A dynamic Bayesian network model for autonomous 3d reconstruction from a single indoor image

Erick Delage Honglak Lee Andrew Y. Ng
Computer Science Department
Stanford University
Stanford, CA 94305
{edelage,hllee,ang}@cs.stanford.edu

Abstract

When we look at a picture, our prior knowledge about the world allows us to resolve some of the ambiguities that are inherent to monocular vision, and thereby infer 3d information about the scene. We also recognize different objects, decide on their orientations, and identify how they are connected to their environment. Focusing on the problem of autonomous 3d reconstruction of indoor scenes, in this paper we present a dynamic Bayesian network model capable of resolving some of these ambiguities and recovering 3d information for many images. Our model assumes a "floor-wall" geometry on the scene and is trained to recognize the floor-wall boundary in each column of the image. When the image is produced under perspective geometry, we show that this model can be used for 3d reconstruction from a single image. To our knowledge, this was the first monocular approach to automatically recover 3d reconstructions from single indoor images.

1. Introduction

When we look at the image in Figure 1,[1] our prior knowledge about the world allows us to resolve approximate depth in the image. Given only a single image, depth estimation cannot be done by geometry-only approaches such as a straightforward implementation of stereopsis. In this paper, we focus exclusively on 3d reconstruction from a *single* indoor image. Our motivation for studying this problem is two-fold. First, we anticipate that monocular vision cues could later be applied in conjunction with binocular ones; however, restricting our attention to monocular 3d reconstruction allows us to more clearly elucidate what sorts of monocular vision cues are useful for depth estimation. Our second motivation is that we consider monocular vision to be interesting and important in its own right. Specifically, monocular cameras are cheaper, and their installation is less complex than, stereo cameras. More importantly, the accuracy of stereo vision is fundamentally limited by the baseline distance between the two cameras, and performs poorly when used to estimate depths at ranges that are very large relative to the baseline distance. Straightforward implementations of stereo vision also tend to fail in scenes that contain little texture, such as many indoor scenes (that contain featureless walls/floors). In these settings, monocular vision may be used to complement, or perhaps even replace standard stereopsis.

Figure 1. Single camera image of a corridor

A number of researchers have attempted to recover 3d information from a single camera. "Shape from shading" [21] is one well-known approach, but is not applicable to richly structured/textured images, such as the image in Figure 1. There is also a number of algorithms that use multiple images, such as "structure from motion" [19] and "shape from defocus" [6]. These methods suffer from similar problems to stereopsis when estimating depths at large ranges. In a manner similar to our earlier discussion, we also view

[1] The figures in this paper are best viewed in color.

our work as potentially complementing some of these approaches. For indoor images such as in Figure 1, methods based on "3d metrology" hold some promise. Given sufficient human labeling/human-specified constraints, efficient techniques can indeed be applied to generate a 3d reconstruction of these scenes. [4, 5, 18] The drawback of these methods is that they require a significant amount of human input (for example, specifying the correspondences between lines in the image and the edges of a reference model).

Recent work strongly suggests that 3d information can be efficiently recovered using Bayesian methods, in which visual cues are combined with some prior knowledge on the geometry of a scene. For example, Kosaka and Kak [11] presented a navigation algorithm that allows a monocular robot to track its position in a building by associating visual cues, such as lines and corners, with the configuration of hallways on a plan. However, this approach would fail in a new environment where the plan of the room is not available beforehand. To succeed more generally, one needs to rely on a more flexible geometric model. With a Manhattan world assumption on a given scene (*i.e.* one that contains many orthogonal shapes, like in many urban environments), Coughlan and Yuille [3], and Schindler and Dellaert [16] have developed efficient techniques to recover autonomously both extrinsic and intrinsic camera parameters from a single image. Another successful attempt in the field of monocular 3d reconstruction was developed by Han and Zhu [7, 8], which used models both of man-made "block-shaped objects" and of some natural objects, such as trees and grass. Unfortunately, this approach has so far been applied only to fairly simple images, and seems unlikely to scale in its present form to complex, textured images as shown in Figure 1.

Figure 2. 3d reconstruction of a corridor from single image presented in Figure 1 using our autonomous algorithm.

Hoiem *et al.* [9] also developed independently an algorithm that focuses on generating aesthetically pleasing "pop-up book" versions of outdoor pictures. Although their algorithm is related in spirit, it is different from ours in detail. We will describe a comparison of our method with theirs in Section 4.2. Using supervised learning, [15] give an approach for estimating a depth map from a monocular image, that applies to outdoor/unstructured scenes. (See also [13].)

