Single Image Depth Estimation From Predicted Semantic Labels

Beyang Liu
Dept. of Computer Science
Stanford University
beyangl@cs.stanford.edu

Stephen Gould
Dept. of Electrical Engineering
Stanford University
sgould@stanford.edu

Daphne Koller
Dept. of Computer Science
Stanford University
koller@cs.stanford.edu

Abstract

We consider the problem of estimating the depth of each pixel in a scene from a single monocular image. Unlike traditional approaches [18, 19], which attempt to map from appearance features to depth directly, we first perform a semantic segmentation of the scene and use the semantic labels to guide the 3D reconstruction. This approach provides several advantages: By knowing the semantic class of a pixel or region, depth and geometry constraints can be easily enforced (e.g., "sky" is far away and "ground" is horizontal). In addition, depth can be more readily predicted by measuring the difference in appearance with respect to a given semantic class. For example, a tree will have more uniform appearance in the distance than it does close up. Finally, the incorporation of semantic features allows us to achieve state-of-the-art results with a significantly simpler model than previous works.

1. Introduction

Recovering the 3D structure of a scene from a single image is a fundamental problem in computer vision that has application in robotics, surveillance and general scene understanding—if we can estimate scene structure then we can better understand the scene by knowing the 3D relationships between the objects within it. However, estimating structure from raw image features is notoriously difficult since local appearance is insufficient to resolve depth ambiguities (e.g., sky and water regions in an image can have similar appearance but dramatically different geometric placement within the scene). Intuitively, semantic understanding of a scene plays an important role in our own perception of scale and 3D structure.

Producing spatially plausible 3D reconstructions of a scene from monocular images annotated with geometric cues (such as horizon, vanishing points, and surface boundaries) is a well understood problem [3]. However, to uniquely determine absolute depths, additional information such as texture, relative depth, and camera parameters (pose and focal length) is needed. Much recent work on automated 3D scene reconstruction [19, 12, 4, 11, 18] has focuses on extracting these geometric cues and additional information from novel images.

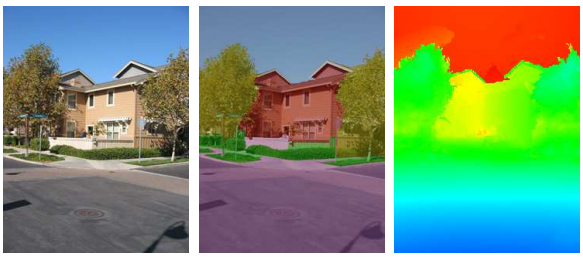


Figure 1. Example output from our model showing how semantic class prediction (center) strongly informs depth perception (right). Semantic classes are shown overlayed on image. Depth indicated by colormap (red is more distant). See Figure 6 for color legend.

These works largely ignore the task of semantic understanding and jump straight to estimating depth or geometry from image features. Using machine learning techniques, these approaches determine a direct mapping from image features to depth. However, this puts an enormous burden on the learning algorithm, which must now implicitly reason about semantic context (e.g., the difference between visually similar patches of sky and water) to accurately learn depth. Thus, much effort in these approaches is in the design of sophisticated image features to unburden the learning algorithm.

In this paper we propose a different approach that reasons about the semantic content of a scene and uses this information as context for depth reconstruction (see Figure 1). The incorporation of semantic class knowledge allows us to do two things: First, we can take advantage of class-related depth and geometry priors. For example, sky is always at the farthest depth possible; grass and road form supporting ground planes for other objects. Second, by conditioning on semantic class, we can better model depth as a function of local pixel appearance. For example, uniformity of texture may be a good indicator for the depth of a tree, but not useful when estimating the depth of a building. Our model is therefore able to use much simpler image features while

still achieving state-of-the-art depth perception results.

Our approach reflects the current trend in computer vision to combine multiple tasks for holistic scene understanding [9, 13, 6, 14]. Ours is a two-phase approach. In the first phase, we use a learned multi-class image labeling MRF [21, 7, 6] to infer the semantic class for each pixel in the image. We currently label pixels as one of: *sky*, *tree*, *road*, *grass*, *water*, *building*, *mountain*, and *foreground object*. The first seven classes cover a large portion of background regions in outdoor scenes while the last class captures the eclectic set of foreground objects such as cars, street signs, people, animals, *etc*.

