Automatic Single-Image 3d Reconstructions of Indoor Manhattan World Scenes

Erick Delage, Honglak Lee, and Andrew Y. Ng

Stanford University, Stanford, CA 94305 {edelage,hllee,ang}@cs.stanford.edu

Summary. 3d reconstruction from a single image is inherently an ambiguous problem. Yet when we look at a picture, we can often infer 3d information about the scene. Humans perform single-image 3d reconstructions by using a variety of single-image depth cues, for example, by recognizing objects and surfaces, and reasoning about how these surfaces are connected to each other. In this paper, we focus on the problem of automatic 3d reconstruction of indoor scenes, specifically ones (sometimes called "Manhattan worlds") that consist mainly of orthogonal planes. We use a Markov random field (MRF) model to identify the different planes and edges in the scene, as well as their orientations. Then, an iterative optimization algorithm is applied to infer the most probable position of all the planes, and thereby obtain a 3d reconstruction. Our approach is fully automatic—given an input image, no human intervention is necessary to obtain an approximate 3d reconstruction.

1 Introduction

When viewing a single image such as that in Figure 1, most humans have little trouble estimating the 3d shape of the scene. Given only a single image, depths are inherently ambiguous, and thus 3d reconstruction cannot be achieved using naive, geometry-only approaches such as a straightforward implementation of stereopsis (binocular vision). In this paper, we consider the task of monocular (single camera) 3d reconstruction, specifically of indoor scenes consisting mainly of orthogonal planes. Our motivation for studying the monocular 3d reconstruction problem is two-fold. First, although one may envision systems that use both monocular and binocular cues, as a scientific endeavor we find it most enlightening to focus exclusively on monocular vision; specifically, this allows us to try to elucidate how monocular cues—which have heretofore been little-exploited in automatic 3d reconstructions—can be used. Second, we consider monocular 3d reconstruction to be interesting and important in its own right. For example, unlike stereo vision, it works well even at large distances (if, say, the images are taken through a zoom lens). In contrast, 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.

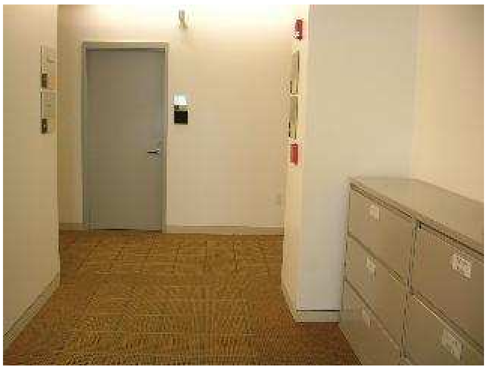
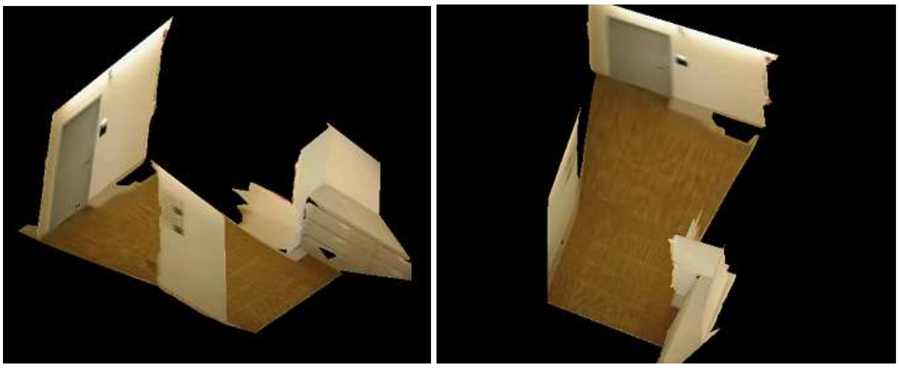


Fig. 1. Single camera image of a corridor.

Apart from stereopsis, there are many other algorithms that use multiple images to estimate depths, such as structure from motion [23] and shape from defocus [8]. These methods suffer from similar problems to stereopsis when estimating depths at large ranges. A number of researchers have attempted to recover 3d information from a single image. Shape from shading [25] is one well-known approach, but is not applicable to richly structured/textured images such as that in Figure 1. For such indoor images, methods based on "3d metrology" hold some promise. Given sufficient human labeling/human-specified constraints, efficient techniques can be used to generate a 3d reconstruction of the scene. [5, 6, 21, 22] However, these methods tend to require a significant amount of human input (for example, specifying the correspondences between lines in the image and the edges of a reference model), and are thus limited in their applicability.

Recent work strongly suggests that 3d information can be efficiently recovered using Bayesian methods that combine visual cues with some prior knowledge about the geometry of a scene. For example, Kosaka and Kak [13] give 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 floor plan. However, this approach would fail in a new environment in which such a floor plan is not available beforehand. A more flexible algorithm, due to Han and Zhu [11], 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. Saxena, Chung and Ng [19] apply an MRF to directly estimating depths from a monocular image, focusing mainly on unstructured (for example, outdoor) scenes. (See also [18].) The “Manhattan world” assumption [3, 4] (i.e., that the environment contains only orthogonal planes, as in many urban environments) has been used to develop algorithms for estimating camera calibration parameters [20] and camera pose [3, 4] from complex images. In this paper, we exploit this same assumption to obtain single-image 3d reconstructions.