Our approach uses a dynamic Bayesian network (DBN) to approximate a distribution over the possible structures of a scene. Assuming a "floor-wall" geometry in the scene, the model uses a range of visual cues to find the most likely floor-wall boundary in each column of the image. When the image is produced under perspective geometry and contains only a floor and vertical walls, we show that this can be used to obtain a 3d reconstruction. As an example, Figure 2 shows the 3d reconstruction generated by our algorithm using the image in Figure 1. In Section 2, we define the "floor-wall" geometry and outline our method for recovering 3d information. Section 3 develops the DBN, its training process, and the methods for inferring the most likely floor-wall boundary in an image. Finally, in Section 4 we present a quantitative analysis of the accuracy of reconstruction on test images, and demonstrate the robustness of the algorithm by applying it to a diverse set of indoor images.

2. Background Material

In this paper, we focus on 3d reconstruction from indoor scenes of the sort that are typically seen by an indoor mobile robot. We make the following assumptions about the camera:

1. The image is obtained by perspective projection, using a calibrated camera[2] with a calibration matrix K. Thus, as presented in Figure 3, a point $\mathbf{Q}$ in the 3d world is projected to pixel coordinate $\mathbf{q}$ (represented in homogeneous coordinates) in the image if and only if:[3]
$$\mathbf{Q} \propto K^{-1}\mathbf{q}. \quad (1)$$
2. The image contains a set of N vanishing points corresponding to N directions, with one of them normal to the floor plane. (For example, in a Manhattan world in which all surfaces are orthogonal, $N = 3$.)[4]

[2] A calibrated camera means that the orientation of each pixel relative to the optical axis is known.

[3] Here, K, $\mathbf{q}$ and $\mathbf{Q}$ are as follows:
$$K = \begin{bmatrix} f & 0 & \Delta_u \\ 0 & f & \Delta_v \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{q} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}.$$
Thus, $\mathbf{Q}$ is projected onto a point $\mathbf{q}$ in the image plane if and only if there is some constant α so that $\mathbf{Q} = \alpha K^{-1}\mathbf{q}$.

[4] Vanishing points in the image plane are the points where all lines that are parallel in 3d space meet in the image. This is a consequence of using perspective geometry. Because of the frequency of parallel lines in artificial scenes, they form important cues for depth reconstruction. In a scene that has mainly orthogonal planes—such as in many indoor scenes—the

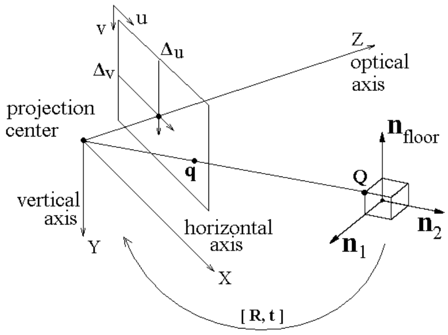


Figure 3. Coordinate systems involved in 3d reconstruction.

3. The scene consists only of a flat floor and straight vertical walls (the "floor-wall" model).

4. The camera's vertical axis is orthogonal to the floor plane, and the floor is on the lower part of the image.

5. The camera center (origin) is at a known height above the ground.[5]

For many indoor scenes, one can make the "Manhattan world" assumption (i.e., that the environment contains only orthogonal planes), and estimate the camera calibration parameters K from the position of three vanishing points. [1] Also, an accurate estimate of the floor plane's normal vector in 3d can be obtained in a similar manner, and thus any small misalignment of the camera's vertical axis can easily be compensated for. Therefore, in Manhattan worlds, it it still possible to generate good reconstructions even if assumptions 1 and 4 are violated. Assumption 2 is often at least approximately satisfied when the scene contains many man-made objects. Finally, the work of [20, 9] suggests that assumption 3 would often be reasonable for indoor and even outdoor scene reconstruction.

In an indoor image, the assumptions above are sufficient to ensure that the full 3d geometry of a scene is exactly specified, given the location of the floor boundary in every column of the image[6]. This result is a direct consequence of perspective geometry. We now describe how to unambiguously reconstruct the 3d location of every pixel in the image.

edges (including the floor-wall boundary) will lie in one of N possible directions, and thus there will be N vanishing points in the image.

[5] If the height of the camera is unknown, then it is still possible to recover the 3d scene up to an unknown scaling factor.

[6] In a column of the image in which the floor is not present, we take this to be the bottom pixel.

