Sampling Bedrooms

Luca Del Pero[†] Jinyan Guan[†] Ernesto Brau[†] Joseph Schlecht[‡] Kobus Barnard[†]

[†]University of Arizona
Tucson, Arizona
{delpero, jguan1, ernesto, kobus}@cs.arizona.edu

[‡]University of Heidelberg
Heidelberg, Germany
schlecht@uni-heidelberg.de

Abstract

*We propose a top down approach for understanding indoor scenes such as bedrooms and living rooms. These environments typically have the Manhattan world property that many surfaces are parallel to three principle ones. Further, the 3D geometry of the room and objects within it can largely be approximated by non overlapping simple structures such as single blocks (*e.g. *the room boundary), thin blocks (*e.g. *picture frames), and objects that are well modeled by single blocks (*e.g. *simple beds). We separately model the 3D geometry, the imaging process (camera parameters), and edge likelihood, to provide a generative statistical model for image data.*

We fit this model using data driven MCMC sampling. We combine reversible jump Metropolis Hastings samples for discrete changes in the model such as the number of blocks, and stochastic dynamics to estimate continuous parameter values in a particular parameter space that includes block positions, block sizes, and camera parameters.

We tested our approach on two datasets using room box pixel orientation. Despite using only bounding box geometry and, in particular, not training on appearance, our method achieves results approaching those of others. We also introduce a new evaluation method for this domain based on ground truth camera parameters, which we found to be more sensitive to the task of understanding scene geometry.

1. Introduction

We propose a top down approach for understanding indoor scenes such as bedrooms and living rooms. Recent interest in this domain has led to approaches for determining the orientation of the main surfaces [6, 7, 18, 27, 29], the room box in the presence of clutter [12], and generic objects within using learned appearance models and geometry [13]. A key motivation is to understand the 3D geometry of rooms to help identify objects and their locations. We have essentially the same goal, but we advocate even a more top-down approach and a more unified representation. For example, we agree with Hedau *et al*. [12] that clutter in rooms makes finding the room box difficult. However, instead of thinking of the clutter as confounds, we would like to directly fit objects to it. Doing so simultaneously achieves understanding the inside more fully, and allows these two processes to help each other.

In this paper, we propose a simple generative statistical framework for modeling rooms and simple objects within them, and a comprehensive inference approach to understand room scenes based on that model. These environments typically have the Manhattan world property [5] that many surfaces are parallel to three principle ones. Further, the 3D geometry of the room and objects within it can largely be approximated by non overlapping simple structures such as single blocks (*e.g*. the room boundary), thin blocks (*e.g*. picture frames), objects that are well modeled by single blocks (*e.g*. simple beds). We impose further structure by introducing the notion of a room block type. The statistics of where blocks are located within room blocks is conditioned on the room block. For example, frames are constrained to lie on room surfaces, and objects are constrained to lie on the floor and are more likely to also be positioned against a wall. Objects can be modeled as simple blocks, either as a good approximation of them, or as a bounding box for a more detailed model. Our approach easily extends to objects modeled as collections of contiguous blocks as suggested by Schlecht and Barnard [23]. In this case the block locations are conditioned on the object position, and the statistics of the component block sizes and configuration are conditioned on the model of the category. We develop the room model in detail in §2.

Manhattan room images come from Manhattan rooms with a camera constrained to be within the room. A model hypothesis then explains the edges in the images based on the projection of the model using the hypothesized camera. Objects within the room are considered opaque, so that they can explain missing edge elements from the edges they oc-

clude. We develop the likelihood function in detail in §2.2.

To fit our model to room images we use MCMC sampling. Sampling qualitatively different structures typically leads to a change in model dimensionality (*e.g.* adding a block). For such "jump" proposals we use reversible jump Metropolis Hastings [3, 8] which defaults to standard Metropolis Hastings [3, 19] if the dimensionality does not change. To make these proposals more efficient we rely on data driven methods [32]. In particular, we integrate the method of Shi and Lui [24] to suggest 3D corners from detected 2D ones. To search efficiently over regions of continuous parameters space, we use stochastic dynamics [21]. Examples of such "diffusion" proposals include changing the camera parameters and adjusting the size of a block. Details of our sampling approach are in §3.