(a) (b)

Fig. 2. 3d reconstruction of a corridor from single image presented in figure 1.

Our approach uses a Markov random field (MRF) to estimate whether each point in an image represents a surface or an edge, and also the orientation of the surface or edge. Using this information, we then use an iterative algorithm to try to infer the 3d reconstruction of the scene. Figure 2 shows an example of our algorithm’s output, generated fully automatically from the image in Figure 1. To our knowledge, our work represents the first fully automatic algorithm for 3d reconstruction from single indoor images.

The remainder of this paper is structured as follows. In Section 2, we describe the basic geometric calculations used by our algorithms. Section 3 presents the MRF model; and Section 4 then describes how we compute a 3d reconstruction from the MRF’s output. In Section 5, we present experimental results.

2 Preliminaries

We make the following assumptions:

1. The image is obtained by perspective projection, using a calibrated camera[1] with 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:[2]
$$\mathbf{Q} \propto K^{-1}\mathbf{q}. \tag{1}$$
2. The objects in the image are composed of planes in each of three mutually orthogonal orientations. Thus, the image also contains three vanishing points corresponding to three different directions (one of them orthogonal to the floor plane).[3]
3. The camera's vertical axis is orthogonal to the floor plane, and the floor is in the lower part of the image.(Figure 3)[4]
4. The camera center (origin of the coordinate frame) is at a known height above the ground.[5]

Assumption 2 is often called the Manhattan world assumption [3].

In an image that has no occluding edges, the assumptions above are sufficient to ensure that the full 3d geometry of a scene is exactly specified, given only a segmentation of the scene into surfaces (together with labels indicating the surfaces' orientations). Thus, knowledge of the segmentation and orientations is sufficient to unambiguously reconstruct the 3d location of every pixel in the image. This result is a completely straightforward consequence of perspective geometry. Still assuming the absence of occluding edges, we now describe how this 3d reconstruction can be obtained.

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

$$\mathbf{Q}_i = \lambda_i K^{-1}\mathbf{q}_i \tag{2}$$

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

[2] 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 λ so that $\mathbf{Q} = \lambda K^{-1}\mathbf{q}$.

[3] Vanishing points in the image plane are the points where lines that are parallel in the 3d space meet in the image. In a scene that has mainly orthogonal planes—such as in many indoor scenes—most edges (in the 3d world) will lie in one of three possible directions, and thus there will be three vanishing points in the image.

[4] Small misalignments of the camera's vertical axis can also be easily compensated for (e.g., see [3, 4]).

[5] If the height of the camera is unknown, then the 3d reconstruction will be determined only up to an unknown scaling factor.

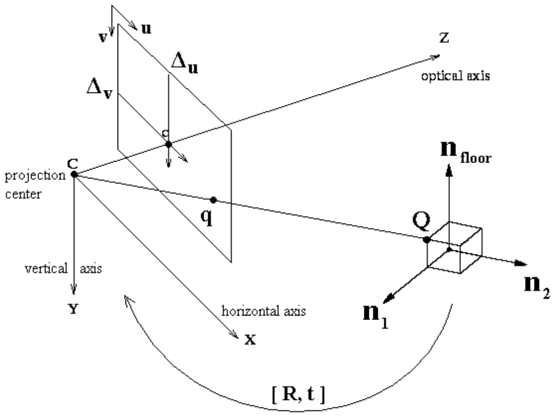


Fig. 3. Coordinate system used by algorithm.

for some λ_i. Thus, $\mathbf{Q}_i$ is restricted to a specific line that passes through the origin of the camera. Further, if this point lies on some plane p that has normal vector $\mathbf{n}_p$, then we have

$$\mathbf{n}_p \cdot \mathbf{Q}_i = \lambda_i \ \mathbf{n}_p \cdot (K^{-1}\mathbf{q}_i) = d_p, \tag{3}$$

where d_p is the distance of the plane from the camera center (the origin of the 3d coordinate frame). Thus, λ_i can be exactly determined given only d_p; and therefore estimating the position of every pixel in the image reduces to the problem of finding d_p for all planes p.

Since we assumed that there are no occluding edges, every two adjacent pixels in the image are also physically adjacent in 3d.[6] Since each point $\mathbf{q}_i$ (with variable λ_i) is part of some plane, each variable λ_i is constrained by at least one equation of the form in Equation (3). Moreover, if there are no occluding edges in the image, then the points lying on the boundary of two adjacent/connected planes participate in two different constraints (one for each of the two neighboring planes). By incorporating assumption 4, we also know the distance d_p from the floor plane to the camera. Except in degenerate cases, this is sufficient to ensure that, treating the λ_i and d_p as variables, the system of equations given in Equation (3) are sufficiently constrained to have a unique solution.