First, by perspective projection, the 3d location $\mathbf{Q}_k$ of a pixel at position $\mathbf{q}_k$ in the image plane must satisfy:

$$\mathbf{Q}_k = \alpha_k K^{-1}\mathbf{q}_k, \tag{2}$$

for some α_k. Thus, $\mathbf{Q}_k$ is restricted to a specific line that passes through the origin of the camera. Further, if this point lies on the floor plane with normal vector $\mathbf{n}_{\text{floor}}$, then we have

$$d_{\text{floor}} = -\mathbf{n}_{\text{floor}} \cdot \mathbf{Q}_k = -\alpha_k\, \mathbf{n}_{\text{floor}} \cdot (K^{-1}\mathbf{q}_k)\,, \tag{3}$$

where d_{floor} is the known distance of the camera from the floor. Thus, the 3d positions of the floor pixels (points in the image located below the floor boundary) can be exactly determined.

As for a point $\mathbf{q}_k$ in the wall portion of column j of the image, its 3d location can easily be determined using the knowledge that it is restricted to a vertical segment starting from the known 3d position $\mathbf{Q}_{\mathrm{b}(j)}$ of the floor boundary point in column j.[7] This reduces to solving the following set of linear equations:

$$\mathbf{Q}_{\mathrm{b}(j)} + \lambda_k \mathbf{n}_{\text{floor}} = \mathbf{Q}_k = \alpha_k K^{-1}\mathbf{q}_k, \tag{4}$$

where λ_k and α_k are variables that we need to solve for.[8] In this manner we can reconstruct the 3d position of all the remaining points in the image.

The process described above required knowledge of the floor boundary in each column of the image. In the next section, we will present a data-driven approach to recognizing the floor boundary in an image.

3. Floor boundary estimation using a dynamic Bayesian network

In our experiments, we discovered (perhaps somewhat surprisingly) that simple heuristics for detecting the floor/wall boundary work well only on a very limited set of scenes. Specifically, they often fail when we train and test on images from different buildings, so that the test set's floor color, wall color, floor texture, etc. are not known in advance. (See the discussion in Section 4.2.) We therefore developed a more complex dynamic Bayesian network model to solve this task.

In our model, we let C be a random variable indicating the floor chroma.[9] For each column j in the image,

[7] $\mathbf{Q}_{\mathrm{b}(j)}$ is determined from $\mathbf{q}_{\mathrm{b}(j)}$ and Equation 3, where $\mathbf{q}_{\mathrm{b}(j)}$ is an intersection of the floor boundary and a line which passes through $\mathbf{q}_k$ and the vertical vanishing point. In the simplest case where the optical axis is parallel to the floor, then $\mathbf{q}_{\mathrm{b}(j)}$ is the floor boundary pixel in column j.

[8] In the case that, due to noise in the measurements, this set of equations has no solution, one could simply use the point that minimizes the distance between the two 3d lines:
$(\hat\alpha_k, \hat\lambda_k) = \arg\min_{\alpha_k,\lambda_k} \|\mathbf{Q}_{\mathrm{b}(j)} + \lambda_k \mathbf{n}_{\text{floor}} - \alpha_k K^{-1}\mathbf{q}_k\|_2$.

[9] In this work, we used the CIE-Lab color space for our measurements.

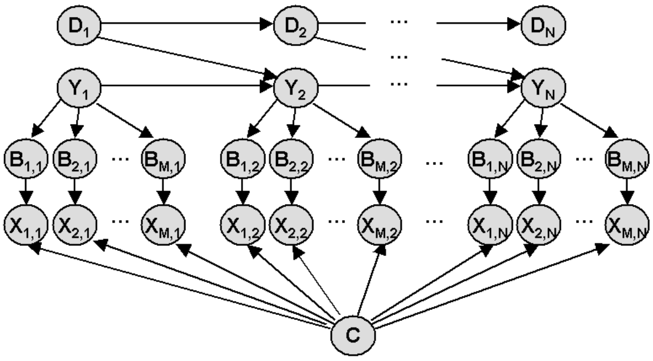


Figure 4. Dynamic Bayesian network model.

let Y_j denote the position of the floor boundary in that column, and let D_j be a latent random variable that indicates the orientation (in the image) of the floor boundary, taking on values corresponding to the vanishing points in the image.[10] Finally, for a point (i, j) in the image, $B_{(i,j)}$ indicates whether there is a floor boundary at that point, and we let $X_{(i,j)}$ denote all local image measurements made at that point. We use the Bayesian network shown in Figure 4 to model the joint distribution of these variables $P(D_{1:N}, Y_{1:N}, B_{(1:M,1:N)}, X_{(1:M,1:N)}, C)$, where M and N are the number of rows and columns of the image respectively. The model assumes (naively) that the local image measurements are conditionally independent of each other given the information about the floor chroma C and the floor boundary $Y_{1:N}$. We have:

$$P(D, Y, B, X, C) = P(D_1)P(Y_1)P(C) \cdot \prod_{j=2}^{N} P(D_j|D_{j-1})P(Y_j|Y_{j-1}, D_{j-1}) \cdot \prod_{i=1}^{M} P(B_{(i,j)}|Y_j)P(X_{(i,j)}|B_{(i,j)}, C) \qquad (5)$$