Other related work. In addition to the room understanding papers already mentioned above, this work also relates more broadly to recent interest in understanding and exploiting geometry in scenes (*e.g.* [14, 15, 22]). Using sampling to understand scenes as a collection of primitives relates to work on image parsing [10, 25]. In Han and Zhu [10] the prevalence of Manhattan scene surfaces was modeled by planar rectangles under possible perspective distortion. Our model can be interpreted as a grammar model for a 3D scene that is projected with a camera model that relates to recent interest in representing images using grammars (*e.g.* [30, 31], although these alternative approaches focus on image features directly, not 3D world or object characteristics as we do here. Finally, this work also relates to much earlier work on fitting 3D models of objects to images (*e.g.* [16, 20]).

Reproducibility. The two image datasets were collected by others and are available on-line [12, 28], as is our ground truth for the UCB data set developed for this work [2]. We have also provided an executable program and input scripts that reproduce our results [1].

2. A generative model for room images

We denote the model parameters by $\theta = (r, c)$, where r is the set of room parameters and c the set of camera parameters. We model the room as a box (right parallelepiped) r_b (representing walls, ceiling and floor) containing n objects

$$r = (r_b, n, o_1, ..., o_n) \quad , \tag{1}$$

where n is not known a priori. We constrain the floor to be parallel to the x-z plane of the world reference frame, and allow the room box to rotate around its vertical axis. This leads to the following parametrization

$$r_b = (x_r, y_b, z_b, w_b, h_b, l_b, \gamma) \quad , \tag{2}$$

where x_b, y_b, z_b are the coordinates of the room centre in 3D space, w_b, h_b, l_b are respectively its width, height and length, and γ is the rotation angle around the room vertical axis. Objects in the room are similarly modeled by blocks

$$o_i = (x_i, y_i, z_i, w_i, h_i, l_i) \quad . \tag{3}$$

For example, a single block lying on the floor could approximate a simple bed or a cabinet, or provide a bounding box for a more complex object, such as a table. Windows, doors and pictures are approximated with thin blocks (frames) attached to a wall. All these entities share the same orientation γ of the room block, under the Manhattan world assumption [5] that most planes are aligned with the three main world axes. Finally, objects have to be fully contained in the room, and they can not intersect each other.

2.1. The camera model

We use a standard perspective camera model defined in terms of extrinsic and intrinsic parameters. Since it is not possible to reconstruct the absolute position and size of a scene from a single image, we arbitrarily keep the camera at the world origin, and we infer the scene up to a scale factor. Intuitively, we keep the camera at a fixed location and let the room translate, expand and rotate around its vertical axis (Figure 1, first column). It is then not necessary to have a parameter for the height of the camera from the floor, as this is determined by the distance between the camera centre (fixed) and the room floor, which can move in space.

The orientation of the camera coordinate system is defined in terms of rotation angles around the camera z axis (roll angle ψ) and x axis (pitch angle ϕ), as shown in Figure 1, second and third column. The yaw is not relevant as it is fully determined by the rotation angle γ of the room. The amount of perspective distortion is determined by the focal length f. Finally, we assume that there is no skew, that the aspect ratio is unity, and that the principal point is in the centre of the image. The camera model, c, is then fully specified by the parameters

$$c = (\psi, \phi, f) \quad . \tag{4}$$

The focal length has to be positive, and we constrain pitch and roll to fall within ranges of plausible values for indoor scenes ($\phi \in [-60°, 60°]$), $\psi \in [-10°, 10°]$). Notice that the camera points in the direction defined by the positive z axis when $\phi = 0$.

2.2. The image model

Our image model is similar to the one used by Schlecht and Barnard [23], where a 3D model hypothesis is projected into the image plane using the camera hypothesis, and the projected model generates image features $F = (f_1, ..., f_s)$. Here we limit our features to the set of edge points $E = f_1$. To compute the projection of the model into the image plane we take advantage of graphics hardware and software using offscreen rendering. We attach detected image edge

