Understanding Bayesian rooms using composite 3D object models

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

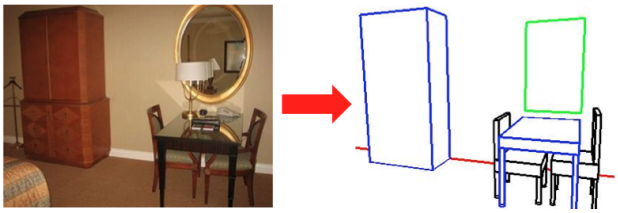

Figure 1. The use of more specific and detailed geometric models as proposed in this paper enables better understanding of scenes, illustrated here by localizing chairs tucked under the table in 3D.

Abstract

We develop a comprehensive Bayesian generative model for understanding indoor scenes. While it is common in this domain to approximate objects with 3D bounding boxes, we propose using strong representations with finer granularity. For example, we model a chair as a set of four legs, a seat and a backrest. We find that modeling detailed geometry improves recognition and reconstruction, and enables more refined use of appearance for scene understanding. We demonstrate this with a new likelihood function that rewards 3D object hypotheses whose 2D projection is more uniform in color distribution. Such a measure would be confused by background pixels if we used a bounding box to represent a concave object like a chair.

Complex objects are modeled using a set or re-usable 3D parts, and we show that this representation captures much of the variation among object instances with relatively few parameters. We also designed specific data-driven inference mechanisms for each part that are shared by all objects containing that part, which helps make inference transparent to the modeler. Further, we show how to exploit contextual relationships to detect more objects, by, for example, proposing chairs around and underneath tables.

We present results showing the benefits of each of these innovations. The performance of our approach often exceeds that of state-of-the-art methods on the two tasks of room layout estimation and object recognition, as evaluated on two bench mark data sets used in this domain.

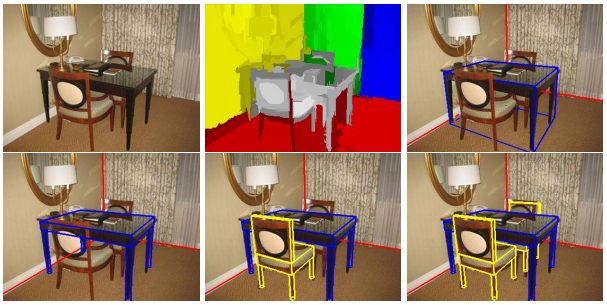

Figure 2. A visual summary of the main contributions of this work. 1) Detailed geometric models, such as tables with legs and top (bottom left), provide better reconstructions than plain boxes (top right), when supported by image features such as geometric context [5] (top middle), or an approach to using color introduced here. 2) Non convex models allow for complex configurations, such as a chair under a table (bottom middle). 3) 3D contextual relationships, such as chairs being around a table, allow identifying objects supported by little image evidence, like the chair behind the table (bottom right). **Best viewed in color.**

1. Introduction

Indoor scene understanding from monocular images has received much recent interest, and advancements in estimating the 3D geometry of such environments [2, 7, 11, 17] have enabled several interesting applications. For example, Gupta et al. [1] used extracted 3D information to predict human activities, while Hedau et al. [3] and our previous work [10] showed that knowledge of the 3D environment helps object recognition. Further, Karsch et al. [6] exploited the inferred 3D geometry to insert realistic computer graphics objects into indoor images.

State-of-the-art approaches to modeling indoor scenes largely use right-angled parallelepipeds (3D boxes) as containers. A single box is used to approximate the walls, floor, and ceiling enclosing the scene (**room box**), and also to represent objects inside it, such as beds and tables. Alternatively, blocks have been used to reason about clutter, but without further understanding of what is in the scene [7]. These representations allow promising reconstruction results, and hold the advantage that gross geometric structure simplifies inference.

However, these representations have four main limitations, which we illustrate using Figure 2. First, bounding boxes of concave objects projected into images tend to include much background, which is confusing evidence for inference. For example, the middle-top image shows the output of an appearance-based 2D classifier (geometric context [5]), where pixels with higher probability of being part of an object instead of the wall or the floor are colored gray. Fitting a single 3D block to this feature map will be hampered by the confusing evidence, whereas a more articulated table model with legs and top explains the classification results for pixels between the legs of the table. Second, even if a single-bounding-box representation succeeded in discovering the presence of an object in the image, the parameters of a single fitted block have only modest power to distinguish objects. We previously showed that it is possible to classify furniture objects based only on 3D bounding box dimensions [10], but with much confusion when objects are similar in size. Having a class-dependent topological structure should help resolve such ambiguities, and in this case a composite table model is clearly a better fit than a simple box. Third, plain blocks cannot capture complex spatial configurations, and would not allow sliding chairs under the table (Figure 2, bottom row). Finally, a finer representation is also more useful for robot applications. For example, a small robot would be able to infer that there is a possible path between the table legs.