We now define the probability distributions in the model. To avoid dealing with a continuous random variable, we first run K-means clustering to identify 4 dominant chroma groups present in the bottom part of the image (in practice, 4 groups seemed to be enough to find good chroma candidates). Then, the chroma random variable is treated, for this image, as a discrete random variable that can uniformly take any of the corresponding 4 mean values of these groups. The initial boundary direction variable D_1, and the initial position Y_1 are both modeled using a uniform distribution over their domains. A simple multinomial model is used for the transitions $P(D_j|D_{j-1})$.[11] We model Y_j as $Y_j = f(j, Y_{j-1}, D_{j-1}) + N_j$, where N_j is a noise variable and $f(j, y, d)$ is a function that returns where the line, that passes through $(j-1, y)$ and vanishing point d, intersects column j. From our experiments, we observed that N_j was best modeled by a heavy-tailed distribution. The conditional probability $P(Y_j|Y_{j-1}, D_{j-1})$ was therefore modeled by a mixture of two Gaussians (with variances σ_1^2 and σ_2^2), both of which were centered at the predicted boundary position in column j given Y_{j-1} and D_{j-1}. Finally, $P(B_{(i,j)}|Y_j)$ is deterministic (*i.e.* $B_{(i,j)} = 1\{i - 0.5 \leq Y_j < i + 0.5\}$).

Since X is a high-dimensional continuous-valued random variable, to succinctly represent the conditional probability $P(X_{(i,j)}|B_{(i,j)}, C)$, one natural option would be to use a generative model and assume that it has a multivariate Gaussian distribution for every possible instance of $B_{(i,j)}$. However, in our experiments this learning procedure, which is often referred to as Gaussian Discriminant Analysis (GDA), resulted in poor floor detection performance. This appeared to correspond to the the well known fact that discriminative learning algorithms, which directly estimate the conditional probability distributions used at testing time, can significantly outperform generative learning algorithms. (See [14].) Since at testing time the $X_{(i,j)}$ are observed, the task of inferring the floor boundary is better done by modeling the conditionals $P(B_{(i,j)}|X_{(i,j)}, C)$. In order to keep our suggested Bayesian network structure and at the same time learn a more meaningful conditional distribution from data, we apply Bayes rule and express $P(X_{(i,j)}|B_{(i,j)}, C)$ in a *discriminative* form:

$$P(X_{(i,j)}|B_{(i,j)}, C) = \frac{P(B_{(i,j)}|X_{(i,j)}, C)P(X_{(i,j)}|C)}{P(B_{(i,j)}|C)} \qquad (6)$$

We assumed $P(B_{(i,j)}|C)$ to be uniform, and modeled $P(X_{(i,j)}|C)$ as a Gaussian distribution over the difference between local chroma and the floor chroma C. Finally, we used the following logistic regression model for the conditional distribution $P(B_{(i,j)}|X_{(i,j)}, C)$:

$$P(B_{(i,j)}|X_{(i,j)}, C) = \frac{1}{1 + e^{-\theta \cdot \phi(X_{(i,j)}, C)}}. \qquad (7)$$

Here, ϕ denotes features computed from $X_{(i,j)}$ and C; and θ are the parameters of the model. This logistic regression model is similar to that of [12], which demonstrated good

[10] The vanishing points of an image can be found using completely standard algorithms. [16]

[11] To make this model invariant to vanishing point labeling, we used a distribution with 3 parameters which only assumes that "V_1" is the label given to the vertical vanishing point:

$$\begin{aligned} \theta_1 &= P(D_j = D_{j-1}|D_{j-1} \neq V_1), \\ \theta_2 &= P(D_j = V_1|D_{j-1} \neq V_1), \\ \theta_3 &= P(D_j = D_{j-1}|D_{j-1} = V_1). \end{aligned}$$

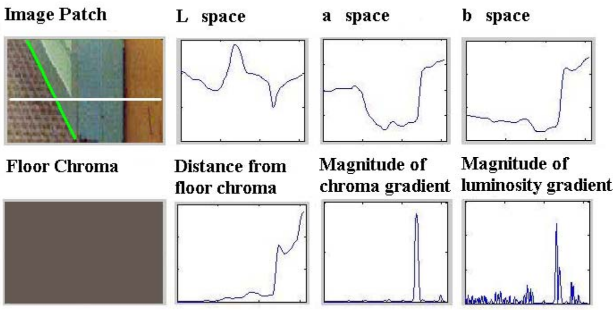


Figure 5. Local image measurements used for floor boundary detection on an example of an image patch, with a specified floor chroma. The measurements were extracted along the white line, while the green line indicates the true floor boundary.

performance in a similar context. (See Section 4.1 for a comparison between this discriminative form of the algorithm and the generative one.)