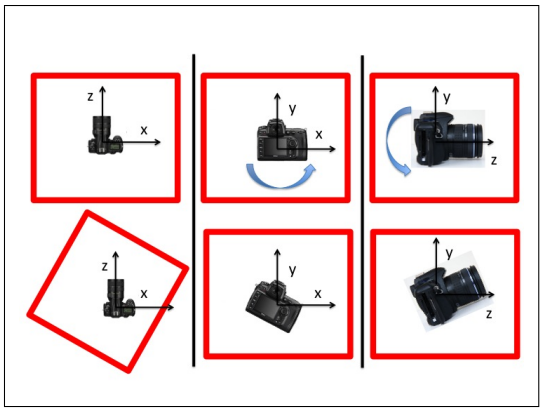

Figure 1. The camera extrinsic parameters define the position and orientation of the camera with respect to the world reference frame. When reconstructing from single images, absolute sizes and position cannot be determined, and we can choose to arbitrarily position the camera at the origin of the world coordinate system. As illustrated in the left column, the room box (in red) can move and also rotate around its y-axis which determines the yaw of the camera (see text). Two more angles, a rotation around its z-axis (roll, mid column) and a rotation about the x-axis (pitch, right column) complete the camera orientation specification.

points to projected model edge points using greedy assignment based on both edge distance and edge angle (see (6)). Notationally, $E = \{E_j, E_{noise}\}$, where j indexes projected model edges.

Given this correspondence we assume that edge points are conditionally independent given the model. Following Schlecht and Barnard [23], this leads to the likelihood function

$$p(E|\theta) \approx e_{noise}^{N_{noise}} e_{miss}^{N_{miss}} \prod_j \prod_k e(x_{jk}) \quad , \qquad (5)$$

where θ is defined above, e_{miss} is the probability of a projected point of the scene model not being matched to a detected edge in the scene, and e_{noise} is the probability density of a detected edge point not being matched to any model point. N_{noise} and N_{miss} are respectively the number of unmatched detected edge points and the number of unmatched projected model points.

The probability density of matching a detected edge point x_k to model edge E_j is given by

$$e(x_{jk}) = \mathcal{N}(d_{jk}, 0, \sigma_d)\mathcal{N}(\phi_{jk}, 0, \sigma_\phi) \quad , \qquad (6)$$

where d_{jk} is the distance between x_k and the projected model point m_j. This distance is computed along the direction of the gradient at m_j while setting up the correspondence. ϕ_{jk} is the difference in orientation between the detected edge and the corresponding projected model edge. In this work we use $\sigma_d = 20$ pixels and $\sigma_\phi = 0.5$ radians. The likelihood is maximized when most of model edge points are matched to a detected edge point (small N_{miss}), few detected edge points are unmatched (small N_{noise}) and each model edge k is well aligned with the corresponding detected edge j (*i.e.* $\phi_{kj} \approx 0$, $d_{kj} \approx 0$).

3. Inference

To find good parameters for the observed image, we sample from the posterior distribution

$$p(\theta|E) \propto p(E|\theta)p(\theta) \quad , \qquad (7)$$

where $p(E|\theta)$ is the likelihood function defined in equation (5), and $p(\theta)$ is the prior over the model parameters.

Our sampling space is defined over all camera and scene parameters. Sampling moves are selected at random and fall into two categories: "jump" moves (§3.2), which change the discrete structure of the model by adding and removing blocks from the scene, and "diffusion" moves" (§3.1) that allow efficient sampling within the parameter space for a particular structure.

3.1. Diffusion sampling using stochastic dynamics

Diffusion moves are used for continuous changes in a parameter space. We sample over phase space by following Hamiltonian dynamics using Neal's formulation [21]. Our energy function $H(\theta)$ is defined in terms of the joint distribution of the parameters and the image features $p(\theta, F)$:

$$H(\theta) = -log(p(F|\theta) - log(p(\theta)) \quad . \qquad (8)$$

We follow the dynamics with leapfrog discretization, and compute the derivative of the potential energy with numerical approximation, which is the computational bottleneck.