The process described above required knowledge of the segmentation of the scene into planes as well as knowledge of the orientation of the planes. In Section 3, we describe an algorithm for estimating these quantities. Furthermore, the assumption that there are no occluding edges will often fail to hold

[6] Section 4 will address the case of occluding edges.

in indoor scenes; in Section 4, we describe a reconstruction algorithm that applies even in the presence of occluding edges.

3 Markov random field model

Given an image of a scene comprising planes in three mutually orthogonal directions, there are standard algorithms for recovering the three vanishing points in the image. (E.g., [17, 20]) We use [20] to identify these vanishing points; by doing so, we also identify the three possible orientations for the planes $\mathbf{n}_{\text{floor}}$, $\mathbf{n}_1$, and $\mathbf{n}_2$ (one orthogonal to each of the vanishing point directions).

In our Manhattan world, the edges (boundaries) of a plane cannot be oriented in the same direction as its normal. If there is no occlusion, this gives us a constraint on the possible directions for the edges of a surface. (For example, the floor should not be bordered by edges that point upwards in the direction $\mathbf{n}_{\text{floor}}$). Our MRF model will incorporate this constraint.

Our MRF is structured as a 320*240 grid (each node corresponding to a different position in the image). Each node corresponds to a random variable that takes on one of 6 possible values, that indicate whether the node is on a line pointing toward one of the three vanishing points (labels e_1, e_2, e_3), or whether it lies on a plane whose normal is oriented in one of the three orthogonal directions (labels p_1, p_2, p_3). Figure 4 shows the 6 labels. The MRF models the joint probability distribution of this 320*240 grid of label values; and will be used to infer the most likely set of labels given a new image.

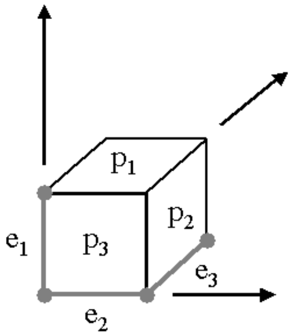


Fig. 4. The 6 possible labels for the MRF nodes (points in the 2d image).

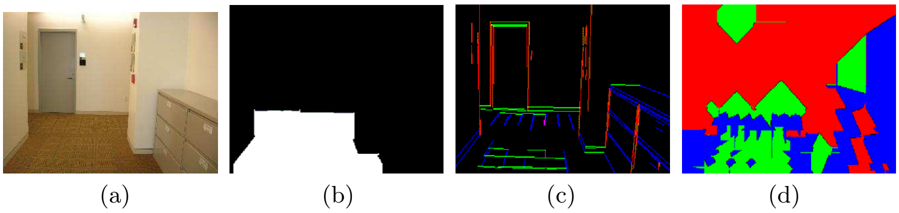


Fig. 5. Example of features extracted from an image. (a) The input image. (b) A mask identifying the floor pixels. (c) Lines extracted using [16] and classified according to their orientations. (d) Labeling of each pixel with the direction of the edge in (c) which is closest in the same row or column.

3.1 MRF features

This section briefly describes the image features used in our MRF model.

Edge statistics features

Statistics about edges were computed using the Canny edge detector [2], the phase congruence [15], and Sobel edge filter [10]. Using the orientation of intensity gradients, we also determined for each location the most likely vanishing point of each edge. Line extraction algorithms from [14] and [16] were used to obtain a list of lines in the image (generating two different sets of features). Each line was also identified according to its vanishing point.[7] We also created additional features based on the nearby edges' orientations.[8]

Segmentation-based features

Surfaces often have fairly uniform appearances in texture and color, and thus image segmentation algorithms provide another set of useful features. Specifically, pixels that are members of the same segmented group should usually be labeled with the same orientation. We used a graph-based segmentation algorithm [9] to generate a partition of the image, and assigned a unique identifier to each partition output by the segmentation algorithm. For each pair of adjacent nodes in the grid, we also generated a pairwise/relational feature in our MRF model indicating whether the nodes were members of the same partition of the image.

[7] Lines which diverged from all three vanishing points were discarded. Some lines whose 3d orientations were ambiguous were assigned to two vanishing points.

[8] At a given position in the image, we add an extra feature corresponding to the orientation of the closest line (measured either in the same row or column) in the image. (See Figure 5d.) We also created additional features corresponding to the second and third closest lines.

