# Manhattan scene understanding using monocular, stereo, and 3D features

**Conference Paper** *in* Proceedings / IEEE International Conference on Computer Vision. IEEE International Conference on Computer Vision · November 2011

DOI: 10.1109/ICCV.2011.6126501 · Source: DBLP



**3 authors**, including:

Alex Flint
Cruise Automation
**9** PUBLICATIONS **390** CITATIONS

SEE PROFILE

# Manhattan Scene Understanding Using Monocular, Stereo, and 3D Features

Alex Flint, David Murray, and Ian Reid
Active Vision Laboratory
Oxford University, UK
{alexf,dwm,ian}@robots.ox.ac.uk

## Abstract

*This paper addresses scene understanding in the context of a moving camera, integrating semantic reasoning ideas from monocular vision with 3D information available through structure–from–motion. We combine geometric and photometric cues in a Bayesian framework, building on recent successes leveraging the indoor Manhattan assumption in monocular vision. We focus on indoor environments and show how to extract key boundaries while ignoring clutter and decorations. To achieve this we present a graphical model that relates photometric cues learned from labeled data, stereo photo–consistency across multiple views, and depth cues derived from structure–from–motion point clouds. We show how to solve MAP inference using dynamic programming, allowing exact, global inference in* $\sim$100 *ms (in addition to feature computation of under one second) without using specialized hardware. Experiments show our system out–performing the state–of–the–art.*

## 1. Introduction

Over the past decade, computer vision researchers working with monocular images have pursued substantially different research agendas to those working with multiple views. The focus for monocular images has increasingly been to infer high–level facts about the world, such as the locations of and interactions between objects, semantic scene categories, and the spatial layout of the environment. In contrast, much of the work concerning multiple views has focused on reconstructing metric scene structure and camera poses using techniques such as structure–from–motion, stereo, and multiple–view stereo.

In this paper we leverage multiple view geometry for image understanding purposes. We assume a moving camera with a structure–from–motion system estimating its trajectory, and show how to infer semantically meaningful models of the environment. We focus on the *indoor Manhattan representation*[14, 7], in which the world is modeled in terms of floor, wall, and ceiling surfaces. This representation captures many semantically meaningful aspects of the environment, including (i) scale: the distance from floor to ceiling suggests a scale for distances in the environment; (ii) boundaries: walls constrain movement in the environment and suggest locations for doors, windows, and other objects; (iii) gravity: the orientation of the ground plane implies the direction in which gravity operates, which constrains the arrangement of objects resting upon one another; and (iv) shape: the organisation of walls in an environment suggests a functional category (such as "kitchen" or "office").

Figure 1. Automatic reconstructions from our system.

Indoor Manhattan models are useful because they capture these properties explicitly, whereas to extract such properties from a dense polygonal mesh would require additional non–trivial inference after reconstruction. Our approach infers semantic properties of the scene directly from multiple–view data, without an intermediate dense reconstruction step. This makes sense if, as in our case, the semantic properties constitute the ultimate goal of the system. Of course, if a photo–realistic reconstruction is itself the end goal then our approach is not suitable.

We build on recent work highlighting the efficacy of the indoor Manhattan representation for single view reconstruction [14, 7]. Previous work employed a set of heuristics as a cost function and was limited to monocular images. We give a fully Bayesian account in which information from image features, stereo, and 3D point clouds is integrated into a single MAP optimization, and we learn all parameters from labeled examples. We show that MAP inference in our model can be solved exactly and efficiently ($\sim$100 ms per frame) using a generalization of the dynamic programming algorithm of [7].

The remainder of this paper is organised as follows. Sec-

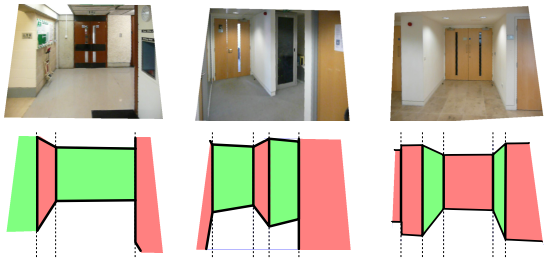

Figure 2. Three input images and the indoor Manhattan models we seek. Notice how each image column intersects exactly one wall.

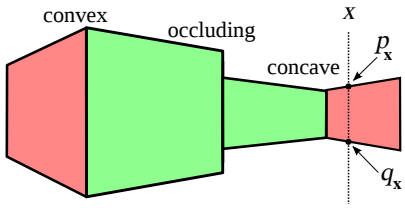


Figure 3. Each corner in an indoor Manhattan environment can be categorized as concave, convex, or occluding. Each vertical line intersects exactly one wall segment, the top and bottom of which we denote $\boldsymbol{p}_x$ and $\boldsymbol{q}_x$ respectively.

tion two places our contribution in context with related work. We introduce our model in section three, then we describe inference in section four and learning in section five. We give experimental results in section six, followed by concluding remarks in section seven.

## 2. Background

The Manhattan world assumption was introduced by Coughlan and Yuille[4] over a decade ago and has seen increasing attention in the computer vision literature over past years [4, 13, 14, 10, 7]. Furukawa *et al*. [10] proposed a Manhattan–world stereo algorithm based on graph cuts. While their approach is concerned with dense photo–realistic reconstructions, ours is intended to capture semantic properties of the scene using a concise representation. The output of their approach — a polygonal mesh — has no immediate semantic interpretation, whereas our models, though less detailed, come packaged with a direct interpretation. A by–product is efficiency: we count computation time in hundreds of milliseconds, where as Furukawa *et al*. report waiting more than an hour.