In the second phase, we use the predicted semantic class labels to inform our depth reconstruction model. Here, we first learn a separate depth estimator for each semantic class. We incorporate these predictions in a Markov random field (MRF) that includes semantic-aware reconstruction priors such as smoothness and orientation of different semantic classes. Motivated by the work of Saxena *et al*. [19], we explore both pixel-based and superpixel-based variants of our model. In the pixel-based variant, we construct a second-order MRF over individual pixels with a preference for smoothness. Our second variant constructs an MRF over small regions (or superpixels) where each region is assumed to be planar. This formulation reduces the number of variables in the model (and hence computational cost). It also allows more global constraints such as the orientation of individual superpixels, and connectivity and co-planarity between neighboring superpixels. These constraints are conditioned on the semantic class of the region and are learned from data.

We test our model on a challenging set of 534 outdoor scenes made publicly available by Saxena *et al*. [19] and compare to other published results on this dataset. Results show that our model outperforms state-of-the-art approaches and yields qualitatively excellent reconstructions.

2. Background and Related Work

There have been many different approaches to the problem of 3D scene reconstruction from monocular images. These can be roughly partitioned into two groups: geometric models and depth perception models.

For indoor environments, Delage *et al*. [4] use an MRF for reconstructing the location of walls, ceilings and floors using geometric cues (such as long straight lines) derived from the scene. More recently, Hedau *et al*. [8] recover the spatial layout of cluttered rooms using similar geometric cues. Both models make strong assumptions about the structure of indoor environments (such as the "box" model of a room [8]) and are not suitable to the less structured outdoor scenes that we consider.

An early approach to outdoor scene reconstruction is the innovative work of Hoiem *et al*. [10] who cast the problem as a multinomial classification problem. In their work, pixels are classified as either ground, sky, or vertical. A simple 3D model can then be constructed by "popping up" vertical regions. The model was later improved [12] to incorporate a broader range of geometric subclasses (porous, solid, left, center, right). Unlike our approach, these models make no attempt to estimate absolute depth. Furthermore, many objects commonly found in everyday scenes (e.g., cars, trees, and people) do not neatly fit into the broad classes they define. A car, for example, consists of many angled surfaces that cannot be modeled as vertical.

It is interesting to note that our semantic classes loosely correspond to the geometric subclasses defined by Hoiem *et al*. [12]: trees are generally porous; buildings are vertical; and road, grass and water are horizontal. Indeed, the scene decomposition model of Gould *et al*. [6] demonstrates the strong correlation between geometry and semantics in outdoor scenes. However, unlike these models which use hard geometric labels, our model allows the soft prior on the orientation of the various semantic classes to be overridden given sufficient contradictory evidence.

A more semantically motivated approach was recently adopted by Russell *et al*. [17] who utilize detailed human-labeled segmentations to infer the geometric class of regions (ground, standing, attached) and region edges (support, occlusion, attachment). In their model, depth inference is done by modeling support and attachment relationships relative to a ground plane. Currently, their model relies on detailed human annotation of regions (and in particular, their polygonal boundaries) within the scene. Our model, on the other hand, infers semantic content from image features.

Our work is most heavily influenced by the work of Saxena and colleagues [19, 18] who take a very different approach to the task of 3D reconstruction. Instead of inferring geometric class labels, they infer absolute depth of the pixels in the image. However, unlike their approach, which completely ignores semantic context, our work makes use of semantic information to guide depth perception. This has a number of advantages: First, we can use simpler features, since depth perception in our model is conditioned on semantic class and thus avoid the need for features that correlate with depth in all classes. Second, we avoid the need for modeling occlusions and folds since these can be easily obtained from the semantic labels (sky is always occluded; ground plane classes "fold" into foreground classes). Last, co-planarity and connectivity constraints can be imposed differently within each semantic class. For example, a building is more likely to be planar than a tree.

The success of holistic scene understanding models has also been a key motivation for this work. These models combine multiple computer vision tasks with the goal of mutual improvement across all tasks [13, 9, 14]. Most closely related to our work is the model of Heitz *et al*. [9],

who combine object detection, multi-class image labeling, and depth perception. However, their model only uses local semantic class information around each pixel as a feature in their linear-regression model. This essentially acts as a simple global depth prior for the semantic class. Our approach, on the other hand, conditions on the semantic label so that different model parameters can be tuned to different semantic classes. Furthermore, we incorporate global features that reason about the structure of semantic classes such as the co-planarity of building regions.