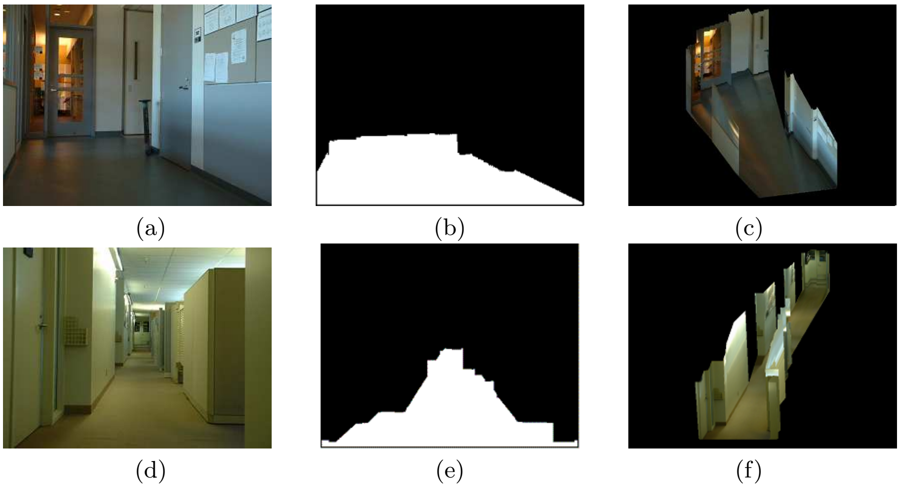


Fig. 6. Results from DBN floor segmentation algorithm of [7]. (a),(d) original image. (b),(e) floor mask. (c),(f) 3d reconstruction (obtained assuming presence only of floor and walls in image).

Floor segmentation features

Since many planes (e.g., most walls) are connected to the floor, correct labeling of the floor plane plays an important role in 3d reconstruction. Building on our earlier work [7], we used a dynamic Bayesian network (DBN) to identify the floor boundary in the image plane. Our DBN is a probabilistic model that incorporates a number of local image features, and tries to reason about the chroma of the floor, the position of the floor boundary in each column of the image, and the local direction of the floor boundary. The DBN output is then used to generate a "floor mask" feature indicating whether each pixel was identified as part of the floor.[9] (See Figure 5b.)

In [7], it was shown that if the image contains only the floor and vertical walls, then (under mild assumptions) knowledge of this floor boundary is sufficient to give a complete 3d reconstruction of the scene. The basic idea is that, given the camera height and orientation, every point in the ground plane can be reconstructed exactly. Then, because the position of each point on the boundary between the floor and each wall is now known (because these points also comprise part of the ground plane), we also now know the 3d position of the lower-edge of each wall. This is sufficient to exactly reconstruct the position of each wall. Figure 6 shows some examples of results obtained using this procedure. We note, however, that this procedure does not apply to scenes

[9] Two additional features were created using the DBN output: one to identify edges of the floor boundary; the other to identify sharp changes in direction of the floor boundary (which are often indicative of a transition between two wall planes).

that have other orthogonal surfaces (e.g., the top surfaces of desks and filing cabinets), such as in the Manhattan worlds considered in the present paper.

3.2 MRF parameterization

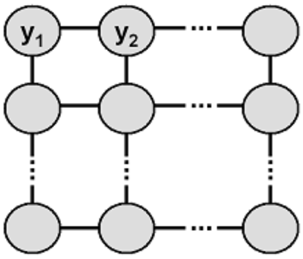


Fig. 7. Markov random field model over the image.

As discussed previously, each node can take on one of 6 possible label values: 3 for plane orientations (labels p_1, p_2, p_3) and 3 for edge orientations (labels e_1, e_2, e_3).[10] We used a grid-structured Markov random field. Figure 7 shows the structure of the MRF. We use V to denote the set of nodes in the model, and E to denote the edges. Let $y_v \in \{p_1, p_2, p_3, e_1, e_2, e_3\}$ denote the value associated with vertex $v \in V$, and let x_v denote the vector of features computed at position v in the image (and similarly $x_{u,v}$ be computed from positions u and v). The MRF defines a joint probability distribution over all label assignments y:

$$P_\theta(y|x) = \frac{1}{Z_\theta(x)} \exp\left(-\sum_{v \in V} \Psi_1(y_v, x_v; \theta_1) - \sum_{(u,v) \in E} \Psi_2(y_u, y_v, x_{u,v}; \theta_2) \right) . \tag{4}$$

Here, Ψ_1 is the potential function for individual nodes, Ψ_2 gives the pairwise potentials in the MRF, $\theta = [\theta_1, \theta_2]$ are the parameters of the model, and $Z_\theta(x)$ is the partition function.

Using the features described in Section 3.1, we chose $\Psi_1(y_v, x_v; \theta_1)$ to be a weighted linear combination of features indicative of the label at a vertex v:[11]

[10] A plane with orientation p_i has a normal in direction e_i. Thus, a plane with orientation p_1 would typically be bordered by edges of type e_2 and e_3

[11] For example, given a specific edge-based feature $C_1(v, x_v)$ from Section 3.1 (one that is indicative of whether an edge at position v heads towards e_1), we create the following MRF features:

$$\Psi_1(y_v, x_v; \theta_1) = \theta_1^{\mathsf{T}} \cdot \Phi(y_v, x_v). \tag{5}$$

Similarly, we used

$$\Psi_2(y_u, y_v, x_{u,v}; \theta_2) = \theta_2^{\mathsf{T}} \cdot \Phi(y_u, y_v, x_{u,v}), \tag{6}$$

where $\Phi(y_u, y_v, x_{u,v})$ were chosen to be features indicative of whether y_u and y_v are likely to be the same label (e.g., the segmentation-based feature of Section 3.1). We also included features in the pairwise potential that measure "consistency" between the plane and the edge orientations.[12] For example, these features can be used to help capture the fact (discussed earlier) that a plane with normal p_i is unlikely to be bordered by edges of orientation e_i.