These observations lead us to propose a principled framework for modeling indoor scenes with representations for articulated objects, such as the table and the chairs in Figure 2. As in our previous work [10], we set out to simultaneously infer the 3D room box, the objects in it, their identity, and the camera parameters, all from a single image. Our first key contribution to this goal is to integrate **composite 3D geometry for objects**. Our results show that doing so improves both the global 3D reconstruction of the scene and object recognition. We also show that more accurate geometry both supports, and benefits from, higher level image features, such as pixel grouping based on appearance.

A second key contribution of this work is a **geometric representation based on re-usable parts**, from which we build more complex models. We designed data-driven inference for each of these parts. Importantly, inference strategies designed for a specific part are naturally shared by all the objects containing that part, and the modeler can create models using the available parts without having to worry about the inference. A third contribution is showing how to exploit **contextual relationships between objects** to help inference if there is significant occlusion or weak image evidence. For example, we show how to improve recognition of chairs by looking around tables. There is often little image evidence supporting the identification of a chair, perhaps just a leg or the top of a backrest (Figure 2, bottom right), but this can be addressed using top-down information, by looking for chairs in places that are likely based on the current model hypothesis.

Other related work. To our knowledge, the first attempt in this domain at using geometry other than blocks was by Hedau et al. [4], who used a plane to model backrests on top of blocks. However, if we exclude the backrest, their model still relies on the block representation, and they do not attempt to distinguish among object classes based on their geometry. Satkin et al. [12] then proposed to match full models of bedrooms and living rooms available from Google Warehouse, but they do not allow any variability in the size and the arrangement of the objects in the model. We also relate to recent work in object recognition that relies on modeling the 3D geometry of object categories [8, 16]. A first important difference is that our method understands objects in the context of the scene, with a likelihood function that evaluates the fit of the entire scene (Sec. 2.4), as opposed to having a different appearance model trained for each category. Second, we advocate a stronger 3D representation, where geometric variations within an object category are modeled in 3D, for example using priors on 3D size, instead of learning orientations and distances among the parts of an object in 2D [16]. Our 3D representation is also independent of the camera, hence we do not need to discretize the viewpoint and learn a different model per viewpoint [8, 16]. Lastly, this work is related to that of Schlecht and Barnard [13] on learning topologies of furniture from images by assembling re-usable parts.

2. Modeling indoor scenes

We use a Bayesian generative model, where we assume that images are generated by the projection of the 3D objects in the scene [10, 11]. We partition model parameters, θ, into scene parameters, s, encoding the 3D geometry, and camera parameters, c, modeling the perspective transformation. We define the posterior distribution as

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

where D are features detected on the image plane and $p(\theta)$ is the prior distribution over model parameters.

Scene parameters $s = (r, o_1, ..., o_n)$ include the room box and objects in it, where the number of objects n is not known a priori. We model the room as a right-angled parallelepiped [7, 3, 10, 11], defined in terms of its 3D center, width, height and length

$$r = (x_r, y_r, z_r, w_r, h_r, l_r, \gamma_r) \quad , \tag{2}$$

where γ_r is the amount of rotation around the room y axis (yaw) [11]. We model the imaging process with a standard perspective camera model

$$c = (f, \phi, \psi) \quad , \tag{3}$$

where f, ϕ and ψ are, respectively, the focal length, the pitch and the roll angle. Since absolute positions cannot be determined when reconstructing from single images, we arbitrarily position the camera at the origin of the world coordinate system, pointing down the z-axis. [10, 11]. Further, the extrinsic camera rotations (three degrees of freedom) are fully determined by ϕ, ψ and the yaw γ_r of the room.

A key contribution of this work is representing object models by assemblages of re-usable geometric primitives (parts), as opposed to simple bounding boxes. For example, Figure 3 shows a chair built by stacking a set of four symmetric legs, and an L-shaped structure. Here, we provide a generic formulation that is independent of the parts used, and in §2.3 we describe the parts used in our experiments.

Each object, o_i, is defined by its type, t_i (e.g., chair, couch, table), and its geometrical structure s_{ti}, which is a function of the object type:

$$o_i = (t_i, s_{ti}, x, y, z, w, h, l) \quad . \tag{4}$$

The last six parameters are the 3D center and size of a bounding box containing the entire object model. We define the size and position of object parts relative to the object bounding box, as one does not have access to absolute sizes when reconstructing from single images. We also assume that objects are aligned with the room walls [7, 11].