We use the following diffusion moves, which follow the dynamics over a subset of parameters, and are executed in a random order:

- Sample over room box parameters. This move translates, expands and rotates the room box around its y axis. Objects in the room are attached to the floor or to the walls and thus follow the room as it moves
- Sample over the size and position in the room of an object. This slides and/or expands an object on the wall or floor it is attached to.
- Stretch an object or room box. In this alternative way to change object size, we sample over one dimension of the object by keeping the position of one of its faces fixed. This solves a sampling problem illustrated in Figure 2.
- Sample over camera parameters except focal length. We often sample these parameters together with the room box parameters.
- Sample over focal length and scene scale. Changing the focal length alone causes objects to shrink or expand in the image plane. This causes a dramatic

Figure 2. From an upper right image corner (in blue) we proposed a door (in green) with a random height and width (left image). To get the correct fit, the door must be stretched and its center (in red) must be moved down (right image). In order to do this efficiently, we introduce diffusion moves that vary the dimension of an object by keeping the position of one edge fixed. In this example, when this move is chosen for the top edge, the sampler finds the right solution very rapidly. **Best viewed in color.**

change in the likelihood for the proposal, and the move is rarely accepted. Hence, we change the focal length while keeping the ratio between the scene scale and the focal length constant.

While executing any of these moves, we enforce that every object is completely inside the room, objects do not intersect, and that the camera is contained inside the room box at all times. When we detect that a move would cause an object to be partly outside the room, either we shrink the object or allow the room to expand.

3.2. Reversible jump sampling for jump moves

We need jump moves to change the number of objects in the room, which is not known a priori. In particular, jump moves can add, remove, or replace objects and frames, or propose a different room box to start fresh. For such moves we use the reversible jump modification to the standard Metropolis Hastings acceptance formula [8, 9]. Since the sampling space is so large, naive jump proposals (*e.g.* samples from the prior) are unlikely to be accepted, leading to unacceptably long running times. Hence we use data driven sampling [26] that conditions the sampling on the data, and allows fast bottom-up processing to ensure that samples have some evidence in the data.

To support data driven sampling, we detect the vanishing points (§3.3) which provide a good estimate of the focal length, and also allows us to assign line segments to one of the three main orthogonal directions defining the Manhattan world (Figure 3, top right). We then detect corners on the image plane by finding the intersections of line segments converging to different vanishing points (Figure 3, top right). Such corners are likely to be generated by the projection of an orthogonal corner in the 3D world. Given the position and orientation of one such corner on the image plane and a reasonable estimate of the focal length, we are able to propose blocks in 3D which are accepted with high probability (§3.4). The same strategy is also used to propose the position and the orientation of the room box (Figure 3, bottom right).

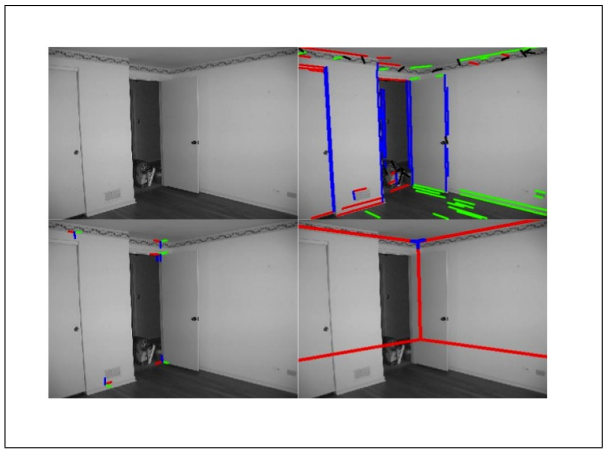

Figure 3. We use detected image corners to propose blocks that will be accepted with a high probability. We detect a triplet of orthogonal vanishing points from the original image (top left), and use them to assign detected line segments to one of the three main orthogonal directions defining the Manhattan world (top right). Each line is draw with a different color according to the vanishing point it converges to, with black lines being outliers. Intersections of three line segments each converging to a different vanishing point are generated by 3D orthogonal corners (*i.e.* a block corner or a room box corner) projected onto the image plane (bottom left). We can use one of such corners to propose the position and orientation of a block in 3D [24]. This is illustrated in the bottom right image, where we used a corner (in blue) to propose a block with random size, whose projection onto the image plane is rendered in red. **Best viewed in color.**

3.3. Vanishing points and corner detection

We detect edge points using a Canny edge detector [4], link them into edge chains, and fit line segments to them. Then, we detect vanishing points using the algorithm proposed by Lee et al [18]. Here, we do not refine the position of the vanishing points using non linear optimization as they do, since the diffusion sampling naturally does a similar thing, but relative to our likelihood function. We compute an estimate of the focal length using the Choleski decomposition of the absolute conic matrix, which can be easily retrieved from the position of the three vanishing points [11].