Putting all the features together, $\Phi(y_v, x_v)$ was a 75 dimension vector, and $\Phi(y_u, y_v, x_{u,v})$ was a 9 dimension vector.

3.3 Training and inference

In order to train the model parameters θ_1 and θ_2, we hand-labeled two images with their ground-truth labels y. This set of two images made up our training set. Unfortunately, maximum likelihood parameter learning is intractable in grid-structured MRF models; thus we learned the parameters using an objective similar to pseudo-likelihood.[13] [1]

$$\begin{aligned}
\Phi_1(y_v, x_v) &= C_1(v, x_v) \times 1\{y_v = e_1\} \\
\Phi_2(y_v, x_v) &= C_1(v, x_v) \times 1\{(y_v = e_2) \ \vee \ (y_v = e_3)\} \\
\Phi_3(y_v, x_v) &= C_1(v, x_v) \times 1\{(y_v \neq e_1) \ \wedge \ (y_v \neq e_2) \ \wedge \ (y_v \neq e_3)\},
\end{aligned}$$

[12] For example:

$$\begin{aligned}
\Phi_1(y_u, y_v, x_{u,v}) &= 1\{y_u = \text{plane} \ \wedge \ y_v = y_u\} \\
\Phi_2(y_u, y_v, x_{u,v}) &= 1\{y_u = \text{plane} \ \wedge \ y_v = \text{plane} \wedge \ y_u \neq y_v\} \\
\Phi_3(y_u, y_v, x_{u,v}) &= 1\{y_u = \text{edge} \ \wedge \ y_v = \text{edge}\} \\
\Phi_4(y_u, y_v, x_{u,v}) &= \sum_{i=1}^{3} 1\{y_u = p_i \ \wedge \ y_u = e_i\} \\
\Phi_5(y_u, y_v, x_{u,v}) &= \sum_{i=1}^{3} 1\{y_u = p_i \ \wedge \ y_v = \text{edge} \wedge \ y_u \neq e_i\}
\end{aligned}$$

[13] In our experiments, straightforward pseudo-likelihood (or generalized pseudo-likelihood [12] using small clusters of nodes) did not work well. Our parameters were actually learned using a product approximation over 3-node networks. More formally, we used:

$$\max_{\theta} \prod_{(u,v,w) \in \mathbf{F}} \hat{P}_\theta(y_u, y_v, y_w | x),$$

Finally, after learning the parameters, the inference task in our Markov random field is to compute the most likely set of labelings, given a feature vector x from a new image:

$$\hat{y} = \arg\max_{y} P_\theta(y|x), \tag{7}$$

Exact inference in a grid-structured MRF is intractable. We approximated this using the algorithm of Wainwright et al. [24].

4 Using the MRF output for 3d reconstruction

We now address the problem of 3d reconstruction given an image in which the planes have been segmented and labeled with their orientations, for example by our MRF. Sturm and Maybank [22] proposed an algorithm for a similar problem, and demonstrated good 3d reconstruction given human-labeled images. However, their algorithm is not directly applicable to an image labeled by our MRF, as it requires that occlusion vs. non-occlusion edges be labeled (i.e., labels indicating whether two adjacent planes in the image are physically connected in 3d). This is difficult to infer from local image features, and is not part of the information output by our MRF. Their algorithm has also been tested only on instances with perfectly correct human-generated labels. We now present an algorithm, a modification and generalization of Sturm and Maybank's algorithm, for 3d reconstruction from an image given possibly noisy labels of the planes and edges.

If we examine an individual "edge" point $\mathbf{q}_i$ that is on the boundary between two planes p and p', this point can either be part of an occluding edge between the two planes or part of an edge that physically connects the two planes p and p'. Therefore, in the latter case we would want to find a 3d reconstruction where the following distance is small:

$$\Delta_{i,p,p'} = \|\mathbf{Q}_{i,p} - \mathbf{Q}_{i,p'}\|_2.$$

Here, $\mathbf{Q}_{i,p}$ (respectively $\mathbf{Q}_{i,p'}$) is the 3d position in the plane of p (respectively p') that would appear at position $\mathbf{q}_i$ in the image plane. Thus, we can informally think of $\Delta_{i,p,p'}$ as the distance between (two specific points on) the planes p and p'.

Thus argument above applies if an edge is known to be non-occluding. However, it is usually not obvious if an edge is indeed occluding, and thus

where

$$\hat{P}_\theta(y_u, y_v, y_w|x) = \frac{1}{\hat{Z}_\theta(x)} \exp\left(- \sum_{i \in \{u,v,w\}} \Psi_1(y_i, x_i; \theta_1) - \sum_{(i,j) \in \{(u,v),(v,w)\}} \Psi_2(y_i, y_j, x_{i,j}; \theta_2) \right).$$

Above, $\mathbf{F}$ is set of randomly sampled regions of three connected vertices.

occlusion vs. non-occlusion must be inferred. We model the distance $\Delta_{i,p,p'}$ using a Laplacian probability distribution parameterized by $\alpha_{p,p'}$:

$$P_{\alpha_{p,p'}}(\Delta_{i,p,p'}) = \alpha_{p,p'} \exp(-\alpha_{p,p'}\Delta_{i,p,p'}), \quad \forall\ i \in R_{p,p'}, \tag{8}$$

where $R_{p,p'}$ is the set of (indices of) points that are on the boundary between the planes p and p'.