3. Depth Estimation Model

As discussed above, we make use of semantic information to constrain the possible 3D reconstructions of a scene. Our algorithm works in two phases. The first phase predicts the semantic class of each pixel and the location of the horizon. Given this semantic and geometric context, the second phase then estimates depth. We begin the discussion of our approach by describing our method of producing a semantic decomposition of the image and estimating the horizon. Then, we present a brief overview of the geometry of image formation, from which we derive our depth perception models. Finally, we present two variants of the depth reconstruction model—one pixel-based and one superpixel-based—that make use of the semantic information.

3.1. Semantic Labeling and Horizon Prediction

Our model can use any multi-class image labeling method that produces pixel-level semantic annotations [21, 7, 6]. Concretely, we require a model that will assign to each pixel p, in the image $\mathcal{I}$, a class label L_p from some fixed label set $\{\mathcal{L}\}$ (currently *sky*, *road*, *water*, *grass*, *tree*, *building*, *mountain* and *foreground object*).

In our implementation, we use a standard pairwise MRF over pixel labeling $\mathbf{L}$. Briefly, the MRF is defined by

$$\mathbf{E}\left(\mathbf{L} \mid \mathcal{I}\right) = \sum_{p} \psi_p(L_p) + \lambda \sum_{pq} \psi_{pq}(L_p, L_q) \qquad (1)$$

where ψ_p is the unary potential for assigning label L_p to pixel p and ψ_{pq} is a contrast-dependent smoothing prior that penalizes adjacent pixels p and q for taking different labels.

Our unary potentials are learned boosted decision tree classifiers over a standard set of 17 filter response features [21] computed in a small neighborhood around each pixel.[1] Our pairwise potential is also standard: $\psi_{pq} = \exp\left(-c^{-1}\|x_p - x_q\|^2\right)$ if $L_p \neq L_q$ and 0 otherwise, where x_p and x_q are the color vectors for pixels p and q, respectively, and c is the mean square-difference between color vectors over all adjacent pixels in the image. The parameter λ determines the strength of this smoothness prior and is chosen by cross-validation on the training set. We use α-expansion [1] to find an approximate solution to the energy function giving us the semantic labels.

Many semantic decomposition models also predict the location of the horizon (e.g., [6, 11]). When this is not the case, a simple adaption of the ideas from Gould *et al*. [6] can be used to produce a good estimate of the horizon from the semantic labeling itself. Here, we assume that the horizon v^{hz} can be modeled as a row within the image and define a prediction model as

$$\mathbf{E}\left(v^{\text{hz}} \mid \mathcal{I}, \mathbf{L}\right) = \log \mathcal{N}(v^{\text{hz}}; \mu, \sigma) + \sum_{p} \psi_p(v^{\text{hz}}; L_p) \qquad (2)$$

where $\mathcal{N}(v^{\text{hz}}; \mu, \sigma)$ is a normal distribution reflecting our prior estimate of the horizon location and $\psi_p(v^{\text{hz}}; L_p)$ penalizes inconsistent relative location between the horizon and pixels with a given semantic class (e.g., ground plane pixels should be below the horizon and sky pixels above). The location of the horizon is determined as the minimizing assignment to v^{hz}.

3.2. Image Formation and Scene Geometry

Consider an ideal camera model (i.e., with no lens distortion). Then, a pixel p with coordinates (u_p, v_p) (in the camera plane) is the image of a point in 3D space that lies on the ray extending from the camera origin through (u_p, v_p) in the camera plane. The ray r_p in the world coordinate system is given by $r_p \propto \mathbf{R}^{-1}\mathbf{K}^{-1}[u_p\ v_p\ 1]^T$, where $\mathbf{R} \in SO(3)$ defines the transformation (rotation) from camera coordinate system to world coordinate system, and $\mathbf{K} \in \mathbb{R}^{3\times 3}$ is the camera matrix [16].[2] In the sequel, we will assume that r_p has been normalized (i.e., $\|r_p\|_2 = 1$). For an ideal camera, the camera matrix has the form

$$\mathbf{K} = \begin{bmatrix} f_u & 0 & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (3)$$