Given the vanishing points, line segments in the scene can be assigned to one of the three main orthogonal directions (Figure 3, top right). The intersection of three lines, such that each of them converges to a different vanishing point, are likely to be the projection of a 3D corner onto the image plane (Figure 3, bottom left).

3.4. Proposing a 3D orthogonal corner from a 2D corner and the focal length

We use the method of Shi and Liu [24] to propose the camera parameters and a 3D orthogonal corner given the 2D projection of the corner onto the image plane and the focal length. Consider the case where we want to propose the room box from a corner (*e.g*. the blue corner shown in Figure 3, bottom right). In this case, we can position the camera at the world origin, aligned with the world axes. By fixing the focal length, which we have estimated from the vanishing points, we can cast a ray between the camera centre and the 2D corner position in the image plane, and we can cast three similar rays from the camera centre and a point lying on each of the 2D directions of the corner. By fixing the distance between the camera and the 3D corner, which we can choose arbitrarily, these four rays and the focal length fully determine the coordinates of the corner in 3D and its orientations [24]. We can then hypothesize the position and orientation of a corner of the room box. Finally, we randomly select the room box size, ensuring that the camera is contained within it. The red lines in Figure 3, bottom right, are the projection of the room box proposed from the corner shown in blue on the image.

We use simple geometric operations to transform this box to a coordinate system such that the floor lies on the XZ plane and the camera is at the world origin, as required by our model. The same procedure can be used to propose blocks inside the room. In this case, we make sure that the object is contained inside the room box, expanding the latter if necessary. Notice that the procedure above can be applied also to corners found by intersecting only two lines converging to different vanishing point, by augmenting the corner with a line converging to the third vanishing point.

3.5. Delayed acceptance

In order to further increase the acceptance probability for a jump move, we use delayed acceptance. More specifically, once we propose a block, it is likely to have the wrong size, since we randomly choose its dimensions. To overcome this impediment, we allow diffusion sampling shrink or stretch the object for some time, such that its projection moves towards image edges, thus increasing the likelihood value for this proposal. Only then do we decide whether to accept or reject. This is particularly useful when proposing to replace a block already providing a good fit with a different one. Notice that this expedient is acceptable because we are using sampling only for optimization and not integration.

4. Experiments

We performed most of our experiments on the UC Berkeley room dataset, which consists of 340 images of bedrooms, kitchens, living rooms corridors, etc., under a wide variety of camera parameters. We evaluate our ability to fit the room model by comparing against:

- Ground truth box layout. For each image we manually labeled each pixel according to the room face it belongs to (*i.e*. 1= ceiling, 2= floor, 3 = right wall,etc.). We compare the projection of the room box estimated by our algorithm against these manual labels, measuring the percentage of pixels that were classified correctly. This is a standard measure that has been used in previous work in this domain [12, 27]
- Ground truth camera parameters, prepared using the procedure discussed in Section 4.1. Estimating correct camera parameters is an important indicator of scene understanding, but it is not commonly tested against. Note that a good result on the room box pixel orientation can be achieved when the camera parameters are very different from the correct ones (see Figure 4)

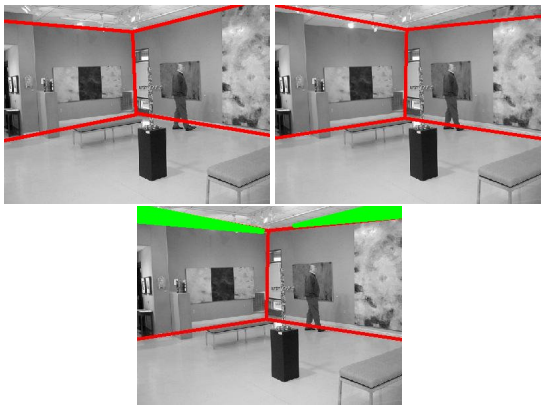

Figure 4. The correct room box (in red) was rendered on the top left image using the ground truth camera parameters. Consider now the parameters used to render the room box on the top right image. When comparing against the room box ground truth labels (see text), this hypothesis would get a very high score, since only the green regions shown in the bottom image would be labeled as error. However, the camera parameters are wrong, as shown by the red edges delimiting the room ceiling on the top right image. **Best viewed in color.**

4.1. Ground truth camera parameters

We determine the ground truth camera parameters for an image using a semi-automated procedure. First, we manually draw the room box onto the image plane, as illustrated in Figure 5. We then use the diffusion moves described in Section 3.1 to fit a camera and a room box to the edges manually drawn on the image. We obtain the exact solution in nearly all images by running the diffusion moves for a few minutes. We manually checked each result, and in a few cases we had to adjust the camera parameters further.