A model's structure is created by choosing from a set of available parts. We constrain the modeler to stack the parts vertically, although extensions are possible. Notationally,

$$s_{ti} = (p_1, ..., p_n, hr_1, ..., hr_n) \quad . \tag{5}$$

Here $(p_1, ..., p_n)$ is the collection of parts, which is fixed for each object class. Variable hr_i denotes the height of the ith part expressed as a ratio of the total object height, with $\sum_{i=1}^{n} hr_i = 1$. For example, the legs component in the chair model accounts for half of the chair height, implying $hr_1 = 0.5$ and $hr_2 = 0.5$ (Figure 3, bottom left). Parts are ordered vertically from bottom to top, and we will refer to the bottom one as the *support*. Each part p_i comprises a set of internal parameters $p_{\theta i}$, which are defined relatively to the bounding box occupied by that part. The height of a chair's seat is an example of an internal part parameter, and Figure 3 (bottom right) shows changing it while keeping the part bounding box fixed.

To summarize, an object is a vertical stack of parts. The object size and position in the room are specified by its bounding box, while part heights $(hr_1, ..., hr_n)$ determine bounding boxes for each part. Last, internal part parameters are defined relatively to these boxes. An advantage of this representation is that object parameters are subdivided into three sets, and this is is very convenient for inference (§3). Changes in the bounding box parameters propagate to all the parts (Figure 3, top right), changes in the part heights propagate to the parameters for the affected parts (Figure 3, bottom left), and changing the internal parameters results in changes local to the specific part (Figure 3, bottom right).

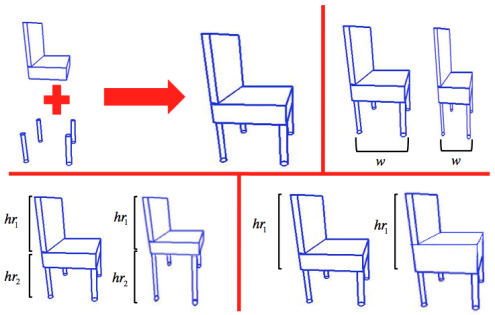

Figure 3. Top left: A chair is built by stacking two parts, a group of four legs and an L-shaped component. Top right: Changes in the object bounding box propagate to each part. Here the object width is divided by two, and this results in parts that are half as wide. Bottom left: Parts are stacked vertically, with their height defined as a ratio of the total object height (two different ratios shown). Bottom right: Changing the internal parameters of a part, here the L shaped one, while keeping the part bounding box fixed.

We impose two simple containment constraints: 1) objects must be entirely inside the room; and 2) objects cannot overlap. We emphasize that precise geometry enables configurations that bounding boxes would not, for example sliding a chair under a table (Figure 2). For efficiency, during inference we first check if the objects' bounding boxes collide, and only if that is the case do we check collisions using the geometry of the individual parts.

2.1. Prior distributions

Priors on the room box, objects, and camera parameters help constrain the search over parameter space, and also allow for recognition based on size and position [10]. Since absolute size and position cannot be inferred from single images, priors are defined in terms of size and position ratios. We extend this basic approach to composite object models that enable much better recognition.

Assuming independence among objects [10], we define

$$p(\theta) = \pi(c_h)\pi(r)\prod_{i=1}^{n}\pi(o_i) \quad , \tag{6}$$

where $\pi(c_h)$ is a prior on the camera height with a normal distribution with parameters (μ_h, σ_h) [10], and $\pi(r)$ is a relatively non-informative normally distributed prior on the room box parameters, parameterized by the ratio between room width and length r_1, and the ratio between the floor area and the room height r_2. Specifically,

$$\pi(r) = \mathcal{N}(r_1, \mu_{r1}, \sigma_{r1})\mathcal{N}(r_2, \mu_{r2}, \sigma_{r2}) \quad . \tag{7}$$

These parameters are learned from training images [10].

An object prior is defined over the size and position of its bounding box. We consider the ratios between

- height and largest dimension $o_{r1} = h/max(w, l)$
- width and length $o_{r2} = max(w, l)/min(w, l)$
- room height and object height $o_{r3} = h_r/h$

In our previous work [10], we showed how these quantities help distinguish between object classes. For example o_{r2} discriminates between roughly square furniture, such as chairs, and objects with a rectangular base, such as couches [10]. All these quantities are assumed to be normally distributed [10]. Finally, we also use a Bernoulli distributed binary variable d, encoding whether an object is against a wall [10] (e.g., usually true for beds). For frames we similarly encode the probability that they touch the floor. The full prior for an object is then

$$\pi(o) = \pi(d) \prod_{j=1}^{3} \mathcal{N}(o_{rj}, \mu_j, \sigma_j) \quad . \tag{8}$$

We set the parameters of object priors from text data available from online furniture and appliance catalogs [10].