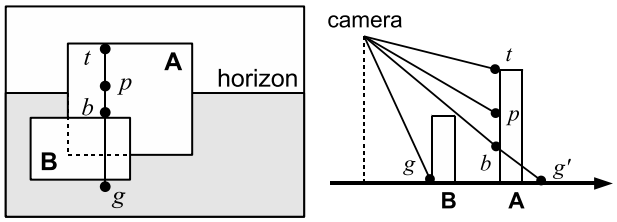
where f_u and f_v are the (u- and v-scaled) focal lengths of the camera, and the principal point (u_0, v_0) is the center pixel in the image. As in Saxena *et al*. [19] we assume a reasonable value for the focal length (in our experiments we set $f_u = f_v = 348$ for a 240×320 image). We further assume that the image was taken with the camera's horizontal (x) axis parallel to the ground, and we estimate the yz-rotation of the camera plane from the predicted location of the horizon (assumed to be at depth ∞)

[1] Our filter-bank consists of Gaussians (on all three color channels) at scales 1, 2 and 4, x- and y-derivatives of Gaussians (on the luminance color channel) at scales 2 and 4, and Laplacian of Gaussians (on the luminance color channel) at scales 1, 2, 4 and 8.

加起来刚好17个滤波核

[2] In our model, we assume that there is no translation between the world coordinate system and the camera coordinate system, and that the images were taken from approximately the same height above the ground.

只有旋转无平移，保证相机高度一样



(a) Image view (b) Side view

Figure 2. Illustration of semantically derived geometric constraints. See text for details.

as $\theta = \tan^{-1}(\frac{1}{f_v}(v^{\text{hz}} - v_0))$. This yields the rotation matrix

只在yz平面旋转

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & cos\theta \end{bmatrix} \quad (4)$$

With the camera rays r_p in the world coordinate system we can now easily encode constraints about the geometry of a scene. Consider, for example, the simple scene in Figure 2, and assume that we would like to estimate the depth of some pixel p on a vertical object **A** attached to the ground. We define three key points that are strong indicators of the depth of p. First, let g be the topmost ground pixel in the same column below p. The depth of g is a lower bound on the depth of p. Second, let b be the bottommost visible pixel b on the object **A**. By extending the camera ray through b to the ground, we can calculate an upper bound on the depth of p. Third, the topmost point t on the object may also be useful since a non-sky pixel high in the image (e.g., an overhanging tree) tends to be close to the camera.

Simple geometric reasoning allows us to encode the first two constraints as

$$d_g \left(\frac{r_g^T e_3}{r_p^T e_3} \right) \leq d_p \leq d_g \left(\frac{r_g^T e_2}{r_b^T e_2} \right) \left(\frac{r_b^T e_3}{r_p^T e_3} \right) \quad (5)$$

e是标准正交基

where d_p and d_g are the distances to the points p and g, respectively, and e_i is the i-th canonical vector (i.e., vector with i-th element one and the rest zero). The third constraint can be similarly encoded as $d_t r_t^T e_3 \approx d_p r_p^T e_3$. dt=dp

In the following sections we show how these constraints are incorporated as features and potential terms in our depth perception MRF models.

3.3. Features and Pointwise Depth Estimation

Our goal is to predict the depth of every pixel in the image. We begin by constructing a descriptor $f_p \in \mathbb{R}^n$ for each pixel, which includes local appearance features and global geometry features derived from our semantic understanding of the scene (discussed below). Standard depth perception models utilize the fact that the local appearance of a pixel changes with depth and attempt to learn a function that describes this relationship. However, this relationship depends on the semantic class (e.g., a distant tree will appear less textured than a nearby one, but not necessarily so with more uniform classes such as building or road). Furthermore, the distance of some classes is tightly constrained by the class itself (e.g., sky). Accordingly, we learn a different local depth predictor for each semantic class.