This procedure is equivalent to calibrating a camera

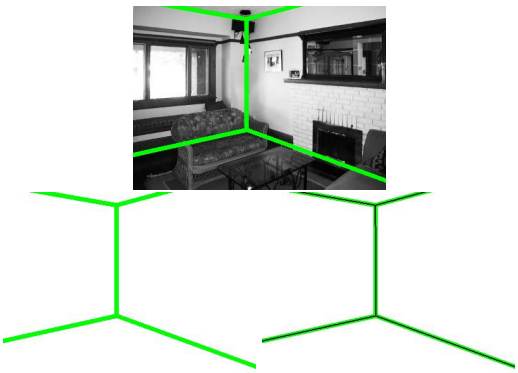

Figure 5. Determining ground truth camera parameters from an image. We manually draw the room box edges (in green) onto the image plane (top), producing the image shown in the bottom left. We fit a room box and a camera to this image by running the sampling diffusion moves (Section 3.1) for a few minutes. This procedure finds the exact solution, which is rendered in black on the bottom right image. We can see that the projection of the model correctly matches the image edges. We use the camera parameters found in this way as ground truth. **Best viewed in color.**

given a set of correspondences between 3D lines and the corresponding projected lines onto the image plane. Normally, six of such correspondences are needed, provided that the directions of the 3D lines are all linearly independent. Since we know that the edges onto the image plane all come from the projection of a right-angled parallelepiped, and given our assumptions on the camera parameters, we only need 4 of such lines to determine the focal length. It follows that this approach cannot work when only three edges of the room box are visible. Hence, we do not evaluate on camera parameters in these cases (just room box labels).

The procedure just described also allows to estimate the ground truth pitch and roll of the camera, in a coordinate system where the room floor is parallel to the XZ plane and the camera centre is at the world origin, as required by our model. However, we cannot compare against the room size and position estimated, since reconstruction from single images is possible only up to a scale factor.

4.2. Results

We started with a very simple experiment where we proposed a room box and camera parameters from every detected image corner. We further evaluated the proposals with the highest likelihood value by running the diffusion moves for a few thousand iterations. The diffusion sampling allows to fix the size and position of the room and the camera parameter. The whole process takes around 10 seconds per image on a machine with a fast graphics card (no noticeable improvement was detected when running the algorithm for a longer time). As a second experiment, we ran

Figure 6. Examples of the estimated room model backprojected onto the original image under the estimated camera parameters. The room box is rendered in red, objects in blue, and frames in green. **Best viewed in color.**

the full algorithm by allowing a maximum of three objects per room, with a running time of ten minutes on the same machine as before.

We first evaluated against the ground truth room box labels and camera parameters on the UC Berkeley dataset (340 images). We show qualitative results in Figure 6, where we have good fits of the room box, and in Figure 7, where we successfully detected several objects and frames.

The quantitative results in Table 1 show that adding blocks results in a substantial improvement in the focal length estimation. Proposing blocks adds more edges to the scene, and this provides more evidence to what the correct foreshortening is, since all blocks share the same orientation. An example is shown in the top row of Figure 8.

Adding blocks also reduces the error on the room box layout estimation (Figure 8, bottom row). This is even more evident on the Hedau dataset [12], shown in Table 2. (We only test on pixel orientation, as we do not have camera parameters for this data). Our results approach that of Hedau