The logistic regression model uses three different types of local image features ϕ. Our set of features includes standard multi-scale intensity gradients in both the horizontal and vertical directions, as well as their absolute values and their squares. Also for each pixel (i, j), a set of five samples of color are taken at coordinates $(i - 10, j - 10)$, $(i+10, j-10)$, (i, j), $(i-10, j+10)$, and $(i+10, j+10)$ and are used as features. Finally, a measure of similarity to the floor chroma[12] is extracted at (i, j), $(i - 20, j)$, $(i + 20, j)$, $(i, j - 20)$, $(i, j + 20)$. Overall, about 50 features are used to predict if a given pixel is on the floor boundary (see an example of extracted features in Figure 5).

To train and evaluate our algorithm, we acquired a set of 48 pictures taken inside eight different buildings. Each pair of buildings had visibly different interior designs or were patterned on fairly dissimilar themes (for example, different carpeting, different styles of doors, etc.). The actual floor boundary in each image was then hand-labeled. We used standard maximum likelihood estimate to train the parameters for $P(X_{(i,j)}|B_{(i,j)}, C)$ and $P(X_{(i,j)}|C)$. The parameters for $P(Y_j|Y_{j-1}, D_{j-1})$ and $P(D_j|D_{j-1})$ were estimated using the EM algorithm since our training set did not have explicit labels for the floor boundary directions.

Given our learned model of the joint distribution $P(D, Y, B, X, C)$, we then apply it to floor boundary detection in novel images by finding the MAP estimate of the most likely sequence for (D, Y, B, C) given the image. We use:

$$(D, Y, B, C) = \arg \max_{D,Y,B,C} P(D, Y, B, X, C) \quad (8)$$

In order to make inference tractable, we first add to the above optimization problem the constraint that Y_j take only discrete values ($Y_j \in \{1, ..., M\}$). We then formulated the junction tree shown in Figure 6, which is a chain that represents the same distribution as our DBN. Since X is always observed and all other variables are now discrete, we can therefore apply standard forward-backward belief propagation [10] to compute approximately the max-product of Equation 8. Having extracted the floor boundary $Y_{1:N}$, following the discussion of Section 2, a 3d reconstruction of the scene can be generated.

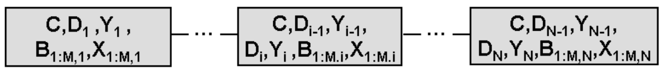


Figure 6. Equivalent junction-tree.

4. Experimental Results

4.1. Accuracy of Algorithm

All 48 images used in this section (including Figure 1) were taken with a calibrated digital camera in 8 buildings of Stanford university's campus and had size 960*1280. The 3d reconstructions were obtained using a form of leave-one-out cross validation in which we repeatedly train our model on images from 7 out of the 8 buildings, and test on the held-out building. Since no two buildings in the training set had similar interior designs nor were otherwise patterned on a common theme, we believe that these results are therefore indicative of the algorithm's ability to generalize to novel buildings and scenes.

Figure 8 shows the 3d reconstructions resulting from the detected floor boundaries.[13] Even in fairly complex environments, the algorithm is able to detect robustly the floor boundary and generate accurate 3d reconstructions.

To evaluate our algorithm more formally, we test it according to two different criteria. First, we measure the RMS error (in pixel space) between the detected floor boundary and the actual floor boundary in the image, and plot the result as a function of the position of the true floor boundary in the image (see Figure 7(a)). Then, in Figure 7(b), the error of the distances recovered (for each column of pixels)

[12] Similarity was measured using Euclidean distance in the CIE-Lab color space.

[13] We have also placed VRML files of the 3d reconstructions online at: http://www.stanford.edu/~edelage/indoor3drecon/.