Motivated by the desire to more accurately model the depth of nearby objects and the fact that relative depth is more appropriate for scene understanding, we learn a model to predict log-depth rather than depth itself. We thus estimate pointwise log-depth as a linear function of the pixel features (given the pixel's semantic class), 预测log深度且是特征的组合

$$\log \hat{d}_p = \theta_{L_p}^T f_p \quad (6)$$

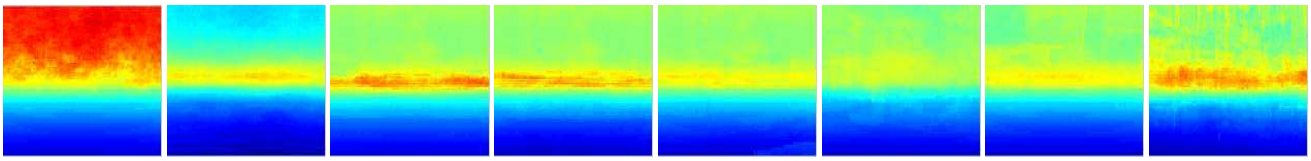
where $\hat{d}_p$ is the pointwise estimated depth for pixel p, L_p is its semantic class label predicted from Equation (1), $f_p \in \mathbb{R}^n$ is a local feature vector, and $\{\theta_l\}_{l \in \mathcal{L}}$ are the learned parameters of the model.

Our basic pixel appearance features are the 17 raw filter responses used in the semantic model and also the log of these features. We also include the (u, v) coordinates of the pixel and an a priori estimated log-depth for pixel coordinates (u_p, v_p) and semantic label L_p. These are all adjusted to a consistent world coordinate system. The prior log-depth is learned, for each semantic class, by averaging the log-depth at each (u, v)-pixel location over the set of training images. Since not all semantic class labels appear in all pixel locations, we smooth the priors with a global log-depth prior (the average of the log-depths over all the classes at the particular location).

Figure 3 illustrates these features for the eight semantic classes in our model. It is interesting to note the differences between the semantic classes. A tree pixel towards the top of the image, for example, is likely to be closer than a tree pixel near the horizon (center of the image). This supports the observation made in Section 3.2 that the topmost pixel in a region may be a more useful (a priori) indicator for region depth than a pixel within the region.

We encode additional geometric constraints as features by examining the three key pixels discussed in Section 3.2. For each of these pixels (bottommost and topmost pixel with class L_p and topmost ground pixel), we use the pixel's prior log-depth to calculate a depth estimate for p (assuming that most objects are roughly vertical) and include this estimate as a feature. We also include the (horizon-adjusted) vertical coordinate of these pixels as features. Note that our verticality assumption is not a hard constraint, but rather a soft one that can be overridden by the strength of other features. By including these as features, we allow our model to learn the strength of these constraints.

Finally, we add the square of each feature, allowing us to learn quadratic depth correlations.



(a) sky (b) tree (c) road (d) grass (e) water (f) building (g) mountain (h) fg. obj.

Figure 3. Smoothed per-pixel log-depth prior for each semantic class with horizon rotated to center of image. Colors indicate distance (red is further away and blue is closer). The classes "water" and "mountain" had very few samples and so are close to the global log-depth prior (not shown). See text for details.

We learn the parameters of the model $\{\theta_l\}_{l\in\mathcal{L}}$ by linear regression on the log-depths from a set of training images. For numerical stability, we first normalize each feature to zero mean and unit variance.

3.4. MRF Models for Depth Reconstruction

The pointwise depth estimation provided by Equation (6) is somewhat noisy and can be improved by including priors that constrain the structure of the scene. We develop two different MRF models—one pixel-based and one superpixel-based—for the inclusion of such priors. The priors constrain the depth relationships between two or more pixels (superpixels) and will usually be weighted by the contrast between the pixels (superpixels), e.g., we would like to allow for depth discontinuities at strong boundaries.

3.4.1 Pixel-based Markov Random Field

Our pixel-based MRF includes a prior for smoothness. Here, we add a potential over three consecutive pixels (in the same row or column) that prefers co-linearity. We also encode semantically-derived depth constraints which penalize vertical surfaces from deviating from geometrically plausible depths (as described in Section 3.2). Formally, we define the energy function over pixel depths $\mathbf{D}$ as