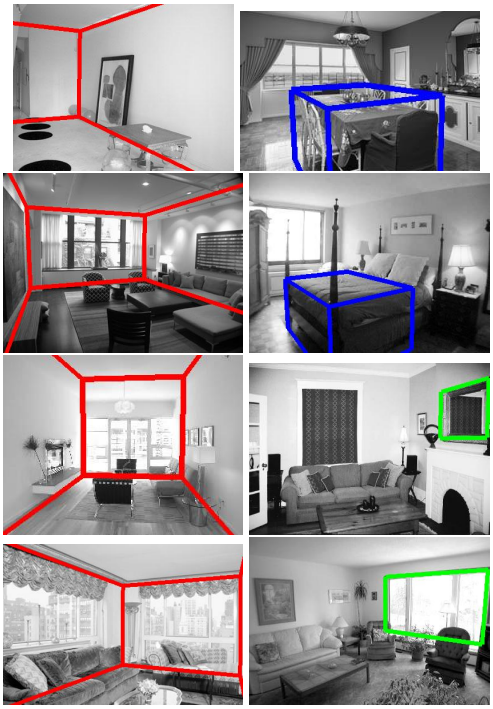

Figure 7. Additional results . In the left column we rendered only the room box (in red) without the detected objects, in the right column we can see some promising detected blocks (blue) and frames (green). **Best viewed in color.**

Experiment	box error	f length	pitch	roll
No objects	24.8%	93 ± 7	3.8 ± 0.2	4.3 ± 0.7
3 objects	24.0%	75 ± 6	3.8 ± 0.2	4.0 ± 0.8

Table 1. Error for room box layout and camera parameters estimation on the UC Berkeley dataset. The focal length error is in pixel units, pitch and roll are in degrees

	No blocks	3 blocks
Box error	30.2%	26.8%

Table 2. Error for room box layout estimation on the Hedau dataset

et al. [12] and Wang *et al*. [27], who respectively report errors of 21.5% and 20.1% on this dataset. To compare properly, the results in Table 2 are for the 105 images used for testing in those works, despite the fact that our method does not use training data. In contrast, these alternative methods use sophisticated models for clutter appearance that is trained on the rest of the data (200 images).

5. Conclusions

We have developed a generative modeling framework for understanding Manhattan rooms and an efficient method for simultaneously fitting the camera and the room model of unknown structure and dimension to image data. The method achieves comparable results to others without any use of appearance models learned from data on the task of identifying surface labels.

We have also tested our method against ground truth camera parameters which confirm our suspicion that surface label performance is a relatively insensitive measure. By identifying some objects, we are able to improve the camera parameter measure without significant difference in surface labels. While detection of frames and objects remains difficult, the camera parameter improvement, together with visual inspection, suggest an improvement in scene understanding. We have put our ground truth on-line [2], and we

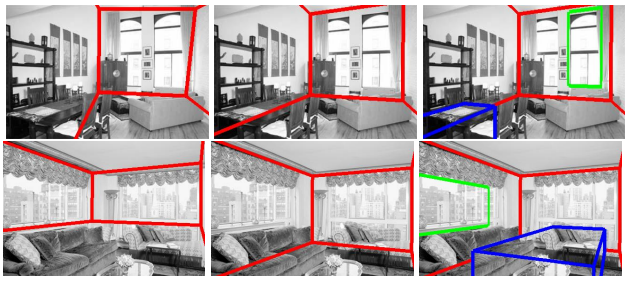

Figure 8. Adding blocks in the room produces a better reconstruction. Here we show the room box found by fitting the room box alone (left), the room box found when adding objects (middle), and the full reconstruction (right). In the top row, adding blocks drives the sampler towards better camera paramaters. In the bottom row, the blue block helps explaining occlusions and forces the room box to expand, even if the blue block does not provide a good fit for an object in the room. **Best viewed in color.**

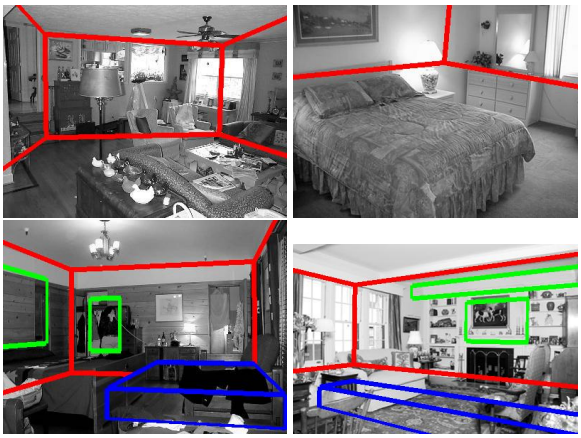

Figure 9. Limitations of our algorithm. Top left: when the error in the estimates of the vanishing points is large, blocks proposals are completely wrong and our algorithm often provides the wrong solution. Top right: when we do not add objects to the scene, room box edges that are completely occluded cannot be explained by our model. Bottom left: our likelihood function is edge based, and objects that explain image edges very well might not correspond to actual pieces of furniture in the room. Bottom right: we only use a weak prior on object size and position. As a consequence, we sometimes propose blocks with unlikely size. **Best viewed in color.**

are keen to see what others can learn from it.