Another approach to interpreting Manhattan worlds is to model scenes as a union of cuboids. This approach has a long history beginning with Roberts' 1965 thesis [15], and has recently been revisited using modern probabilistic techniques [11, 17].

Lee *et al*. [14] first proposed indoor Manhattan models (a sub–class of general Manhattan models) for monocular reconstructions. They used a branch–and–bound algorithm together with a line–sweep heuristic for approximate inference. Flint *et al*. [7] employed a similar model but showed a dynamic programming algorithm that performed exact inference in polynomial time. In earlier work [8] Flint *et al*. also demonstrated Manhattan reconstructions integrated with a SLAM system, but this work inferred models from single frames and then extrapolated these forward in time. In contrast, our work incorporates both multiple view geometry and 3D points directly into a joint inference procedure. We also learn parameters in a Bayesian framework, where as neither Lee nor Flint utilized training data in any form.

Felzenszwalb and Veksler [6] posed the reconstruction problem in terms of energy minimization, which they showed could be solved using dynamic programming, while Barinova *et al*. [2] modeled outdoor scenes using a CRF. However, these approaches do not permit strong geometric constraints and so cannot be extended to multiple views.

Semantic scene understanding has, broadly speaking, seen less attention within the multiple view community. The CamVid [3] database of outdoor videos with semantic segmentations is an important and encouraging exception. Brostow *et al*. [3] showed that simple structure–from–motion cues lead to pleasing segmentations. Sturgess *et al*. [1] extended this approach to a CRF framework. We compare our method with this approach in section 6.

## 3. Proposed Model

In this section we describe the indoor Manhattan model. We consider three sensor modalities: monocular image features, stereo features, and 3D point clouds. For each we present a generative model relating observed features to the Manhattan scene structure, which we denote $M$. For each sensor modality we show that MAP inference can be reduced to maximization over a payoff function $\pi(x, y)$. This allows us to present a unified dynamic programming solution in section 4, which efficiently solves MAP inference for all three sensor modalities.

General Manhattan environments have structural surfaces oriented in three cardinal orientations. *Indoor* Manhattan environments are a special case that consist of a floor plane, a parallel ceiling plane, and a set of vertical walls extending between them. Each wall extends all the way from the floor to ceiling, and walls meet at vertical edges. We always consider environments observed from a camera located between the floor and ceiling. Since each wall extends from floor to ceiling, indoor Manhattan environments always project as a linear chain of walls in the image, as shown in figure 1. Further, the edges at which adjacent walls meet can be categorized as concave, convex, or occluding, as illustrated in figure 3 and discussed further in [14].

We assume that vanishing points for the three Manhattan directions are given. We use the vanishing point detector described by Zhang *et al*. [13] in the monocular setting

and that of [8] in the multiple view setting. It will greatly simplify the remainder of this paper if we can assume that vertical lines in the world appear vertical in the image. To this end we apply the simple rectification procedure of [8].

We now describe our parametrization for indoor Manhattan models. Let the image dimensions be $N_x \times N_y$. Following rectification, the vertical seams at which adjacent walls meet project to vertical lines, so each image column intersects exactly one wall segment. Let the top and bottom of the wall in column $x$ be $\boldsymbol{p}_x = (x, y_x)$ and $\boldsymbol{q}_x = (x, y'_x)$ respectively (depicted in figure 3). Since each $\boldsymbol{p}_x$ lies on the floor plane and each $\boldsymbol{q}_x$ lies on the ceiling plane, we have

$$\boldsymbol{p}_x = H\boldsymbol{q}_x \ . \tag{1}$$

where $H$ is a planar homology [5]. We show how to recover $H$ in section 3.5. Once $H$ is known, any indoor Manhattan model is fully described by the values $\{y_x\}$, leading to the simple parametrization,

$$M = \{y_x\}_{x=1}^{N_x} \ . \tag{2}$$

We query this parametrization as follows. To check whether a pixel $(x_0, y_0)$ lies on a vertical or horizontal surface we simply need to check whether $y_0$ is between $y_{x_0}$ and $y'_{x_0}$. If we know the 3D position of the floor and ceiling planes then we can recover the depth of every pixel as follows. If the pixel lies on the floor or ceiling then we simply back–project a ray onto the corresponding plane. If not, we back–project onto the vertical plane defining the wall at that column (the depth of which we can recover from $y_{x_0}$). Note in particular that the orientation and depth of a pixel can be recovered from just the floor/wall intersection in its column; this will be important in later sections.

We now turn to the optimization framework that each subsequent section will feed into. Let $\{c_i\}$ index the columns at which neighbouring walls meet in $M$. We define the payoff for $M$ as

$$\Pi(M) = \sum_{x=1}^{N_x} \pi(x, y_x) - \sum_i \gamma(c_i) \tag{3}$$

where the payoff matrix $\pi$ assigns payoffs for models with floor/wall intersections that pass through each pixel, and $\gamma$ is a per–corner regulariser which penalizes complex models. Note that the value of $\pi(x, y)$ is *not* restricted to dependence on pixel $(x, y)$, nor even to a local region about that pixel; indeed, the payoff functions described in the following sections incorporate image evidence from widely separated image regions.