$$\mathbf{E}\left(\mathbf{D} \mid \mathcal{I}, \mathbf{L}\right) = \underbrace{\sum_{p} \psi_p(d_p)}_{\text{data term}} + \underbrace{\sum_{pqr} \psi_{pqr}(d_p, d_q, d_r)}_{\text{smoothness}} + \underbrace{\sum_{p} \psi_{pg}(d_p, d_g) + \sum_{p} \psi_{pb}(d_p, d_b) + \sum_{p} \psi_{pt}(d_p, d_t)}_{\text{geometry (see §3.2)}} \tag{7}$$

where the data term, ψ_p, attempts to match the depth for each pixel d_p to the pointwise estimate $\hat{d}_p$, and ψ_{pqr} represents the co-linearity prior. The terms ψ_{pg}, ψ_{pb} and ψ_{pt} represent the geometry constraints described in Section 3.2 above. Recall that the pixel indices g, b and t are determined from p and the semantic labels.

The data term in our model is given by

$$\psi_p(d_p) = h(d_p - \hat{d}_p; \beta) \tag{8}$$

where $h(x;\beta)$ is the Huber penalty, which takes the value x^2 for $-\beta \le x \le \beta$ and $\beta(2|x| - \beta)$ otherwise. We choose the Huber penalty because it is more robust to outliers than the more commonly used ℓ_2-penalty and, unlike the robust ℓ_1-penalty, is continuously differentiable (which simplifies inference). In our model, we set $\beta = 10^{-3}$.

Our smoothness prior encodes a preference for co-linearity of adjacent pixels within uniform regions. Assuming pixels p, q, and r are three consecutive pixels (in any row or column), we have

$$\psi_{pqr} = \lambda^{\text{smooth}} \cdot \sqrt{\gamma_{pq}\gamma_{qr}} \cdot h(2d_q - d_p - d_r; \beta) \tag{9}$$

where the smoothness penalty is weighted by a contrast-dependent term and the prior strength λ^{smooth}. Here, $\gamma_{pq} = \exp\left(-c^{-1}\|x_p - x_q\|^2\right)$ measures the contrast between two adjacent pixels, where x_p and x_q are the CIELab color vectors for pixels p and q, respectively, and c is the mean square-difference over all adjacent pixels in the image. Note that this is the same contrast term used by the semantic model. We choose the prior strength by cross-validation on a set of training images.

The soft geometry constraints ψ_{pg}, ψ_{pt} and ψ_{pb} model our prior belief that certain semantic classes are vertically oriented (e.g., buildings, trees and foreground objects). Here, we impose the soft constraint that a pixel within such a region should be the same depth as other pixels in the region (i.e., via the constraint on the topmost and bottommost pixels in the region), and be between the nearest and farthest ground plane points g and g' defined in Section 3.2. The constraints are encoded using the Huber penalty, (e.g., $h(d_p - d_g; \beta)$ for the nearest ground pixel constraint). Each term is weighted by a semantic-specific prior strength $\{\lambda^{\text{g}}_l, \lambda^{\text{t}}_l, \lambda^{\text{b}}_l\}_{l\in\mathcal{L}}$.

3.4.2 Superpixel-based Markov Random Field

While the pixel-based model described above allows us to incorporate semantic information by learning different parameters for mapping pixel appearance to depth and a preference for smoothness, it does not allow us to easily incorporate higher-order geometric constraints such as the planarity of an entire region. We now develop a superpixel-based model that allows the incorporation of such priors.

颜色差异

三点共线

We segment the image into a set of non-overlapping superpixels using a bottom-up over-segmentation algorithm. In our experiments we use mean-shift [2], but could equally have used other approaches [5, 20]. Each superpixel S_i is assumed to be planar, a constraint that we strictly enforce. The plane parameters $\{\alpha_i\}$ are unnormalized so that any point $x \in \mathbb{R}^3$ on the plane satisfies $\alpha_i^T x = 1$. In particular, the depth of pixel p corresponds to the intersection of the ray r_p and the plane, and is given by $(\alpha_i^T r_p)^{-1}$.