2.2. Building object models

As part of this work, we implemented a modeling framework that allows any object assembled from the palette of geometric parts. To create a new object, the modeler specifies how these parts are arranged in the vertical stack, and provides parameters for the prior distributions as described above. The modeler also needs to provide the relative heights $(hr_1, ..., hr_n)$ and the internal parameters of each part $(p_{\theta 1}, ..., p_{\theta n})$. She can either provide a single value for each parameter, which will be kept fixed for each object of that category, or a valid range. In the latter case, we assume that each value in the range is equiprobable. Rather then include the internal parameters of an object as part of its prior, which leads to additional model selection problems, we simply set them as part of the model. In this work, we set values by manually fitting models to training images.

2.3. Designing object parts

We designed object parts to be modular so that they can be re-used in the modeling phase. We further design specific data-driven inference for each part, as we have found that dedicated inference, which takes advantage of part-specific characters, helps deal with the challenges of fitting complex geometry. The inference strategies defined for a part are shared by all objects using that part, and are transparent to the modeler. We emphasize that we only need to implement these once — i.e., the inference for the four legs of a table is the same module as for the four legs of a chair.

All parts used in this work are built from simple geometric primitives, such as blocks and cylinders, which can be easily rendered with OpenGL, as this is required to evaluate the likelihood function (described in the next section). Further, each part module has a collision detection mechanism to determine whether there is 3D overlap with any of the other objects in the scene.

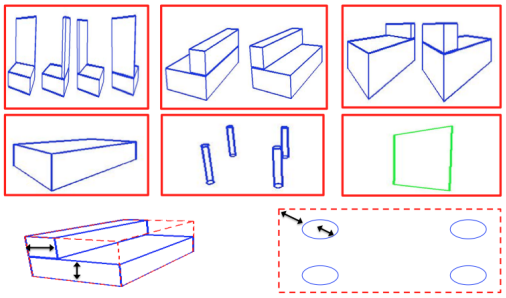

Figure 4. Top row: parts L1, L2, L3. These are distinguished by constraints on where the vertical "back" can be attached. L1 is completely free, and shows all four possibilities. L2 and L3 are for restricting to long side and short side respectively. Middle: a single block, a set of four symmetric legs, and a frame. Bottom: the free parameters of the L-shaped component and of the set of legs are shown by the double arrows.

We use three kinds of parts in this work (Figure 4): an L-shaped component built from two blocks, a set of four symmetric cylindrical legs, and a single block. The L component is parametrized in terms of the height of the horizontal block, and the width of the vertical block relative to the part bounding box. We use a third discrete variable to specify the side where the vertical block is located. Since we assume that all objects are aligned with the room walls, only four configurations are possible (Figure 4, top left). Within this context, we provide three kinds of L-shaped parts (Figure 4, top): L1, where the vertical block can be along any side, L2, where the vertical block is restricted to a long side of the horizontal block, and L3, where it is on a short side.

The set of cylindrical legs is parametrized in terms of the leg radius and the offset between the leg position and the corner, both of which are shared among all legs. Finally, the simple block part does not require any parameters, as we assume that the block is as big as the part bounding box, which is encoded at the next level up.

This modeling system, together with the modest set of parts, is able to capture a large number of configurations common in the base structure of much furniture. Having parts that capture some of the complexities of the objects, while inheriting their bounding box, simplifies the work of the modeler, and proves effective for inference (see Sec. 3).

In this work, we modeled 6 different furniture types: simple beds (a single block), beds with headrests (an L3 component), couches (an L2 component), tables (a stack of four legs and a single block for the top), chairs (a stack of four legs and an L1), and cabinets (a single block). Lastly, we use thin blocks attached to a room wall to model frame categories (Figure 4, middle right), which include doors, picture frames and windows [10].

2.4. The likelihood function

The likelihood function measures how well the detected image features D match the features predicted by the current model hypothesis θ. This function includes three different components that have been proved useful in this domain: edges [10, 11], orientation surfaces [7, 10], and geometric context [2, 5, 7]. We also introduce a new component that evaluates the color grouping predicted by the model.

The edge likelihood [10] $p(E_d|E_\theta)$ has a factor for each edge point matched to a model edge, and factors for missing edge points, and noise (unmatched) edges. Matched edges contribute normal densities for the distance between edge location and the projected model edge, and for the angle between the edge direction and the projected model edge [10].

