A Dynamic Bayesian Network Model for Autonomous 3d Reconstruction from a Single Indoor Image

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

Images obtained with a calibrated digital camera on Stanford University campus.

Abstract

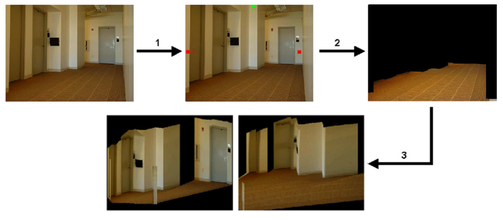
When we look at a picture, our prior knowledge about the world allows us to resolve some of the ambiguities that are inherent to monocular vision, and thereby infer 3d information about the scene. We also recognize different objects, decide on their orientations, and identify how they are connected to 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.

Overview of the Algorithm

Assumptions on the Image

1. The image is obtained by perspective projection, using a calibrated camera with a calibration matrix K.
2. The image contains a set of N vanishing points corresponding to N directions, with one of them normal to the floor plane.
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.

Autonomous Indoor 3d Reconstruction Algorithm



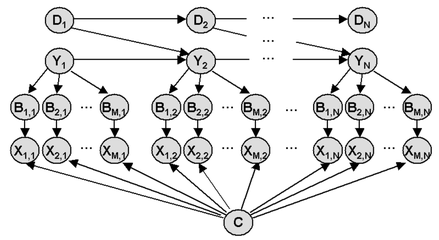
Under the above assumptions, given a single image the algorithm:

1. Extracts the vanishing points using standard techniques (*e.g.* [1, 9]) and identifies the vertical vanishing point.
2. Estimates the location of the floor boundary in every column of the image using a trained Dynamic Bayesian Network and segments the floor pixels in the image.
3. Applies perspective geometry to reconstruct the 3d geometry of the scene.

Floor Boundary Detection

Dynamic Bayesian Network

The Bayesian network models a joint distribution in an MxN image



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

where:

- C is the floor chroma, taking on values corresponding to the means of 4 dominant chroma groups present in the bottom part of the image (identified using K-means clustering).[a]
- Y_j is the position of the floor boundary in column j.
- D_j indicates the orientation (in the image) of the floor boundary, taking on values corresponding to the vanishing points in the image.
- $B_{(i,j)}$ indicates the presence of a floor boundary at (i, j).[b]
- $X_{(i,j)}$ denotes local image measurements made at (i, j). A total of 50 features including standard multi-scale intensity gradients, local color samples, and a similarity measure between local color samples and the floor chroma.[c]
- $P(C)$, $P(D_1)$, $P(Y_1)$ are uniform distribution over their domain.
- $P(D_j|D_{j-1})$ is a multinomial distribution with constrained parameters to ensure invariance to vanishing point labelling.
- $P(Y_j|Y_{j-1}, D_{j-1}) = P(f(j, Y_{j-1}, D_{j-1}) + N_j|Y_{j-1}, D_{j-1})$, where N_j is a noise variable and $f(j, y, d)$ is a function that returns the one step prediction for the boundary given its position Y_{j-1} and direction D_{j-1}. N_j was best modeled by a mixture of two Gaussians (with variances σ_1^2 and σ_2^2).
- $P(B_{(i,j)}|Y_j)$ is deterministic ($B_{(i,j)} = 1\{i - 0.5 \leq Y_j < i + 0.5\}$).
- $P(X_{(i,j)}|B_{(i,j)}, C)$ was described in a *discriminative* form

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

with $P(B_{(i,j)}|C)$ uniform over its domain, $P(X_{(i,j)}|C)$ normally distributed, and $P(B_{(i,j)}|X_{(i,j)}, C) = 1/\left(1 + e^{-\theta \cdot \phi(X_{(i,j)}, C)}\right)$.

[a] In this work, we used the CIE-Lab color space for our measurements.
[b] (i, j) denotes (row, column).
[c] Similarity was measured using Euclidean distance in the CIE-Lab color space.

Training the model's parameters

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.

Detecting the floor boundary

We applied the learned model 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.

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

In order to make inference tractable, we first add the constraint that Y_j take only discrete values ($Y_j \in \{1, ..., M\}$). The floor boundary is found using standard forward-backward belief propagation [6] on a junction tree that represents the same distribution as our DBN.

Indoor Scene Reconstruction

Reconstruction of Floor

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

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

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

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

where d_{floor} is the known distance of the camera from the floor. Thus, the 3d positions of the floor pixels can be exactly determined.

Reconstruction of Walls

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 $\mathbf{Q}_{\text{b}(j)}$ of the known floor boundary point in column j. This reduces to solving the following set of linear equations:

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

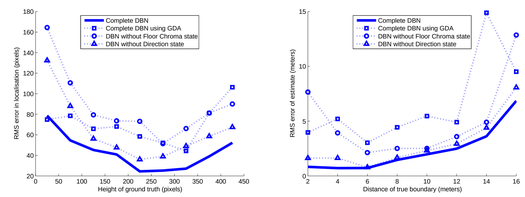
where λ_k and α_k are variables that we need to solve for.[a]

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

Experimental Results

Accuracy of Algorithm

All 48 images used in these tests 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 (train on images from 7 buildings and test on the held-out building).



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

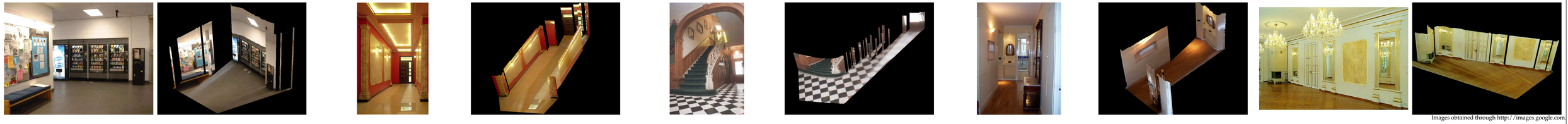
Robustness of Algorithm

On a database of 44 images, with similar resolution, that were obtained by doing searches on http://images.google.com, the algorithm obtains a good estimate of the floor boundary on 35 (80%) of the images, and generates accurate 3d reconstruction on 29 (66%) of them.

Hoiem *et al*. [5] has also developed independently an algorithm to accomplish autonomous reconstruction on outdoor images. However, when applied to indoor images, their algorithm does not explicitly use geometric information 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 these images (*i.e.*, their algorithm generated only 20 (45%) accurate 3d reconstructions).

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.
[2] A. Criminisi, I. D. Reid, and A. Zisserman. Single view metrology. *Int'l J. Computer Vision*, 40(2):123–148, 2000.
[3] 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.
[4] 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.
[5] D. Hoiem, A. A. Efros, and M. Hebert. Geometric context from a single image. In *Int'l Conf. Computer Vision*, 2005.
[6] F. V. Jensen. *Bayesian Networks and Decision Graphs*. Springer-Verlag, New York, USA, 2001.
[7] 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.
[8] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In *Neural Information Processing Systems*, 2005.
[9] 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.
[10] 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.
[11] 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.



Images obtained through http://images.google.com.

Acknowledgements: 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.