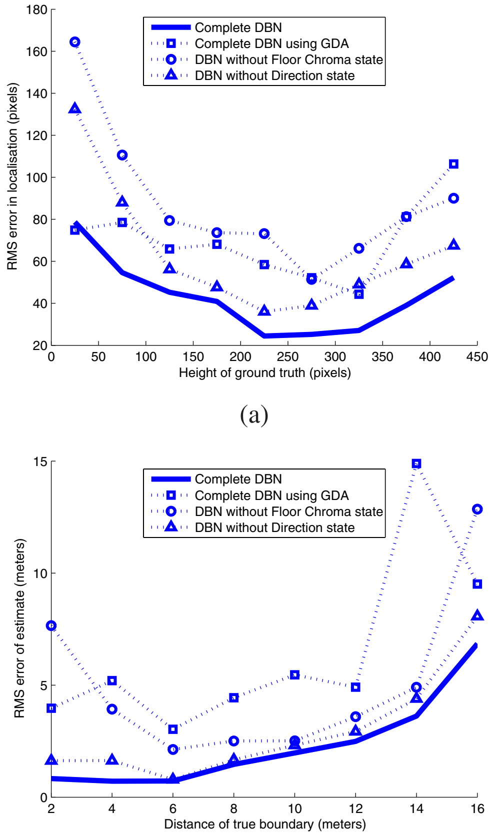
is measured, and plotted as a function of the distance of the nearest wall (to the camera) within that column of pixels. The true distance of the wall is found using the hand-labeled boundary and the camera's calibration parameters. This is a more direct measurement of the algorithm's performance on the depth reconstruction problem.

These figures also show the result of an ablative analysis that we performed to evaluate a number of design choices made in the algorithm, namely the decision to replace the generative form (GDA) for $P(X_{(i,j)}|B_{(i,j)}, C)$ with the discriminative form (logistic regression); the decision to incorporate the hidden floor chroma state C; and the decision to incorporate the hidden direction state D_j. We see from the figure that the full model performs significantly better than one that omits any one of the components mentioned above. This indicates that the directional variables D_j, chroma C, and the discriminative logistic regression all play important roles in reducing error, and that simpler versions of our model do not perform as well. Overall, this validation method estimated our approach to recover depth with an RMS error of less than 0.8 meters for walls in the 3 to 8 meter range (the floor between 0 and 3 meters being out of the camera's view).

4.2. Robustness of Algorithm

In this section, we report results obtained using a database of 44 images, with similar resolution, that were obtained by doing searches on http://images.google.com. Our goal was to verify that our algorithm can accurately obtain 3d reconstructions even from images found on the internet, where we have no guarantees on the orthogonality in the scene or on the alignment of the camera's vertical axis with the floor. Examples of such reconstructions are shown in Figure 9 and on a web-site (refer to footnote 13). Overall, the algorithm obtains a good estimate of the floor boundary on 80% of the images, and generates accurate 3d reconstruction on 66% of them. The main source of errors for reconstruction was the mislabeling of vanishing points when the scene contained many irregular objects. We believe that these errors can thereby be corrected by using more robust algorithms for vanishing point detection. Errors also occurred more often for points on the far left side or far right side of the image, where the algorithm had less information to infer the floor boundary. As expected, the algorithm generated slightly more errors in scenes that were composed of curved walls, or scenes in which the floor had more than one colored region (see Figure 9(d)).

We also used our test set images to compare our algorithm to that of Hoiem *et al.* [9]. Qualitatively, their algorithm showed significantly less accurate 3d reconstructions than ours. More specifically, we used the 44 images as test set. Their algorithm failed more often to reconstruct



(a)

(b)

Figure 7. Comparison of performance of our graphical model to three of its simplified forms. (a) Analysis of floor boundary localization error in the image. (b) Analysis of floor boundary depth estimation error.

the scenes accurately (20 valid reconstructions for theirs, against 29 for ours). We believe that this difference is mainly due to larger errors in the floor segmentation process, and to the heuristic method they use to organize the walls around the floor region. In fact, Hoiem *et al.*'s algorithm does not explicitly use the geometric information about indoor images that our algorithm uses, such as that most walls lie in a small number of directions, and that they connect to each other only in certain ways. This lack of prior knowledge/constraints about indoor environments explains the superior performance of our algorithm on test images.

From our experiments, we also learned that the problem of finding the floor boundary is not a trivial problem, and that specifically it cannot be solved using simple methods. For example, our extensive experiments (not reported here) seemed to indicate that a naive application of the Hough transforms does not work because it is difficult to choose only the relevant edges (lines) for detecting floor boundaries; moreover, there may be errors in edge detection when the image is highly textured. We also made numerous attempts to perform this task using color segmentation [2] and normalized cut [17]. But all of these approaches failed to robustly segment the floor boundary on a majority of the test images, because many of them, like Figure 9(c), contain floor parts which have non-uniform brightness or chroma, and rich textures. We found no other simple heuristic whose performance even approached that of these algorithms. Thus, to successfully detect the floor boundary, we believe that one has to use a model that combines and reasons about many different image cues, such as ones that indicate the color, brightness, and geometry of the scene.

