tau(d)$, which we approximate as the frequency at which an instance of an object of category τ is against a wall, or floor if it is a frame. For training, we used the images in the test split of the Hedau dataset [5]. We did not use images with ambiguous examples, where we could not tell whether a piece of furniture was against the wall or not.

Setting prior probabilities for camera and room parameters. We use training images in order to set these parameters. Following the ground truth procedure we introduced in previous work [12], we manually fit an empty room box and camera to the images in the training set. From this data, we can set the camera height parameters μ_{ch}, σ_{ch}. Setting parameters $(\mu_{r1}, \sigma_{r1}, \mu_{r2}, \sigma_{r2})$ for the room box prior is more challenging, since walls are completely visible only in a few images. For example, we can use the right image in Figure 3 to compare the lenght of the back wall to the room height, but we cannot use the left image, since each wall is partly outside the image plane. We then use images like the former to set mean μ_{r2} and variance σ_{r2} of the ratio between room width and height. Since the main purpose of these two components is to prevent the room box from expanding too much, we can assume roughly square rooms and set $\mu_{r1} = 1$. We use a large variance σ_{r1} to account for non square rooms and corridors.

[1] http://www.ikea.com/us/en/catalog/categories/departments/bedroom/
[2] http://www6.homedepot.com/cyber-monday/index.html

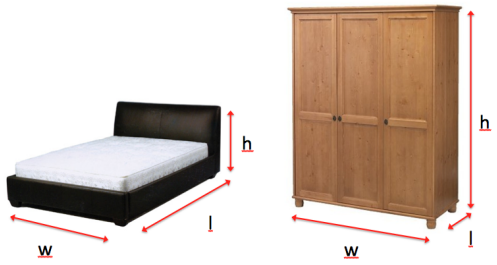

Figure 2. In indoor environments, we can distinguish object classes based on their size. However, when reconstructing from single images we cannot determine absolute sizes, and we thus have to use size ratios. For example, the ratio between the height of an object and the largest between its width and length varies considerably between beds and cabinets. Similarly, the ratio between width and length can be quite discriminative too (to avoid ambiguities, we use the ratio of the largest of the two to the smallest). These two quantities define a prior on object size within a category.

Figure 3. The relative size of an object with respect to the room is a very discriminative feature. We are interested in the ratio between the room height (yellow arrow) and the object height (red arrow). This can be estimated from image data by dividing the length of the yellow arrow (in pixel) by the red arrow, provided that the object is against or close to a wall. Notice that ratios of lengths of collinear segments are normally not preserved by projective transformations (only affine). However, in this domain the vanishing point for vertical segments is usually at infinity, and this method provides a reasonable approximation.

4. Inference

We fit our model to images by sampling from the posterior distribution $p(\theta|D)$ using a reversible jump MCMC strategy. We alternate "jump" moves to add/remove objects to the scene (e.g. add a couch, or remove a door), and "diffusion" moves to sample over continuous parameters (e.g., changing the position and size of an object, of the room box or of the camera).

Diffusion moves. As in our previous work [12], we use Stochastic Dynamics [11] for sampling over subsets of the continuous parameters. We alternatively sample over

- object size and position. Basically, objects are shifted around the room and stretched, and the room box and other objects have to adjust to avoid collisions
- room size and position. Here, we change the room box. When needed, objects must shrink or move in order to remain fully inside the room
- camera parameters

Jump moves. These moves are used to change the discrete structure of the model by adding and removing objects, since the number of objects in the room is not known a priori. Each jump move is accepted or rejected using the Metropolis Hastings acceptance formula. Here, we need a mechanism to propose objects of the right size at the right place, otherwise the acceptance ratio will be extremely low. Specifically, we use a data-driven [14] strategy for adding blocks to the scene, relying on orthogonal corners detected onto the image plane [12], which proved very effective in this domain. Further, instead of proposing an object of random size, we randomly select a category τ from the set of furniture and frame categories available, and draw a sample from the size prior for τ (e.g. we propose adding a bed or a cabinet, rather than adding a generic block). If a proposal gets accepted, the object just added will be considered as an instance of class τ. We also introduce a jump move for proposing a category change for a given object (e.g. we propose to turn a "bed" into a "table"). To summarize, we use the following set of jump moves

- adding an object of a specific type to the scene from a randomly selected orthogonal corner.
- removing an object
- changing the category of an object
- proposing a different room box from a corner to replace the current one

Once an object is added to the scene, diffusion moves will try to change its size and position. As already mentioned, the prior and the likelihood will act as competing forces in this process. The latter will try to change the object parameters to better explain image edges and orientation surfaces, while the former will constrain these changes to regions of parameter spaces that are likely for the class the object belongs to. This prevents objects from assuming unnatural sizes to satisfy the likelihood function, as shown in Figure 4, where we ran the sampler without using the prior. Not using the prior has two negative effects. First, it slows down the sampler, since most of the time is spent exploring regions of parameters space that would be unlikely according to the priors. Second, it also has a negative impact on the final solution, since "objects" that are good fits for noise are clearly not reasonable according to the prior.