To form a 3d reconstruction, we will try to maximize the log-likelihood of d, λ, $\mathbf{Q}$ and α, given the MRF labeling of the planes and edges. More formally, we have:

$$\begin{array}{ll} \text{maximize}_{d,\lambda,\mathbf{Q},\alpha} & \sum_{(p,p')\in \mathrm{B}} \sum_{i\in R_{p,p'}} \log P_{\alpha_{p,p'}}(\|\mathbf{Q}_{i,p} - \mathbf{Q}_{i,p'}\|_2) \\ \text{subject to} & \mathbf{Q}_{i,p} = K^{-1}\mathbf{q}_i\lambda_{i,p} \ \ , \ \forall\ (i,p) \\ & d_p = \mathbf{n}_p^T K^{-1}\mathbf{q}_i\lambda_{i,p} \ \ , \ \forall\ (i,p) \\ & d_{\text{floor}} = c \ , \end{array} \tag{9}$$

where B is the set of pairs (p,p') of planes that share a common boundary in the image.

We apply an efficient alternating maximization algorithm to this optimization problem. For fixed α, maximizing the objective over d, λ and $\mathbf{Q}$ reduces to a linear program:

$$\begin{array}{ll} \text{minimize}_{d,\lambda} & \sum_{(p,p')\in \mathrm{B}} \sum_{i\in R_{p,p'}} w_{i,p,p'}\ |\lambda_{i,p} - \lambda_{i,p'}| \\ \text{subject to} & d_p = \mathbf{n}_p^T K^{-1}\mathbf{q}_i\lambda_{i,p} \ \ , \ \forall\ (i,p) \\ & d_{\text{floor}} = c \ , \end{array} \tag{10}$$

where $w_{i,p,p'} = \alpha_{p,p'}\|K^{-1}\mathbf{q}_i\|_2$. For fixed d, λ and $\mathbf{Q}$, we can maximize over α in closed form:[14]

$$\alpha_{i,j} = \frac{\sum_{i\in R_{p,p'}} 1}{\sum_{i\in R_{p,p'}} \|\mathbf{Q}_{i,p} - \mathbf{Q}_{i,p'}\|_2} \ . \tag{11}$$

We iterate updating d, λ and $\mathbf{Q}$; and updating α, until convergence.[15]

Sturm and Maybank's method—which relied on known occlusion edges—can roughly be viewed as a variant of our algorithm in which a Gaussian

[14] Using a heuristic reminiscent of Laplace smoothing, we actually add 0.5 to the denominator, and 5 to the numerator. This smooths the estimates, and also prevents a small denominator from causing $\alpha_{p,p'}$ from growing without bound. To help the search procedure, we also used a heuristic in which $\alpha_{\text{floor},p'}$ (and $\alpha_{p,p'}$ for horizontal edges) were initialized to be large. Edges that appeared clearly to be occluding, specifically ones parallel to the normal of a plane, were discarded from the optimization (or, less formally, had $\alpha_{p,p'}$ set to an infinitesimally small value).

[15] Other details: During the reconstruction, we also discard the ceiling plane. Also, all planes that were reconstructed as lying outside a reasonable range (a 10m × 10m × 50 m box in front of the camera) were considered outliers, and also discarded.

(instead of Laplacian) model with a fixed variance parameter (rather than the variable α) is used. [19] found Laplacians to be a superior model than Gaussians for modeling differences between distances. In our experiments (described in Section 5), we also find the Laplacian to outperform the Gaussian.

5 Experimental results

We applied our algorithm to a test set of 15 images obtained using a calibrated digital camera in 8 different buildings (all of which had fairly different interior decoration themes from each other, and from the building from which the training set images were taken). Since the test set buildings contained a diverse range of orthogonal geometries (boxes, doors, hallways, cabinets, etc.), we believe that the results we present are indicative of the algorithm's performance on images of new (Manhattan world) buildings and scenes.

Figures 9 shows the labeling obtained by the MRF on 6 images from the test set, as well as the resulting 3d reconstructions. Even in fairly complex environments or ones that do not perfectly respect the Manhattan world assumption, the algorithm is still able to label most of the planes correctly, and obtain reasonable 3d reconstructions.

We also evaluate the algorithm more formally. First, using a hand-labeling of the test set images, we measure the labeling error rate of the MRF. The overall accuracy of the MRF is 79.6%. Given that there are 6 possible labels for each pixel, random guessing would have obtained 16.7% accuracy. Table 1 shows a further breakdown of these results by planes and edges.[16] Although our precision on edges was surprisingly low, this appears to be a consequence of only a very small fraction of the pixels being edge pixels, and did not seem to significantly affect the final reconstruction performance.