5. Conclusion

Monocular 3d reconstruction is inherently an ambiguous problem, but by using prior knowledge about a domain, it is often possible to recover distances from a single image. In this paper, we showed how prior knowledge about indoor scenes can be learned using a dynamic Bayesian network, and demonstrated a successful application of this model to monocular 3d reconstruction. The problem of learning to exploit other monocular cues for indoor and outdoor environments, and building accurate 3d reconstructions of scenes with a more diverse geometry, remains an important research problem. We believe that our approach holds promise for building better systems that can sense in, understand, and navigate rich 3d environments.

Acknowledgments

We give warm thanks to Sebastian Thrun, Gary Bradski, Larry Jackel, Pieter Abbeel, Ashutosh Saxena, and Chuong Do for helpful discussions. This work was supported by the DARPA LAGR program under contract number FA8650-04-C-7134.

References

[1] R. Cipolla, T. Drummond, and D. Robertson. Camera calibration from vanishing points in images of architectural scenes. In *Proc. Tenth British Machine Vision Conference*, pages 382–391, 1999. 3

[2] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24:603–619, 2002. 7

[3] J. M. Coughlan and A. L. Yuille. Manhattan world: Orientation and outlier detection by bayesian inference. *Neural Computation*, 15(5):1063–1088, 2003. 2

[4] A. Criminisi, I. D. Reid, and A. Zisserman. Single view metrology. *Int'l J. Computer Vision*, 40(2):123–148, 2000. 2

[5] P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *Proc. SIGGRAPH*, 1996. 2

[6] P. Favaro and S. Soatto. Shape and radiance estimation from the information-divergence of blurred images. In *Proc. European Conf. Computer Vision*, 2000. 1

[7] F. Han and S.-C. Zhu. Bayesian reconstruction of 3d shapes and scenes from a single image. In *Proc. Int'l Workshop on High Level Knowledge in 3D Modeling and Motion*, 2003. 2

[8] F. Han and S. C. Zhu. Bottom-up/top-down image parsing by attribute graph grammar. In *Int'l Conf. Computer Vision*, 2005. 2

[9] D. Hoiem, A. A. Efros, and M. Hebert. Geometric context from a single image. In *Int'l Conf. Computer Vision*, 2005. 2, 3, 6

[10] F. V. Jensen. *Bayesian Networks and Decision Graphs*. Springer-Verlag, New York, USA, 2001. 5

[11] A. Kosaka and A. C. Kak. Fast vision-guided mobile robot navigation using model-based reasoning and prediction of uncertainties. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 56(3):271–329, 1992. 2

[12] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using brightness and texture. In *Neural Information Processing Systems*, 2002. 4

[13] J. Michels, A. Saxena, and A. Y. Ng. High speed obstacle avoidance using monocular vision and reinforcement learning. In *Proc. Int'l Conf. Machine Learning*, 2005. 2

[14] A. Y. Ng and M. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Neural Information Processing Systems*, 2002. 4

[15] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In *Neural Information Processing Systems*, 2005. 2

[16] G. Schindler and F. Dellaert. Atlanta world: An expectation maximization framework for simultaneous low-level edge grouping and camera calibration in complex man-made environments. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 203–209, 2004. 2, 4

[17] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. 7

[18] H.-Y. Shum, M. Han, and R. Szeliski. Interactive construction of 3d models from panoramic mosaics. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 427–433, 1998. 2

[19] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *Int'l J. Computer Vision*, 9:137–154, 1992. 1

[20] I. Ulrich and I. R. Nourbakhsh. Appearance-based obstacle detection with monocular color vision. In *Proc. 20th Nat'l Conf. Artificial Intelligence (AAAI)*, pages 866–871, 2000. 3

[21] R. Zhang, P.-S. Tsai, J. E. Cryer, and M. Shah. Shape from shading: A survey. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21(8):690–706, 1999. 1