Orientation surfaces [7] are often used in indoor environments, where most pixels are generated by a plane aligned with one of three orthogonal directions, and we estimate which one using the approach by Lee et al. [7]. We approximate the orientation likelihood $p(O_d|\theta)$ as the fraction of pixels such that the orientation predicted by the model matches that estimated from the image data (O_d) [10]. Following Hedau et al. [2], we also consider geometric context labels, which estimate the geometric class of each pixel, choosing between: object, floor, ceiling, left, middle and right wall. This is done using a probabilistic classifier trained on a mixture of color and oriented gradient features. We use the code and the pre-trained classifier available online [2]. For each pixel p_k, this provides a probability distribution $gc_k = [gc_{k1}, ..., gc_{k6}]$ over the six classes. Given the label l predicted by the model for pixel p_k, we define $p(gc_k|p_k) = p(gc_k|p_k = l) = gc_l$, and

$$p(GC|\theta) = \frac{\sum_{p_k \in I} p(gc_k|p_k)}{size(I)} \quad , \tag{9}$$

where we average the contributions of all image pixels. Since the available classifier was trained against data where only furniture was labeled as objects, and not frames, we consider frames as part of the wall they are attached to, and not as objects when we evaluate on geometric context.

We also introduce a new component promoting that pixels from the same color distribution are grouped together (Figure 5). This is similar to evaluating the quality of the grouping provided by a **2D** segmentation algorithm, with the key difference that the grouping is provided top-down by the model hypothesis. Note that the possible grouping hypotheses are significantly constrained compared with segmenting an image without any such guidance. In this scope, detailed geometry and 3D reasoning play an important role, as shown in Figure 2, where structures with legs provide a much better grouping than a plain block. Further, the reasoner does not need to entertain arbitrary groupings, as it would if it were doing bottom up clustering.

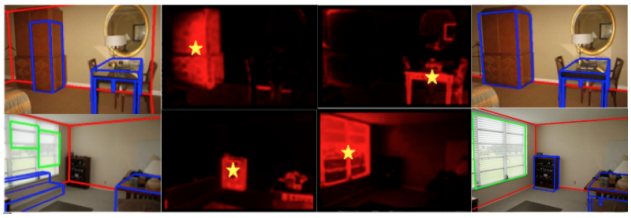

Figure 5. Our color likelihood encourages pixels similar in color to be grouped together. For each pixel, we compute a color histogram in LAB space. In columns 2 and 3, we select a pixel (marked with a yellow star) and compute the chi-square distance between its histogram and that of all other pixels. We show this in a heat map fashion, where we set the red channel proportionally to this distance. In the first row, pixels within the cabinet are very close in LAB space. We can see the benefits of using color by comparing the best fit found without using color (column 1) and with color (column 4). **Best viewed in color.**

We consider two pixels in the same group if they are both part of the projection of the same object, or of the same room surface, but we consider walls as a single group, as they tend to be of the same color. Intuitively, two pixels p_i and p_j have a high probability of being together if their distance d_{ij} in feature space is small. We use

$$p(p_i, p_j|\theta) \approx {d_{ij}}^{(1-I_g(p_i,p_j,\theta))} * (1 - d_{ij})^{\,I_g(p_i,p_j,\theta)} \,, \tag{10}$$

with $d_{ij} \in [0, 1]$ (see below), and $I_g(p_i, p_j, \theta)$ is an indicator function that takes value 1 if the model assigns the two pixels to the same group, 0 otherwise. Notice that we have a high $p(p_i, p_j|\theta)$ in two complimentary cases: 1) the two pixels are assigned to the same group and their distance is small, as we want groups to be perceptually uniform; and 2) the two pixels are in different groups and the distance is large, as groups (objects) tend to be different from each other. We measure the global quality of the grouping over the entire image by averaging the pairwise contributions

$$p(I|\theta) \approx \frac{\sum_{i=1}^{i \le N} \sum_{j=i}^{j \le N} g(p_i, p_j|m)}{(N^2 - N)/2} \tag{11}$$

where N is the number of pixels in the image.

This is a generic formulation of a grouping function, that allows for any choice of feature space. In this work, we experiment with color and use $d_{ij} = \chi(CH_i, CH_j)$, where $\chi(CH_i, CH_j)$ is the chi-square distance between the color histograms computed at pixels i and j over a window of size $n = 15$. We use histograms instead of simple pixel intensities because we want to capture the color distribution of objects and surfaces, which arise at a larger scale than the pixel level. We experimented with the LAB color space, and used a 3-dimensional histogram with 8 bins per dimension, where elements are softly assigned to bins. We denote the contribution of this color distance grouping function by $p(C|\theta)$. To reduce computation time, we only use the center pixels of 5-by-5 grid cells.