### 3.1. Monocular features

To infer indoor Manhattan models from monocular images we assume the graphical model shown in figure 4. We

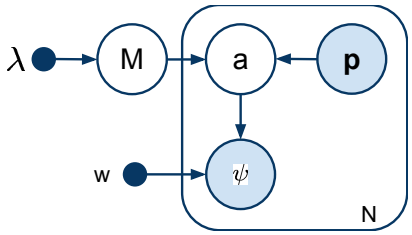


Figure 4. The graphical model relating building structures $M$ to monocular image features $\boldsymbol{\psi}$. $\boldsymbol{p} = (x, y)$ is a pixel location and $a$ is the orientation predicted (deterministically) by $M$ at $\boldsymbol{p}$.

turn first to the prior $P(M \mid \boldsymbol{\lambda})$. For a model with $n_1$ concave corners, $n_2$ convex corners, and $n_3$ occluding corners (*c.f.* figure 3), our prior on models is

$$P(M \mid \boldsymbol{\lambda}) \quad = \quad \frac{1}{Z}{\lambda_1}^{n_1}{\lambda_2}^{n_2}{\lambda_3}^{n_3} \tag{4}$$

which corresponds to a fixed probability for "events" corresponding to each type of corner and penalizes models for additional complexity. $Z$ is a normalizing constant.

Our model includes hidden orientation variables $a_i \in \{1, 2, 3\}$ for each pixel, with values corresponding to the three Manhattan orientations (shown as red, green, and blue regions in figure 1). As described in section 3, $a$ is deterministic given the model $M$. We assume a linear likelihood for pixel features $\boldsymbol{\psi}$,

$$P(\boldsymbol{\psi} \mid a) = \frac{{\boldsymbol{w}_a}^T \boldsymbol{\psi}}{\sum_j {\boldsymbol{w}_a}^T \boldsymbol{\psi}_j} \ . \tag{5}$$

We now derive MAP inference. The posterior on $M$ is

$$P(M \mid \Psi) = \eta P(M) \prod_i P(\boldsymbol{\psi}_i \mid a_i^*) \tag{6}$$

where $a_i^*$ is the orientation deterministically predicted by model $M$ at pixel $\boldsymbol{p}_i$ and $\eta$ is a normalizing constant. We have omitted $P(a_i \mid M)$ since it equals 1 for $a_i^*$ and 0 otherwise. Taking logarithms,

$$\begin{aligned} \log P(M \mid \Psi) = n_1\lambda'_1 + n_2\lambda'_2 + n_3\lambda'_3 \\ + \sum_i \log P(\boldsymbol{\psi}_i \mid a_i^*) + k \end{aligned} \tag{7}$$

where $\lambda'_3 = \log \lambda_3$ and similarly for the other penalties, and $k$ corresponds to the normalizing denominators in (6) and (4), which we henceforth drop since it makes no difference to the optimization to come. We can now put (7) into payoff form (3) by writing

$$\begin{aligned} \pi_{\mathsf{mono}}(x, y_x) &= \sum_{y'} \log P(\boldsymbol{\psi}_i \mid a_i^*) \\ \gamma_{\mathsf{mono}}(c) &= -\lambda_c \end{aligned} \tag{8}$$

where $\lambda_c$ is one of $\lambda_1$, $\lambda_2$, or $\lambda_3$ according to the category of corner $c$. We show how to maximize payoffs of this form in section 4, which will allow us to solve MAP inference.

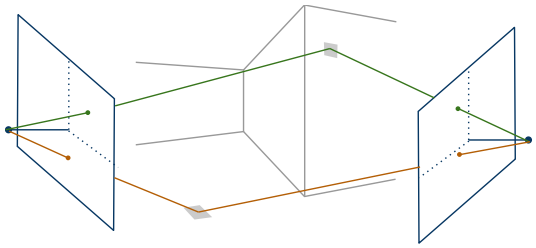

Figure 5. Pixel correspondences across multiple views are computed by back–projection onto the model $M$ followed by re–projection into auxiliary views.

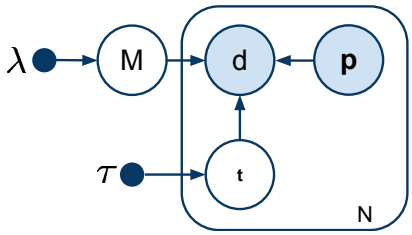


Figure 6. The graphical model relating indoor Manhattan models to 3D points. The hidden variable $t$ indicates whether the point is inside, outside, or coincident with the model.

### 3.2. Multiple view features

We now formulate the payoff function $\pi_{\text{stereo}}$ for the case that multiple views of the scene are available. We assume one base frame $I_0$ and $M$ auxiliary frames $I_1, \ldots, I_M$. We assume that poses are given for each camera, as output for example by a structure–from–motion system, and that cameras are calibrated. We normalize images intensities to zero mean and unit variance.