We emphasize that the key contribution is the alternative top-down Bayesian approach, which is likely to prove more powerful as it integrates more information and more sophisticated object models, which can be simultaneously identified while helping explain occluded features of otherwise correct room components. This approach is also able to relatively easily integrate information about color, texture, lighting, and priors about 3D object spatial context. For example, recent work by Lee *et al*. [17] suggests that incorporating detected surface orientations in the reconstruction process would be beneficial. These directions are the topic of ongoing research.

6. Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 0747511. We also acknowledge the valuable contributions of Emily Hartley and Andrew Emmott who helped prepare the ground truth data for the UCB dataset.

References

[1] Code to reproduce results in this paper. http://kobus.ca/research/programs/CVPR_2011_room.html.

[2] Ground truth room layouts for UCB dataset. http://kobus.ca/research/data/CVPR_2011_room.html.

[3] C. Andrieu, N. d. Freitas, A. Doucet, and M. I. Jordan. An introduction to mcmc for machine learning. *Machine Learning*, 50(1):5–43, 2003.

[4] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[5] J. M. Coughlan and A. L. Yuille. Manhattan world: Compass direction from a single image by bayesian inference. In *International Conference on Computer Vision*, pages 941–947, 1999.

[6] 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.

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

[8] P. Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82:711–732, 1995.

[9] P. J. Green. Trans-dimensional markov chain monte carlo. In *Highly Structured Stochastic Systems*. 2003.

[10] F. Han and S.-C. Zhu. Bottom-up/top-down image parsing by attribute graph grammar. In *International Conference on Computer Vision*, volume 2, pages 1778–1785, 2005.

[11] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

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

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

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

[15] D. Hoiem, A. A. Efros, and M. Hebert. Putting objects in perspective. In *CVPR*, 2006.

[16] D. Huttenlocher and S. Ullman. Recognizing solid objects by alignment with an image. *IJCV*, 5(2):195–212, 1990.

[17] D. Lee, A. Gupta, M. Hebert, and T. Kanade. Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1288–1296, 2010.

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

[19] J. S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer-Verlag, 2001.

[20] D. G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):441–450, 1991.

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

[22] B. Russel and A. Torralba. Building a database of 3d scenes from user annotations. In *IEEE International Conference on Computer Vision*, 2009.

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

[24] F. Shi, X. Zhang, and Y. Liu. A new method of camera pose estimation using 2d-3d corner correspondence. *Pattern Recognition Letters*, 25(10):1155 – 1163, 2004.

[25] Z. Tu, X. Chen, A. L. Yuille, and S.-C. Zhu. Image parsing: Unifying segmentation, detection and recognition. *International Journal of Computer Vision*, 63(2):113–140, 2005.

[26] 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.

[27] H. Wang, S. Gould, and D. Koller. Discriminative learning with latent variables for cluttered indoor scene understanding. 2010. Proc. European Conference on Computer Vision (ECCV).

[28] S. Yu, H. Zhang, and J. Malik. Inferring spatial layout from a single image via depth-ordered grouping. pages 1–7, 2008.

[29] S. X. Yu, H. Zhang, and J. Malik. Inferring spatial layout from a single image via depth-ordered grouping. In *Workshop on Perceptual Organization in Computer Vision*, 2008.

[30] L. Zhu, Y. Chen, and A. Yuille. Unsupervised learning of a probabilistic grammar for object detection and parsing. In *NIPS*, 2006.

[31] S. Zhu and D. Mumford. A stochastic grammar of images. *Foundations and Trends in Computer Graphics and Vision*, 2(4):259–362, 2006.

[32] S.-C. Zhu, R. Zhang, and Z. Tu. Integrating topdown/bottom-up for object recognition by data driven markov chain monte carlo. In *IEEE Computer Vision and Pattern Recognition*, 2000.