The four components are combined into

$$p(D|\theta) \approx p(E_d|E_\theta)p(O_d|\theta)^\alpha p(GC_d|\theta)^\beta p(C|\theta)^\delta. \quad (12)$$

As in our previous work [10], we set $\alpha = 6$. We set $\beta = 12$ and $\delta = 30$ by running our algorithm on the training portion of the Hedau dataset [2]. Here we used a coarse grid search over β and δ, with a step of 2, using the room box layout error (defined in Section §4) as an objective function. For black and white images, we used $\delta = 10$, as only a third of the information is available.

3. Inference

We use MCMC sampling to search the parameter space, defined by camera and room box parameters, the unknown number of objects, their type, and the parameters of each object and its parts. As in our previous work [10, 11], we combine two sampling methods—reversible jump Metropolis-Hastings for discrete parameters (how many objects, what type they are), and stochastic dynamics [9, 10] for continuous parameters (camera, room box, and object parameters). Proposals from these two samplings strategies are referred to as "jump" and "diffusion" moves [15].

Since indoor images satisfy the Manhattan world assumption where most surfaces are aligned with three orthogonal directions, we start by detecting a triplet of orthogonal vanishing points that we use to initialize the camera parameters. This has become a standard procedure in this domain [2, 7, 11, 10]. We initialize the parameters of the room box by generating candidates from orthogonal corners detected on the image plane [10]. We sample over the continuous parameters of each candidate and use the one with the best posterior to initialize the room box parameters. Then, we randomly alternate the following moves:

- sample over room box and camera parameters
- jump move: add/remove an object, change the category of an object
- pick a random object and sample over the parameters of its bounding box, or over $(hr_1, ..., hr_n)$ and $(p_{\theta 1}, ..., p_{\theta n})$. In the latter case, enforce $\sum_i hr_i = 1$, and that all parameters are within the allowed range.

This procedure is executed using twenty threads, where each thread executes the steps above. At the end, we allow thread to exchange the objects they have found, and we keep the best sample found across the threads [10]. The whole procedure takes on average 15 minutes per image.

Jump moves. All jump moves are accepted or rejected using the Metropolis Hastings acceptance formula. Switching the category of an object, causes two changes: 1) the prior distribution used to evaluate the object size and position; and 2) the geometric representation of the object. For example if a simple bed is turned into a couch, the bed "box" is replaced with an L component, by making sure that their bounding boxes coincide. To increase the acceptance ratio of jump moves, we propose objects from image corners in a data-driven fashion [11], and briefly sample its continuous parameters before evaluating Metropolis-Hastings (delayed acceptance [11]).

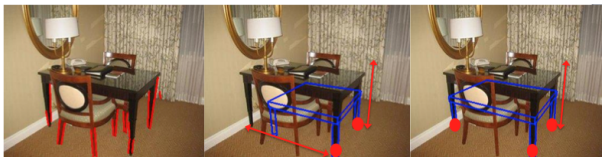

Figure 6. We use detected "pegs" to propose furniture with legs (left). Proposing a table from two pegs (middle) requires estimating the width/length and the height of the table, proposing it from three only leaves the height as a free parameter. **View in color.**

Part-specific inference. Efficient inference of complex structure, such as chairs with legs, seat and backrest, is more exacting than that of simple blocks. We designed specific inference moves for the different parts, which are re-used by all objects containing that part. While all objects share the data-driven proposal mechanism from corners, we use specific inference for the L-component and the set of four legs. For the former, we have to keep in mind that two to four configurations are possible (Figure 4, top row), and we try them all whenever a jump move involves an object containing an L component.

Legs are harder to identify, since they do not generate corners on the image plane that can be used for data-driven proposals. We thus detect peg structures, which are likely candidates for being legs (Figure 6, left), as suggested by Hedau et al. [4]. A peg can be used to propose a four-legged component the same way that a corner is used to propose a block. More effective proposals can be generated from two or even three of such pegs (Figure 6, center). We then use this proposal mechanism, as well as the standard ones, for all objects whose support is a set of four legs, namely tables and chairs. Note that this is different from Hedau's work [4] where objects are modeled with bounding boxes, and pegs are part of the likelihood, as a way to explain the missing edges between the legs of a table. Our likelihood does not need to explain that missing edge, since not finding it is predicted by the strong geometric model.