Intuitively, we treat inference in this settings as follows. We consider models $M$ in terms of their projection into $I_0$. We explained in section 3 that models parametrized in image coordinates specify unique 3D models. Any hypothesized model can therefore be re–projected into auxiliary frames, giving pixel–wise correspondences between frames as shown in figure 5. From this we compute a photo–consistency measure $\text{PC}(\cdot)$, which provides the likelihood $P(\{I_k\} \mid M)$. The prior remains as in (4).

Optimizing over photo–consistency has been standard in the stereo literature for several decades [16]; our contribution is to show that (i) in the particular case of indoor Manhattan models, photo–consistency can be expressed as a payoff matrix; (ii) that we can therefore perform efficient and exact global optimization; and (iii) that this fits naturally within a Bayesian framework alongside monocular and 3D features.

Our approach could also be cast as solving the general stereo problem where in place of priors based on various pixel–wise norms, our prior assigns zero probability to all non–indoor–Manhattan reconstructions.

Let $\text{reproj}_k(\boldsymbol{p}; M)$ be the re–projection of pixel $\boldsymbol{p}$ from the base frame $I_0$ into auxiliary frame $I_k$ via model $M$. Then

$$\log P(\{I_k\} \mid M) = \sum_{\boldsymbol{p} \in I_o} \sum_{k=1}^{M} \text{PC}(\boldsymbol{p}, \text{reproj}_k(\boldsymbol{p}, M)) \,, \quad (9)$$

where in our experiments $\text{PC}(\boldsymbol{p}, \boldsymbol{q})$ is the sum of squared differences between pixels $\boldsymbol{p}$ and $\boldsymbol{q}$.

We explained in section 3 that the depth of each pixel can be recovered from the location of the floor/wall intersection $y_x$ in column $x$. Hence we can replace $\text{reproj}_k(\boldsymbol{p}; M)$ with $\text{reproj}_k(\boldsymbol{p}; y_x)$ and write

$$\pi_{\text{stereo}}(x, y_x) = \sum_{y=1}^{N_y} \sum_{k=1}^{M} \text{PC}(\boldsymbol{p}, \text{reproj}_k(\boldsymbol{p}, y_x)) \,, \quad (10)$$

where $\boldsymbol{p} = (x, y)$. To see this, substitute (10) into (3) and observe that the result is precisely (9).

Note that the column–wise decomposition (10) neither commits us to optimizing over columns independently, nor to ignoring interactions between columns. Such interactions come into effect when we optimize over the full payoff matrix in section 4, and our results will show that widely separated image regions often interact strongly. The derivations in this section follow deductively from the indoor Manhattan assumption; the only approximation is the following.

**Occlusions**. We have ignored self–occlusions in (9). For short baselines (such as frames sampled over a few seconds from a moving camera), this is unproblematic since indoor environments tend to be mostly convex from any single point of view. Even in highly non–convex environments our system achieves excellent results by integrating 3D and monocular features, and enforcing strong global consistency, as will be shown in section 6.

### 3.3. 3D features

In this section we explore the context in which a 3D point cloud is available during inference. The point clouds generated by structure–from–motion systems are typically too sparse for direct reconstruction, but can provide useful cues alongside monocular and stereo data.

Our graphical model for 3D data is depicted in figure 6. The model $M$ is sampled according to the prior (4), then depth measurements $d_i$ are generated for pixels $\boldsymbol{p}_i$. Many such measurements will correspond to clutter or measurement errors, rather than to the walls represented by $M$. Our model captures this uncertainty explicitly through the latent variable $t_i$, which has following interpretation. If $t_i = \textsf{ON}$ then $d_i$ corresponds to some surface represented explicitly in $M$. Otherwise, either $t_i = \textsf{IN}$, meaning some clutter object within the room was measured, or $t_i = \textsf{OUT}$, in which case an object outside the room was measured, such

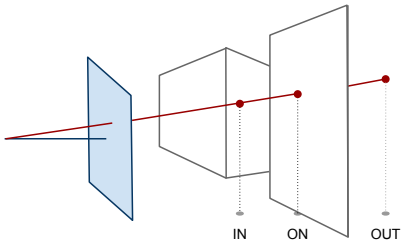


Figure 7. Depth measurements $d_i$ might be generated by a surface in our model (represented by $t_i = \mathsf{ON}$) or by an object inside or outside the environment (in which case $t_i = \mathsf{IN}, \mathsf{OUT}$ respectively).

as through a window. The likelihoods we use are

$$P(d \mid \boldsymbol{p}, M, \mathsf{IN}) = \begin{cases} \alpha, & \text{if } 0 < d < r(\boldsymbol{p}; M) \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

$$P(d \mid \boldsymbol{p}, M, \mathsf{OUT}) = \begin{cases} \beta\,, & \text{if } r(\boldsymbol{p}; M) < d < N_d \\ 0\,, & \text{otherwise} \end{cases} \tag{12}$$

$$P(d \mid \boldsymbol{p}, M, \mathsf{ON}) = \mathcal{N}(d\,;\, r(\boldsymbol{p}; M), \sigma)\,. \tag{13}$$

where $\alpha$ and $\beta$ are determined by the requirement that the probabilities sum to 1 and $r(\boldsymbol{p}; M)$ denotes the depth predicted by $M$ at $\boldsymbol{p}$. We compute likelihoods on $d$ by marginalizing,

$$P(d \mid \boldsymbol{p}, M) = \sum_t P(d \mid \boldsymbol{p}, M, t) P(t)\,, \tag{14}$$

where the prior $P(t)$ is a look–up table with three entries denoted $\tau_{\mathsf{IN}}$, $\tau_{\mathsf{OUT}}$, and $\tau_{\mathsf{ON}}$.