4.1. Initializing and running the sampler.

We use a multi-threaded sampler to efficiently explore more of the space on modern multi-core workstations. Some details on how we initialize the sampler follow.

Finding the most promising room boxes. We start by proposing a room box from each of the detected orthogonal corners, and we initialize the camera parameters from a triplet of orthogonal vanishing points detected onto the image plane, which is relatively standard procedure [5, 9, 10, 12, 15]. We sample briefly over the room box and camera parameters of the most promising proposals, and keep the 20 room boxes with the highest posterior value. We then use a multi-threaded strategy, where we run samplers in parallel, each initialized with one of the 20 best room boxes found so far. Notice that we are not exclusively committing to these 20 boxes, since each thread can change the parameters of the room box during execution.

Finding the most promising corners. As a last part of the initialization process, each thread iterates over the orthogonal corners, and generates object proposals for each category τ, relying on the best room hypotheses found in the previous step. We keep track of the corners that generated the object proposals with the highest posterior, and make sure that they will be used more frequently to propose objects during the sampling process itself. This initialization considerably increases the acceptance ratio of the sampler, since likely objects are proposed more often. We emphasize that we cannot iteratively add the most likely object to the scene, since early commitment to partial configurations leads to error [10].

After initialization, each thread randomly alternates the sampling moves described in the previous section. At the end of the sampling process, we join the threads and return the best global solution. The whole process takes, on average, 12 minutes per image.

5. Results

We start by evaluating the performances of the various components of our algorithm in terms of room layout estimation, which is a standard measure in this field. This measure relies on ground truth data where each pixel in the image was labeled according to the room face it belongs to (i.e. 1= ceiling, 2= floor, 3 = right wall, etc.). The error

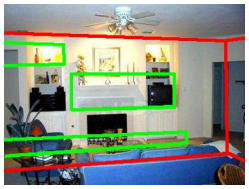

Figure 4. If we run the sampler without a prior on object size, significant time is wasted exploring regions of space that do not correspond to realistic configurations. Here, we can see a sample with a very high likelihood, since the long edges of the frame at the bottom happen to match image edges very well (especially the one formed by the pillows), and the room floor is nicely "latched" to the edge generated by the back of the couch. Without a component in the prior penalizing the unlikely size of the frame, it would take a long time before the sampler gets out of this deep local minimum.

is measured by comparing the projection of the estimated room layout against the ground truth, and computing the ratio of misclassified pixels. In Table 1, we can see the benefits of integrating the camera and room prior in the model, as well as the orientation component of the likelihood. We then compare the scenario where blocks of random size and no prior are used to our full algorithm, where we add specific objects such as beds and tables. We ran the two approaches for the same number of iterations, and we can see that the latter performs much better. In this case, the acceptance ratio of the sampler is much higher, since we are proposing realistic objects in likely positions, and this helps converging to a good solution. Notice also that a prior on objects also solves the problem illustrated in Figure 4. In Table 2, we can see that our results are comparable to the state-of-the-art on two standard datasets.

Then, we measured performances on object recognition. We ground truthed the 340 images in the UCB dataset [16] by manually identifying the seven object classes we experimented with (we did not consider objects occupying less than 1% of the image). We believe this dataset is harder than the Hedau test split [5], and we hope that this ground truth, which we made available online [1], will stimulate further experiments on this data.

To evaluate detection, we project the 3D object hypothesized by our model onto the image plane, and compare this with the ground truth object position. If the intersection of the two areas is more than 50% of their union, we consider it a correct detection. In Table 3, we first report precision and recall for the four furniture classes (beds, couches, cabinets and tables) and for the frame frame classes (door, window, picture frame). Here we consider a piece of furniture as correctly detected even if we confused, say, a couch for a table, and similarly for frames. We also report a confusion matrix for the instances of furniture (and frames) that were correctly identified.

We believe that these results are promising, considering that only 3D geometry and edge features are used. Visual

Method	Error on Hedau (test) [5]
only edge likelihood (no blocks)	26.0 %
+ camera and room prior (no blocks)	24.7 %
+ orientation likelihood (no blocks)	21.3 %
+ random blocks	19.7 %
+ objects	16.3 %

Table 1. Average error on room layout estimation on the test split of the Hedau dataset. We can see the benefit of adding each component discussed so far (see text for details).

Figure 5. Full scene reconstructions. **Best viewed in color.**

Method	UCB room	Hedau (test) [5]
Del Pero CVPR 11 [12]	24.0 %	26.8 %
Lee NIPS 10 [9]	NA	16.2 %
Our method	18.4 %	16.3 %

Table 2. Average error on room layout estimation on two standard datasets. Our approach is comparable to the state-of-the-art.

inspection (Figures 5 and 6) suggests that adding realistic objects to the scene improves the semantic understanding of the scene. The fact that results look consistent across two different datasets is promising too, as reconstructions in Figure 5 come from both datasets. The experiments also suggest that appearance models are needed to achieve better recognition. We can see that it is easy to make confusion between beds, couches and tables, which are quite similar in size compared to cabinets, which get confused less often. The same problem occurs between windows and picture frames. This and other limitations of our algorithm are shown in Figure 5. Visual inspection showed that in many of these cases, the posterior of the right solution is very close to the false positive. This suggests that adding appearance or geometric context information [7, 9] could improve results considerably.