Using context. We found that several objects in indoor images are hard to detect because of clutter and heavy occlusions. Tables and chairs are an example, since they often occlude each other, like the chair behind the table in Figure 2. However, this problem can be addressed by considering contextual relationships between objects in a top-down fashion. Here, we bias the sampler to propose for chairs around detected tables, as shown in Figure 7. Given a table hypothesis (shown in blue), we look for chairs in the red areas in the Figure, whose size and position is defined relatively to the table, by making sure that the backrest of

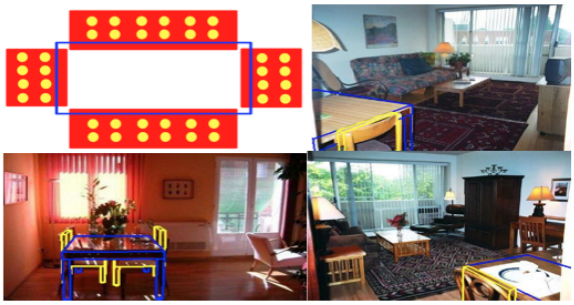

Figure 7. Using context to find chairs around tables. Given a table hypothesis (seen from above in blue, top left), we propose chairs around it. We consider the red areas around each side of the table, and propose a chair centered at each of the yellow dots shown. For each chair we then briefly sample over its continuous parameters, and accept it or reject it using the Metropolis-Hastings acceptance formula. Chairs found with the help of this procedure are shown above in yellow. **Best viewed in color.**

Table 1. Room layout error on Hedau and UCB datasets.

	no color	color	state-of-the-art
HEDAU	13.7	**12.7**	12.8 [14]
UCB	14.2	**14.0**	18.8 [10]

the chair is facing the table. This allowed us to find the chairs drawn in yellow in Figure 7, that were missed by inference without context cues. This strategy can easily be introduced in other cases where contextual cues are strong hints of where to look to make inference more efficient.

4. Results and discussion

All our experiments were performed on the Hedau dataset [2] (104 color images) and the UCB dataset [17] (340 black and white images). We first evaluate the quality of the room box estimation [2, 7, 10, 11], by comparing the projection of the estimated room against the ground truth, where each pixel was labeled according to the surface of the room box it belongs to. The score is computed as the ratio between the pixels correctly labeled and the total number of pixels, averaged over the entire dataset. Results in Table 1 show the benefits of using color, which increased performance on the two standard data sets. With it, we were able to exceed available state-of-the-art values.

We then evaluate object recognition, which is more indicative of our goal of full scene understanding. We are trying to identify eight object classes that belong to two very distinct categories: frames (doors, windows and picture frames), and furniture (beds, cabinets, chairs, couches and tables). We first measure how many objects we correctly identified for each of the two main categories, even if there is confusion within the subcategories (e.g. when we label a table as a couch, or a window as a door) [10]. We provide precision and recall scores based on this criterion.

Second, we measure the accuracy we achieved within each of the two categories, as the percentage of objects that were assigned to the correct subcategory. To decide whether an object was correctly identified, we measure the intersection between the projection of the estimated object and its ground truth position [10]. If the intersection is larger than 50% of the union of this two areas, we consider the object as a correct detection. The ground truth masks used for these experiments are available on our website[1].

Table 2. Precision, recall, and subcategory classification accuracy on the Hedau (left) and UCB (right) datasets.

Furniture	p	r	sc	p	r	sc
no color	50.4	25.8	49.3	35.9	28.6	47.5
color	**53.9**	**35.7**	**57.3**	**38.9**	**32.0**	**52.5**
Del Pero [10]	32.5	20.3	50.0	31.0	20.1	38.0
With chairs	p	r	sc	p	r	sc
no context	53.8	26.2	58.6	37.8	22.0	52.5
context	**54.9**	**28.3**	**61.3**	**38.1**	**22.2**	**53.4**
Frames	p	r	sc	p	r	sc
no color	36.2	33.7	69.6	27.6	37.4	63.3
color	**44.9**	**41.8**	69.3	**33.3**	**42.4**	**63.6**
Del Pero [10]	33.1	18.7	**70.3**	27.7	19.7	60.0

We first compare with our previous results on object recognition [10], where the same furniture and frames categories are used, except for chairs. For proper comparison, we do not include chairs when computing precision and recall, and evaluate with chairs separately. Also, we consider beds with headrest and beds without headrest as both part of the category "bed". Table 2 shows that we improve on all measures. We also report the benefits of using color, which are less evident on the black and white dataset.

In general, there is a trend showing a better precision for furniture with respect to frames. We explain this difference by considering that frames are supported by edges and color only, whereas furniture is detected using a more robust set of features including geometric context and orientation maps. Detailed geometry also allows us to improve on subcategory classification for furniture, as precise topology is a strong hint for distinguishing among categories such as couches and tables. Our color model improves precision and recall for both furniture and frames, as it helps segment objects from the background and from each other. For furniture, color also improves subcategory recognition indirectly by improving object geometry fitting. However, in the case of frames, the geometry is the same for all three types, so while global precision and recall are improved with color, distinguishing among the sub-types is not.