As explained in section 3, computing the depth of the model at pixel $\boldsymbol{p}$ requires knowledge only of the floor/wall intersection $y_x$ in column $x$, so we substitute

$$P(d \mid \boldsymbol{p}, y_x) = P(d \mid \boldsymbol{p}, M)\,. \tag{15}$$

Let $D$ denote all depth measurements, $\boldsymbol{P}$ denote all pixels, and $\mathcal{D}_x$ contain indices for all depth measurements in column $x$. Then

$$P(M \mid D, \boldsymbol{P}) = P(M) \prod_x \prod_{i \in \mathcal{D}_x} P(d_i \mid \boldsymbol{p}_i, y_x) \tag{16}$$

$$\log P(M \mid D, \boldsymbol{P}) = P(M) + \sum_x \Big( \sum_{i \in \mathcal{D}_x} \log P(d_i \mid \boldsymbol{p}_i, y_x) \Big)\,, \tag{17}$$

which we write in payoff form as

$$\pi_{\mathsf{3D}}(x, y_x) = \sum_{i \in \mathcal{D}_x} \log P(d_i \mid \boldsymbol{p}_i, y_x) \tag{18}$$

and the penalty function $\gamma$ remains as in (8).

### 3.4. Combining features

We combine photometric, stereo, and 3D data into a joint model by assuming conditional independence given $M$,

$$\begin{aligned} &P(M \mid X_{\mathsf{mono}}, X_{\mathsf{stereo}}, X_{\mathsf{3D}}) = \\ &\quad P(M) P(X_{\mathsf{mono}} \mid M) P(X_{\mathsf{stereo}} \mid M) P(X_{\mathsf{3D}} \mid M) \end{aligned} \tag{19}$$

Taking logarithms leads to summation over payoffs,

$$\pi_{\mathsf{joint}}(\boldsymbol{x}) = \pi_{\mathsf{mono}}(\boldsymbol{x}) + \pi_{\mathsf{stereo}}(\boldsymbol{x}) + \pi_{\mathsf{3D}}(\boldsymbol{x})\,. \tag{20}$$

### 3.5. Resolving the floor and ceiling planes

We resolve the equation of the floor and ceiling planes as follows. If $C$ is the camera matrix for any frame and $\boldsymbol{v_v}$ is the vertical vanishing in that frame, then $\boldsymbol{n} = C^{-1}\boldsymbol{v_v}$ is normal to the floor and ceiling planes. We sweep a plane with this orientation through the scene, recording at each step the number of points within a distance $\delta$ of the plane ($\delta$=0.1% of the diameter of the point cloud in our experiments). We take as the floor and ceiling planes the minimum and maximum locations such that the plane contains at least 5 points. We found that this simple heuristic worked without failure on our training set.

Let the two non–vertical vanishing points be $\boldsymbol{v_l}$ and $\boldsymbol{v_r}$ and let $\boldsymbol{h} = \boldsymbol{v_l} \times \boldsymbol{v_r}$. Select any two corresponding points $\boldsymbol{x_f}$ and $\boldsymbol{x_c}$ on the floor and ceiling planes respectively. Then the Manhattan homology defined in (1) is given by

$$H = I + \mu \frac{\boldsymbol{v_v} \boldsymbol{h}^T}{\boldsymbol{v_v} \cdot \boldsymbol{h}}\,, \tag{21}$$

where $\mu = < \boldsymbol{v_v}, \boldsymbol{x_c}, \boldsymbol{x_f}, \boldsymbol{x_c} \times \boldsymbol{x_f} \times \boldsymbol{h} >$ is the characteristic cross ratio of $H$.

## 4. Inference

We have reduced MAP inference to optimization over a payoff matrix:

$$\hat{M} = \operatorname*{argmax}_M \sum_x \pi(x, y_x) - \sum_i \gamma(c_i) \tag{22}$$

In previous work [7] we showed that if an indoor Manhattan model $M$ is optimal over image columns $[1, x]$, then the "cropped" model $M'$, obtained by restricting $M$ to the sub–interval $[1, x']$ $x' < x$, must itself be optimal over that sub–interval. This permits a dynamic programming solution in which $\hat{M}$ is built up from left to right.

Our algorithm differs from that of [7] in the following respects. First, we optimize over general payoff matrices of the form (3); whereas neither $\pi_{\mathsf{stereo}}$ nor $\pi_{\mathsf{3D}}$ decomposes as assumed in [7]. Second, we do not include the number of corners as a state variable, but instead accumulate penalties directly into the objective function, which reduces complexity by $O(K)$ where $K$ is the number of walls in the model. For completeness we give revised recurrence relations in an appendix.