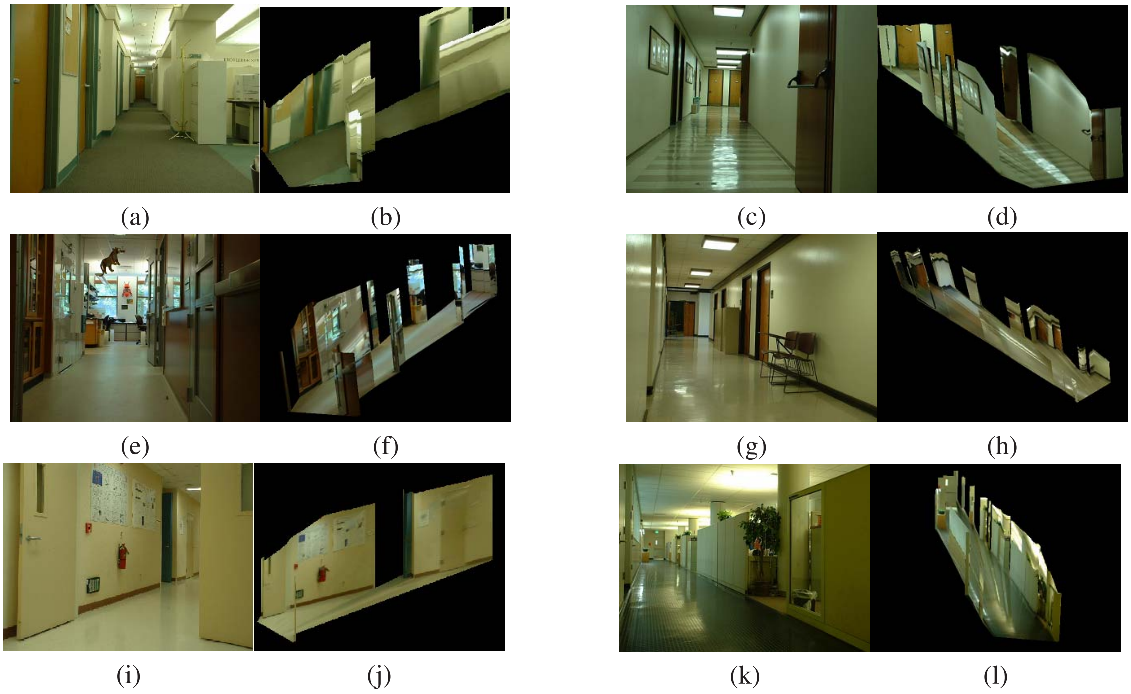


Figure 8. Inferred 3d reconstruction of indoor scenes by our dynamic Bayesian network. (a,c,e,g,i,k) present images obtained with a calibrated camera on Stanford's campus, (b,d,f,h,j,l) present the 3d reconstructions.

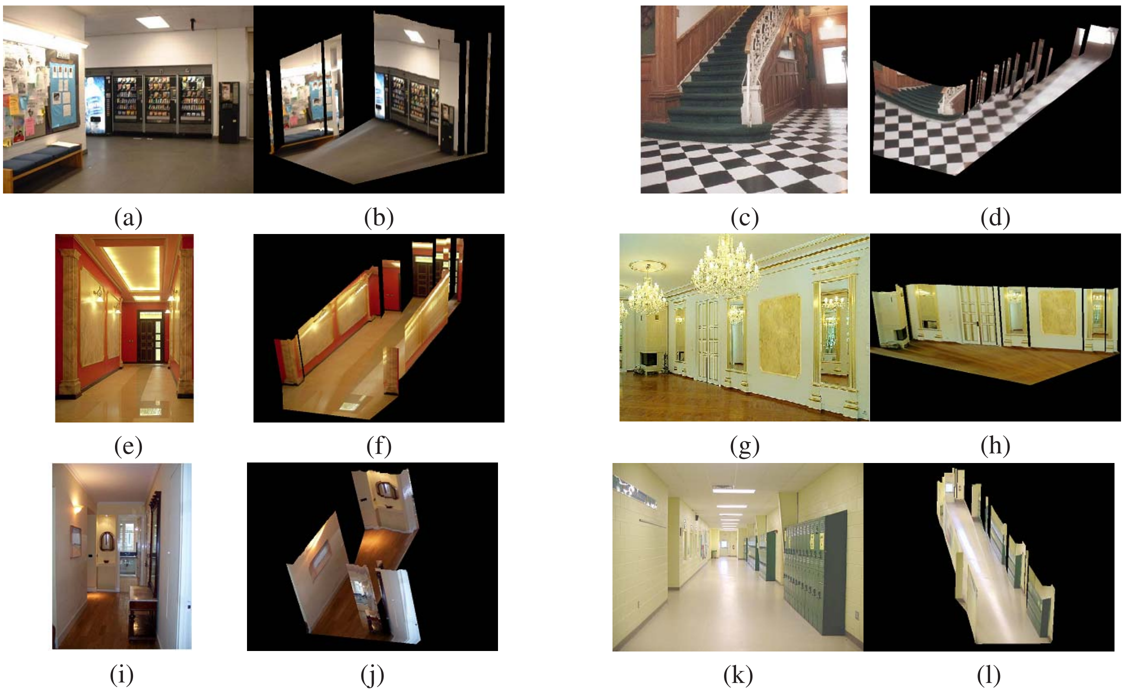


Figure 9. Inferred 3d reconstruction of indoor scenes by our dynamic Bayesian network. (a,c,e,g,i,k) present images obtained from the internet, (b,d,f,h,j,l) present the 3d reconstructions.