Table 1. MRF labeling errors on test set images

	planes	edges
Recall	80.6%	65.7%
Precision	89.1%	29.0%

Using a careful hand-labeling of the test set images (including both plane orientations and occluding edges), we also generated a full ground-truth 3d reconstruction of the test set scenes. We then measured the average errors in the reconstructed distances, for pixels at different ground-truth distances from the camera. These statistics do not include planes that were discarded during the

[16] Recall is the fraction of plane/edge labels that we labeled correctly. Precision is, out of all the times we predicted a specific label, the fraction of times that the prediction was correct.

reconstruction, and thus might reflect a slightly overoptimistic performance metric, but nonetheless represents a fair comparison between the different algorithms.[17] The results are shown in Figure 8. We also compare the Laplacian model with a Gaussian one (that, similar to our procedure for learning $\alpha_{p,p'}$, tries to adapt its variance parameters), and with an implementation of Sturm and Maybank's algorithm that assumes there are no occluding edges. When performing 3d reconstruction using our MRF's output labels, the Laplacian model appears to perform best.

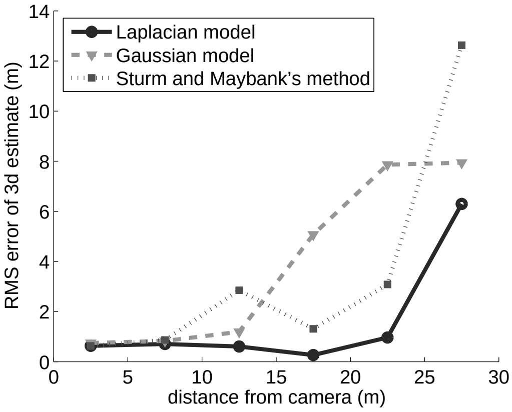


Fig. 8. Errors in 3d reconstructions, for pixels at different ground-truth distances from the camera.

6 Summary

We have presented an algorithm for fully automatic 3d reconstruction of indoor Manhattan world scenes from a single image. Our method uses an MRF to label each pixel as belonging to one of three plane orientations or one of three edge orientations. Given the MRF model's outputs, we use a Laplacian probabilistic model to infer a 3d reconstruction. Our experimental results show the algorithm performing well on a number of indoor scenes, even ones very different from the training set images. The work presented in this paper

[17] See footnote 15 for details. Given the MRF output, all three algorithms discard (the same) 4% of pixels as belonging to the ceiling; 22% of pixels labeled as edges (whose distance is truly ambiguous, since they can be reconstructed as lying on either of two planes); and under 1% as outliers (reconstructed as lying outside the box described in footnote 15).

was restricted to Manhattan worlds, and it remains an important problem to generalize these ideas to other scenes. More generally, we believe that monocular depth estimation holds significant promise, and it remains an important problem to develop algorithms that exploit other single-image cues for depth estimation.