## 5. Training

In this section we address the learning of model parameters from labeled training data. We turn first to the parameter $\boldsymbol{w}$ relating photometric features to the orientation variables $a_i$ via (5). We employ the following bootstrapping algorithm to learn $\boldsymbol{w}$. We begin by sampling $k$ pixels at random from the images in the training set and use these to train a classifier (in our case a multi–class SVM with three classes) for the task of mapping pixel features $\boldsymbol{\psi}$ to orientations $a$. We then run the complete inference procedure on the entire training set, using the current classifier to evaluate the log–likelihood (8). We compute a pixel–wise loss $l_{\boldsymbol{p}}(\hat{M}, M)$ with respect to ground truth. In our experiments $l_{\boldsymbol{p}}$ is the relative depth error,

$$l_{\boldsymbol{p}}(\hat{M}, M) = \left| \frac{r(\boldsymbol{p}; \hat{M}) - r(\boldsymbol{p}; M)}{r(\boldsymbol{p}; M)} \right| . \tag{23}$$

We then sample $k$ additional pixels to be added to the training set, where each pixel is selected with probability proportional to the loss (23), then re–train the pixel classifier and repeat to convergence. That is, we add pixels at which *the image–level inference procedure* is making the greatest mistakes, which biases learning towards portions of the training set where the inference process as a whole, rather than the pixel–level classifier, is making mistakes.

**Model prior parameters $\boldsymbol{\lambda}$.** We assign a beta distribution with $\alpha = \beta = 1$ as hyper–prior for $\boldsymbol{\lambda}$. MAP estimates are then given by

$$\hat{\lambda_k} = \frac{\mathbb{E}[n_k]}{\mathbb{E}[n_k] + 1} \tag{24}$$

where expectations are over the training set.

**3D indicators**. We assign a uniform hyper–prior to all $\boldsymbol{\tau}$ representing a valid probability distribution (*i.e.* positive and unit sum), then perform gradient descent on the posterior $P(\boldsymbol{\tau} \mid \{M, \{p_i, x_i\}\})$.

## 6. Results

Our data–set consists of 18 manually annotated video sequences of indoor scenes averaging 59 seconds in duration. We sample frames at one second intervals and divide frames into consecutive groups of 3 (one base frame and two auxiliary frames). Our training set consists of 150 such triplets generated from 8 different sequences. Our test set contains 204 triplets from the remaining 10 sequences. No sequence appears in both the training and test sets.

To acquire ground truth data we reconstructed camera trajectories using structure–from–motion software (we use the PTAM system of Klein and Murray [12]) then manually specify the ground truth floor–plan. Recall that we seek to recover the *boundaries* of the environment, whether or not they are visible at every point. When our algorithm ignores clutter within a room, we consider that a *success*.

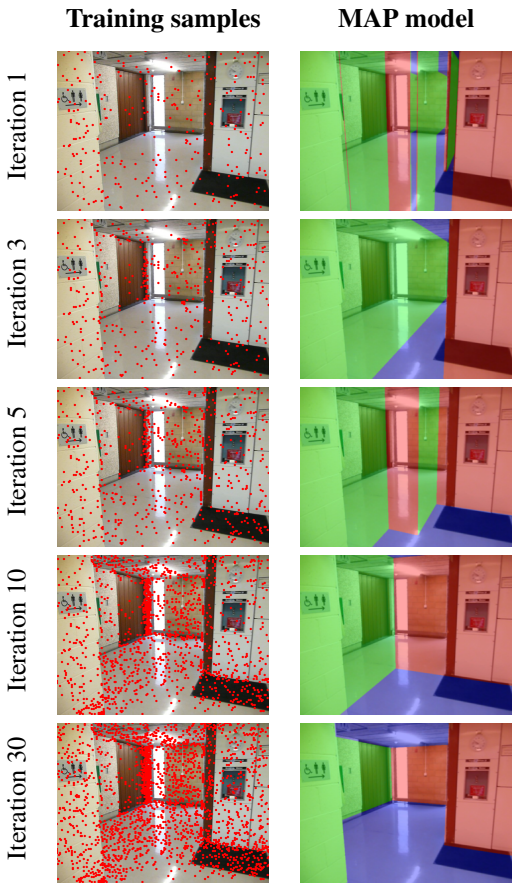


Figure 8. Snapshots of our bootstrap learning algorithm. The left column the pixels that the SVM was trained on in each iteration and the right column shows the corresponding MAP model. Each iteration injects incorrect pixels into the training set, leading to a concentration about surface boundaries since these locations are the most often confused by our model. This corresponds to the intuition that pixels near surface boundaries are the most "important" for the SVM to correctly classify since our model will leverage global consistency to "fill in the blanks" in other regions.

The monocular features $\boldsymbol{\psi}_i$ consist of 3 RGB channels, 3 HSV channels, 24 Gabor filters (4 scales, 6 orientations), and 3 binary line sweep features [14]. For stereo we use patches of size $5 \times 5$.

We compute two error metrics: the labeling accuracy, which is the proportion of all pixels that were labeled with the correct orientation, and the mean relative depth error (23). While the latter better captures similarity to the ground truth, not all the systems we compare against have direct 3D interpretations and in such cases we must compare on labeling accuracy.

To the best of our knowledge, there is no previously published work on precisely this problem (indoor–Manhattan reasoning from multiple views) so we compare with two al-

ternative systems, though neither comparison is ideal.

Our first comparison is with the approach of Brostow *et al*. [3], who performed semantic segmentation by training a per–pixel classifier on structure–from–motion cues. Our implementation of their system uses exactly the features they describe, with classes corresponding to the three Manhattan orientations. While they trained a randomized forest, we trained a multi–class SVM because a reliable SVM library was more readily available to us. Given the margin between our results it is unlikely that a different classifier would significantly change the outcome.