When we also consider chairs, precision and subcategory classification improve, despite the task being harder due to a larger number of categories (compare row two and row five in Table 2). However, recall suffers, as chairs are relatively small and often heavily occluded. Nonetheless, we notice

[1] http://kobus.ca/research/data/CVPR_13_room

the benefits of using context for proposing, which improves all measures, and is a promising step towards dealing with heavy occlusion and scarce image evidence using top-down information. In the case of the Hedau dataset, context allowed us to identify seven more chairs at the cost of one false positive. Qualitative results on using context to find chairs are shown in Figure 7, while full scene reconstructions are shown in Figure 1, 2 (bottom right), and 8, which also includes some typical failures.

Discussion. The experimental results confirmed that the proposed 3D representation indeed has advantages. A very important one is that variation within instances of an object category is reduced because the camera does not contribute to it, and also defining parts relatively to an object's size instead of absolute values further reduces the variability among classes. Consider for example the table model, where we do not impose tables to be any particular height, but the relative amount for the legs part versus the top part is kept within a small learned range (roughly 92% for the legs). Interestingly, we found that most part parameters, such as the leg width ratio, tend to have little variability. In fact, despite keeping them within a small range, learned from a small amount of training data, we could detect a variety of tables in the test data, ranging from small coffee tables (Fig. 8, bottom left) to dining tables (Fig. 7, bottom left). Additionally, since we encode the key structure of an object, minor variations in the object parts do not necessarily create problems. For example, in Fig. 5 a table is detected even if the predicted top is too thick (bottom right).

The experiments also showed that the proposed inference can handle complex 3D models, which introduce a larger (and unknown) number of variables, without being too sensitive to local optima. This is enabled by the fact that objects only interact with others in minimal ways via occlusion and space occupancy constraints. Hence proposing a complex alternative to a bounding box is like an independent local part of the inference, unless it changes what is occluded with what, like switching a block into a table so that chairs can be tucked underneath. However, truly complex objects, such as an exuberant indoor plant, will require additional and potentially quite different approaches.

5. Acknowledgments

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

References

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

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

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

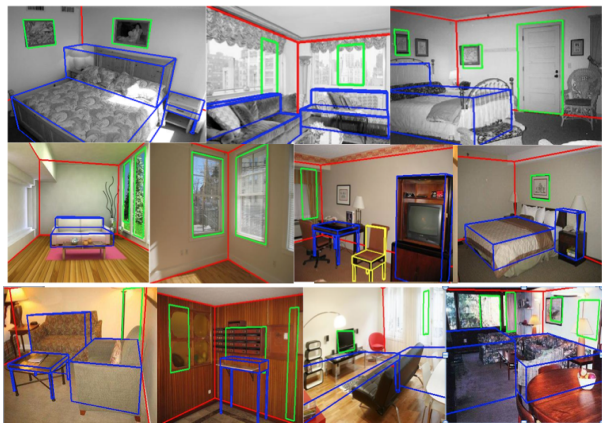

Figure 8. Scene reconstructions (top rows) and failures (bottom row). As shown from left to right in the bottom row, typical failures are due to: 1) confusion between object categories (two couches confused for cabinets), 2) hallucinating objects (the table "latched" to the texture of the wall and the shelf), 3) a poor camera estimate from which the algorithm could not recover, 4) poor fits due to errors in the feature detection process, which mostly occur in blurry images. **Best viewed in color.**

[4] V. Hedau, D. Hoiem, and D. Forsyth. Recovering free space of indoor scenes from a single image. In *CVPR*, 2012.

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

[6] K. Karsch, V. Hedau, D. Forsyth, and D. Hoiem. Rendering synthetic objects into legacy photographs. In *SIGGRAPH Asia*, 2011.

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

[8] J. Liebelt and C. Schmid. Multi-view object class detection with a 3d geometric model. In *CVPR*, 2010.

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

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

[11] L. D. Pero, J. Guan, E. Brau, J. Schlecht, and K. Barnard. Sampling bedrooms. In *CVPR*, 2011.

[12] S. Satkin, J. Lin, and M. Hebert. Data-driven scene understanding from 3d models. In *BMVC*, 2012.

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

[14] A. Schwing, T. Hazan, M. Pollefeys, and U. R. Efficient structure prediction with latent variables for general graphics models. In *CVPR*, 2012.

[15] Z. Tu and S.-C. Zhu. Image segmentation by data-driven markov chain monte-carlo. *IEEE Trans. Patt. Analy. Mach. Intell.*, 24(5):657–673, 2002.

[16] Y. Xiang and S. Savarese. Estimating the aspect layout of object categories. In *CVPR*, pages 3410–3417, 2012.

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