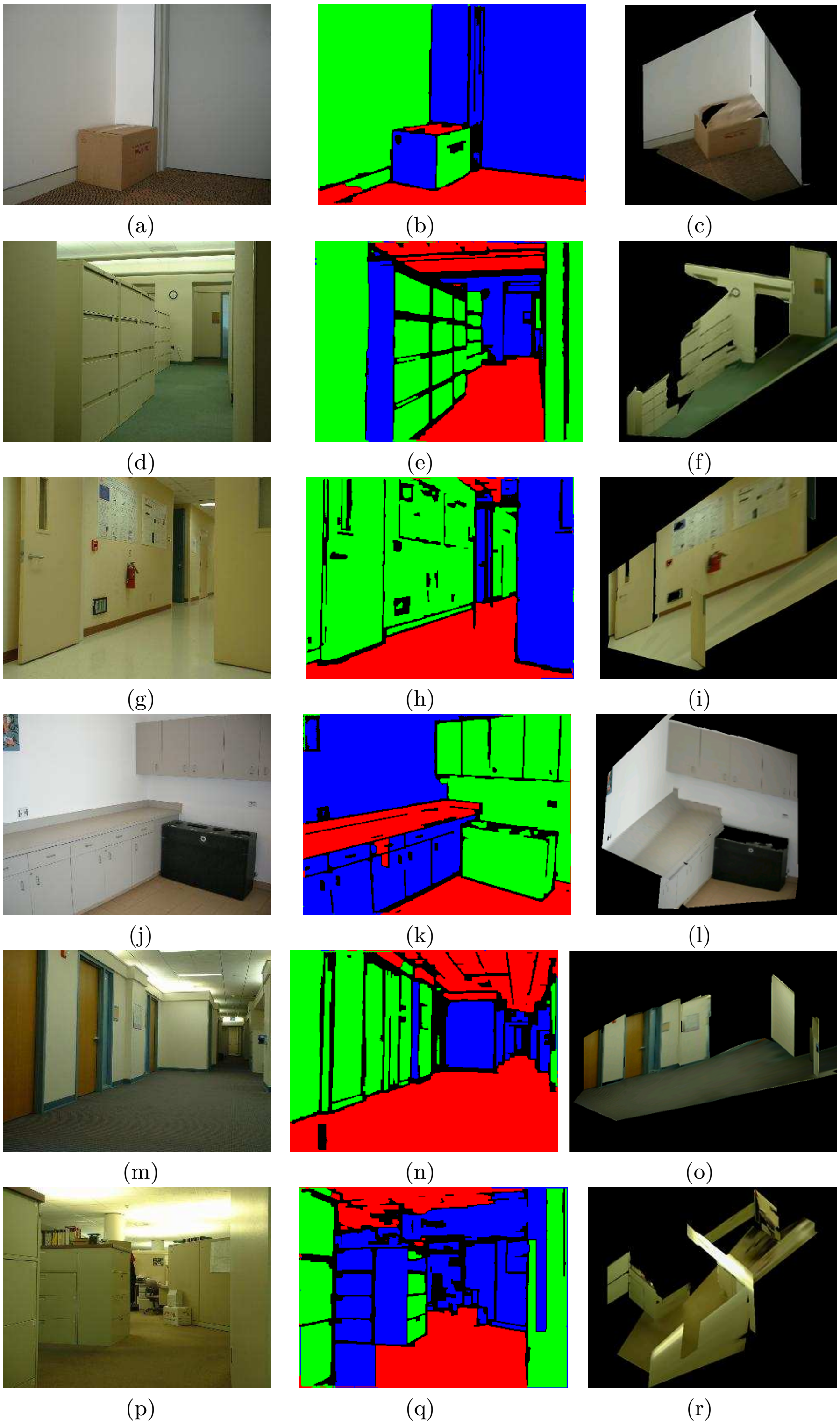
Acknowledgments

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

References

1. J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society, Series B*, 1974.
2. J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.
3. J. Coughlan and A.L. Yuille. Manhattan world: Compass direction from a single image by bayesian inference. In *IEEE International Conference on Computer Vision*, 1999.
4. J. Coughlan and A.L. Yuille. Manhattan world. *Neural Computation*, 15:1063–1088, 2003.
5. A. Criminisi, I. Reid, and A. Zisserman. Single view metrology. *International Journal of Computer Vision*, 40:123–148, 2000.
6. P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs. In *SIGGRAPH*, 1996.
7. E. Delage, H. Lee, and A. Y. Ng. A dynamic Bayesian network model for autonmous 3d reconstruction from a single indoor image. Unpublished manuscript, 2005.
8. P. Favaro and S. Soatto. Shape and radiance estimation from the information divergence of blurred images. In *European Conference on Computer Vision*, 2000.
9. Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59, 2004.
10. R. C. Gonzalez and R. E. Woods. *Digital Image Processing.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1992.
11. F. Han and S. C. Zhu. Bayesian reconstruction of 3d shapes and scenes from a single image. In *IEEE International Workshop on Higher-Level Knowledge in 3D Modeling and Motion Analysis*, pages 12–20, 2003.
12. F. Huang and Y. Ogata. Generalized pseudo-likelihood estimates for Markov random fields on lattice. *Annals of the Institute of Statistical Mathematics*, 2002.
13. A. Kosaka and A. C. Kak. Fast vision-guided mobile robot navigation using model-based reasoning and prediction of uncertainties. *CVGIP: Image Understanding*, 56:271–329, 1992.
14. J. Kosecka and W. Zhang. Video compass. In *European Conference on Computer Vision*, 2002.

15. P. Kovesi. Image features from phase congruency. *Videre: A Journal of Computer Vision Research*, 1, 1999.
16. P. D. Kovesi. MATLAB and Octave functions for computer vision and image processing. School of Computer Science & Software Engineering, The University of Western Australia.
    Available from: http://www.csse.uwa.edu.au/∼pk/research/matlabfns/.
17. E. Lutton, H. Maitre, and J. Lopez-Krahe. Contribution to the determination of vanishing points using hough transform. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 16:430–438, 1994.
18. J. Michels, A. Saxena, and A. Y. Ng. High-speed obstacle avoidance using monocular vision and reinforcement learning. In *International Conference on Machine Learning*, 2005.
19. A. Saxena, S. Chung, and A. Y. Ng. Learning depth from single monocular images. In *Neural Information Processing Systems*, 2005.
20. 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 *IEEE International Conference on Computer Vision and Pattern Recognition*, 2004.
21. H.-Y. Shum, M. Han, and R. Szeliski. Interactive construction of 3d models from panoramic mosaics. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 1998.
22. P. F Sturm and S. J. Maybank. A method for interactive 3d recontruction of piecewise planar objects from single images. In *British Machine Vision Conference*, 1999.
23. C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9:137–154, 1992.
24. M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. Tree-based reparameterization framework for analysis of sum-product and related algorithms. *IEEE Trans. Information Theory*, 49(5):1120–1146, 2003.
25. R. Zhang, P.-S. Tsai, J. E. Cryer, and M. Shah. Shape from shading: A survey. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21:690–706, 1999.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

(m) (n) (o)

(p) (q) (r)

Fig. 9. Inferred 3d reconstructions of test set indoor scenes. Left column: Input image. Middle column: Labeling generated by MRF (red, green and blue correspond to the three plane orientations; black corresponds to all three edge orientations). Right column: Resulting 3d reconstructions.