The second comparison is with the monocular approach of Lee *et al*. [14]. One would of course expect a multiple view approach to outperform a monocular approach, but as one of the very few previous approaches to have explicitly leveraged the indoor Manhattan assumption we feel this comparison is important to demonstrate the benefit of a Bayesian framework and integration of stereo and 3D cues.

The performance of each system is shown in figure 9. Our system significantly out–performs both others. Even when restricted to monocular features, our system outperforms [3], which has access to 3D cues. This reflects the utility of global consistency and the indoor Manhattan representation in our approach.

The initialization procedure of [14] fails for 31% of our training images, so at the bottom of figure 9 we show results for their system after excluding these images. Labeling accuracy increases to within 3% of our monocular–only results, though on the depth error metric a margin of 10% remains. This illustrates the effect of our training procedure, which optimizes for the depth error.

Figure 9 also shows that joint estimation is superior to using any one sensor modality alone. Anecdotally we find that using 3D cues alone often fails within large textureless regions in which the structure–from–motion system failed to track any points, whereas stereo or monocular cues alone often perform better in such regions but can lack precision at corners and boundaries.

Figure 11 shows timing results for our system. For each triplet of frames, our system requires on average less than one second to compute features for all three frames and less than 100 milliseconds to perform optimization.

## 7. Conclusion

We have presented a Bayesian framework for scene understanding in the context of a moving camera. Our approach draws on the indoor Manhattan assumption introduced for monocular reasoning and we have shown that techniques from monocular and stereo vision can be integrated with 3D data in a coherent Bayesian framework.

[1]This row excludes cases for which [14] was unable to find overlapping lines during initialization.

| Algorithm | Mean depth error (%) | Labeling accuracy (%) |
|---|---|---|
| Our approach (full) | **14.5** | **75.5** |
| Stereo only | 17.4 | 69.5 |
| 3D only | 15.2 | 71.1 |
| Monocular only | 24.8 | 69.2 |
| Brostow *et al*. [3] | | 40.6 |
| Lee *et al*. [14] | 79.8 | 45.5 |
| excluding failures[1] | 34.1 | 66.2 |

Figure 9. Performance on our data–set. Labeling accuracy is the percentage of correctly labeled pixels over the data–set, and depth error is a per–pixel average of (23).

In future work we intend to use indoor Manhattan models to reason about objects, actions, and scene categories. We also intend to investigate structural SVMs for learning parameters, which may allow us to relax the conditional independence assumptions between sensor modalities.

## 8. Appendix

**Recurrence relations for MAP inference.** Let $f_{\mathsf{out}}(x, y, a)$, $1 \leq x \leq N_x$, $1 \leq y \leq N_y$, $a \in \{1, 2\}$ be the maximum payoff for any indoor Manhattan model $M$ spanning columns $[1, x]$, such that (i) $M$ contains a floor/wall intersection at $(x, y)$, and (ii) the wall that intersects column $x$ has orientation $a$. Then $f_{\mathsf{out}}$ can be computed by recursive evaluation of the recurrence relations,

$$f_{\mathsf{out}}(x, y, a) = \max_{a' \in \{1,2\}} \begin{cases} f_{\mathsf{up}}(x, y-1, a') - \gamma(x) \\ f_{\mathsf{down}}(x, y+1, a') - \gamma(x) \\ f_{\mathsf{in}}(x, y, a') - \gamma(x) \end{cases} \tag{25}$$

$$f_{\mathsf{up}}(x, y, a) = \max\Big(f_{\mathsf{in}}(\cdot), f_{\mathsf{up}}(x, y-1, a)\Big) , \tag{26}$$

$$f_{\mathsf{down}}(x, y, a) = \max\Big(f_{\mathsf{in}}(\cdot), f_{\mathsf{down}}(x, y+1, a)\Big) , \tag{27}$$

$$f_{\mathsf{in}}(x, y, a) = \max_{x' < x}\Big(f_{\mathsf{out}}(x', y', a) + \Delta\Big) , \tag{28}$$

$$\Delta = \sum_{i=x'}^{x} \pi(i, y') . \tag{29}$$

Here we have treated $f_{\mathsf{in}}$, $f_{\mathsf{up}}$, and $f_{\mathsf{down}}$ simply as notational placeholders; for their interpretations in terms of sub–problems see [7]. Finally, the base cases are