Figure 6. Some objects correctly identified. **Best viewed in color.**

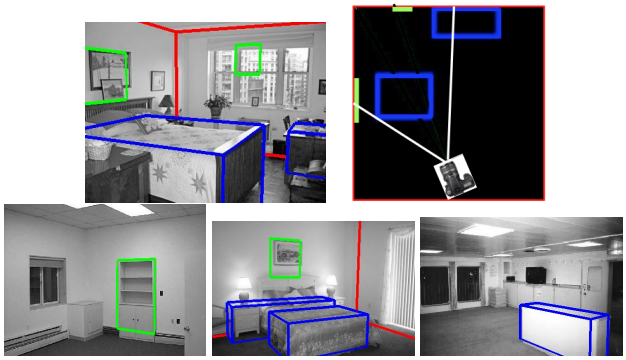

Figure 7. Current limitations of our approach. Top: the block over the bed seems a good fit, but is actually in the middle of the room, and is labeled as a table. We can see this in the birdview on the right, where the white rays indicate the camera field of view. In the birdview we render the full model that was fit to the image, and this includes parts of the room that are not visible in the image plane. Despite being wrong, the table block explains image features very well, and this is a major source of confusion. Bottom left: objects facing the camera directly can be both interpreted as furniture or frames. We think both problems can be solved by incorporating image appearance, which is likely to fix also mistakes like the ones in the second image of the bottom row. Last, using only priors on size generates false positives (bottom right).

	Precision	Recall
Furniture	31.0 %	20.1 %
Frames	27.7 %	19.7 %

	Bed	Cabinet	Couch	Table
Bed	26	1	12	6
Cabinet	1	11	0	3
Couch	16	3	6	14
Table	13	4	5	5

	Door	Picture	Window
Door	6	1	9
Picture	0	23	22
Window	7	14	49

Table 3. Top. Detection accuracy of furniture and frames considered as groups. Bottom two tables. Confusion matrices for identified furniture and frames. Out of the pieces of furniture detected by our approach, 38% are correctly classified. The ratio of correct classifications for frames is 60%.

6. Conclusions

Fitting specific objects, characterized with sensible priors, significantly helps scene understanding. In particular, room layout performance increased, and we had some sense of object identity. Our investigations only considered block edges; we expect that training appearance models would help correctly tag blocks. However, using only geometry, we are able to get state-of-the-art scene layout results.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 0747511.

References

[1] Ground truth object masks for the UCB dataset. http://kobus.ca/research/data/CVPR_12_room. 7

[2] E. Delage, H. Lee, and A. Ng. A dynamic bayesian network model for autonomous 3d reconstruction from a single indoor image. In *CVPR*, pages II: 2418–2428, 2006. 1

[3] E. Delage, H. Lee, and A. Y. Ng. Automatic single-image 3d reconstructions of indoor manhattan world scenes. In *ISRR*, 2005. 1

[4] A. Gupta, S. Satkin, A. A. Efros, and M. Hebert. From 3d scene geometry to human workspace. In *CVPR*, pages 1961–1968, 2011. 1

[5] V. Hedau, D. Hoiem, and D. Forsyth. Recovering the spatial layout of cluttered rooms. In *ICCV*, 2009. 1, 2, 5, 6, 7

[6] V. Hedau, D. Hoiem, and D. Forsyth. Thinking inside the box: Using appearance models and context based on room geometry. In *ECCV*, 2010. 1

[7] D. Hoiem, A. Efros, and M. Hebert. Geometric context from a single image. In *ICCV*, 2005. 7

[8] D. Hoiem, A. A. Efros, and M. Hebert. Closing the loop on scene interpretation. In *CVPR*, 2008. 1

[9] D. Lee, A. Gupta, M. Hebert, and T. Kanade. Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. In *NIPS*, 2010. 1, 2, 3, 6, 7

[10] D. Lee, M. Hebert, and T. Kanade. Geometric reasoning for single image structure recovery. In *CVPR*, 2009. 1, 3, 6

[11] R. M. Neal. Probabilistic inference using markov chain monte carlo methods. Technical report, 1993. 6

[12] L. D. Pero, J. Guan, E. Brau, J. Schlecht, and K. Barnard. Sampling bedrooms. In *CVPR*, 2011. 1, 2, 3, 5, 6, 7

[13] J. Schlecht and K. Barnard. Learning models of object structure. In *NIPS*, 2009. 2, 3

[14] Z. Tu and S.-C. Zhu. Image segmentation by data-driven markov chain monte-carlo. *PAMI*, 24(5):657–673, 2002. 6

[15] H. Wang, S. Gould, and D. Koller. Discriminative learning with latent variables for cluttered indoor scene understanding. In *ECCV*, 2010. 1, 6

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