$$f_{\mathsf{out}}(0, y, a) = 0 \qquad \forall y, a \tag{30}$$

$$f_{\mathsf{up}}(x, 0, a) = \infty \qquad \forall x, a \tag{31}$$

$$f_{\mathsf{down}}(x, N_x, a) = \infty \qquad \forall x, a . \tag{32}$$

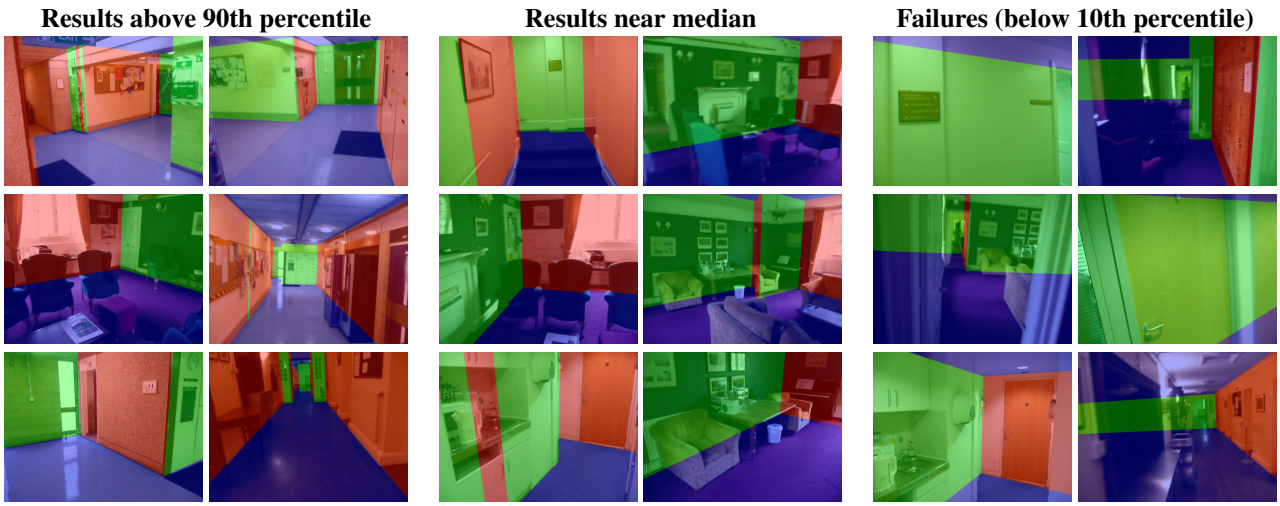


Figure 10. Models output from our system. The left column shows results above the 90th percentile of performance (relative depth error), the middle column shows results near median performance, and the right column shows failure cases.

| Component | Time (ms) | stddev (ms) |
|---|---|---|
| Monocular features | 160 | 7.6 |
| Stereo features | 730 | 43 |
| 3D features | 8.8 | 0.05 |
| Optimization | 102 | 15 |
| **Total** | **997** | **43** |

Figure 11. Timing results for our system, averaged over the test set. Times show complete processing time for each triplet of frames (base frame plus two auxiliary frames).

## References


[1] P. S. amd K. Alahari amd L. Ladicky and P. H. S. Torr. Combining appearance and structure from motion features for road scene understanding. In *Proc 19th British Machine Vision Conference*, 2009. 2

[2] O. Barinova, V. Konushin, A. Yakubenko, K. Lee, H. Lim, and A. Konushin. Fast automatic single-view 3-d reconstruction of urban scenes. In *ECCV*, pages 100–113, 2008. 2

[3] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *Proc 10th European Conf on Computer Vision*, pages 44–57, Berlin, Heidelberg, 2008. Springer-Verlag. 2, 7

[4] J. Coughlan and A. Yuille. Manhattan world: compass direction from a single image by bayesian inference. In *CVPR*, volume 2, pages 941–947 vol.2, 1999. 2

[5] A. Criminisi. *Accurate visual metrology from single and multiple uncalibrated images*. Springer-Verlag New York, Inc., New York, NY, USA, 2001. 3

[6] D. Felzenszwalb and O. Veksler. Tiered scene labeling with dynamic programming. In *Proc 28th IEEE Conf on Computer Vision and Pattern Recognition*, 2010. 2

[7] A. Flint, C. Mei, I. Reid, and D. Murray. A dynamic programming approach to reconstructing building interiors. In *Proc 12th European Conf on Computer Vision*, 2010. 1, 2, 5, 7

[8] A. Flint, C. Mei, I. Reid, and D. Murray. Growing semantically meaningful models for visual slam. In *Proc 28th IEEE Conf on Computer Vision and Pattern Recognition*, 2010. 2, 3

[9] A. Flint, I. Reid, and D. Murray. Learning textons for real–time scene context. In *Proc 27th IEEE Conf on Computer Vision and Pattern Recognition*, 2009.

[10] Y. Furukawa, B. Curless, S. Seitz, and R. Szeliski. Manhattan-world stereo. *CVPR*, 0:1422–1429, 2009. 2

[11] A. Gupta, A. Efros, and M. Hebert. Blocks world revisited: Image understanding using qualitative geometry and mechanics. In *Proc 12th European Conf on Computer Vision*, volume 6314, pages 482–496. Springer Berlin / Heidelberg, 2010. 2

[12] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, Nara, Japan, November 2007. 6

[13] J. Kosecka̋ and W. Zhang. Video compass. In *ECCV*, volume 2353 of *Lecture Notes in Computer Science*, pages 4: 476–490. Springer, 2002. 2

[14] D. C. Lee, M. Hebert, and T. Kanade. Geometric reasoning for single image structure recovery. In *CVPR*, June 2009. 1, 2, 6, 7

[15] L. Roberts. *Machine perception of 3–d solids*. PhD Thesis, 1965. 2

[16] D. Scharstein and R. Szeleski. A taxonomy and evaluation of dense two–frame stereo correspondence algorithms. In *IJCV*, 2001. 4

[17] C. Vanegas, D. Aliaga, and B. Benes and. Building reconstruction using manhattan-world grammars. In *Proc 28th IEEE Conf on Computer Vision and Pattern Recognition*, 2010. 2