Our superpixel-based depth reconstruction model aims to infer the plane parameters of each superpixel given the semantic class. We define an energy function that includes terms that penalize the distance between the superpixel planes and the pointwise depth estimates $\hat{d}_p$ (Equation (6)) and terms that enforce soft connectivity, co-planarity, and orientation constraints over the planes. All of these are conditioned on the semantic class of the superpixel (taken as the majority vote over the superpixel's constituent pixels). Formally, we have

$$\mathbf{E}\left(\boldsymbol{\alpha} \mid \mathcal{I}, \mathbf{L}, \mathbf{S}\right) = \underbrace{\sum_p \psi_p(\alpha_{i\sim p})}_{\text{data term}} + \underbrace{\sum_i \psi_i(\alpha_i)}_{\text{orientation prior}} + \underbrace{\sum_{ij} \psi_{ij}(\alpha_i, \alpha_j)}_{\text{connectivity and co-planarity prior}} \quad (10)$$

Here $\alpha_{i\sim p}$ indicates the α_i associated with the superpixel containing pixel p, i.e., $\alpha_i : p \in S_i$.

Region Data Term. The data term penalizes the plane parameters from deviating away from the pointwise depth estimates. It takes the form

$$\psi_p(\alpha_i) = \frac{1}{\hat{d}_p} h\left(\hat{d}_p \cdot \alpha_i^T r_p - 1; \beta\right) \quad (11)$$

where $h(x;\beta)$ is the Huber penalty as defined in Section 3.4.1 above. We weight each pixel term by the inverse pointwise depth estimate to prefer nearby regions.

逆深度权重更关注近处

Orientation Prior. The orientation prior enables us to encode a preference for orientation of different semantic surfaces, e.g., ground plane surfaces ("road", "grass", *etc.*) should be horizontal while buildings should be vertical. We encode this preference as

$$\psi_i(\alpha_i) = N_i \cdot \lambda_l \cdot \|P_l\left(\alpha_i - \bar{\alpha}_l\right)\|^2 \quad (12)$$

where P_l projects onto the planar directions that we would like to constrain and $\bar{\alpha}_l$ is the prior estimate for the orientation of a surface with semantic class label $L_i = l$. We weight each superpixel by its number of pixels (N_i) and a semantic-class-specific prior strength (λ_l). The latter captures our confidence in a semantic class's orientation prior (e.g., we are very confident that ground is horizontal, but we are less certain a priori about the orientation of tree regions).

Connectivity and Co-planarity Prior. The connectivity and co-planarity term captures the relationship between two adjacent superpixels. For example, we would not expect adjacent "sky" and "building" superpixels to be connected, whereas we would expect "road" and "building" to be connected. Defining B_{ij} to be the set of pixels along the boundary between superpixels i and j, we have

$$\psi_{ij}(\alpha_i, \alpha_j) = \frac{N_i + N_j}{2|B_{ij}|} \lambda_{lk}^{\text{conn}} \cdot \sum_{p \in B_{ij}} \|\alpha_i^T r_p - \alpha_j^T r_p\|^2 + \frac{N_i + N_j}{2} \lambda_{lk}^{\text{co-plnr}} \cdot \|\alpha_i - \alpha_j\|^2 \quad (13)$$

连通性代表拐角，共面性

where we weight each term by the average number of pixels in the associated superpixels and pairwise semantic-class-specific prior strength ($\lambda_{lk}^{\text{conn}}$ and $\lambda_{lk}^{\text{co-plnr}}$).

3.5. Inference and Learning

Both of our MRF formulations (Equation (7) and Equation (10)) define convex objectives which we solve using the L-BFGS algorithm [15] to obtain a depth prediction for every pixel in the image—for the superpixel-based model we compute pixel depths as $d_p = \frac{1}{\alpha_i^T r_p}$ where α_i are the inferred plane parameters for the superpixel containing pixel p. In our experiments, inference takes about 2 minutes per image for the pixel-based MRF and under 30 seconds for the superpixel-based model (on a 240×320 image).

The various prior strengths (λ^{smooth}, *etc.*) are learned by cross-validation on the training data set. To make this process computationally tractable, we add terms in an incremental fashion, freezing each weight before adding the next term. This coordinate-wise optimization seemed to yield good parameters.

4. Experimental Results

We ran experiments on the publicly available dataset from Saxena *et al*. [19]. The dataset consists of 534 images with corresponding depth maps and is divided into 400 training and 134 testing images. We hand-annotated the training images with semantic class labels. The 400 training images were used for learning the parameters of the semantic and depth models. All images were resized to 240×320 before running our algorithm.

We report results on the 134 test images. Since the maximum range of the sensor used to collect ground truth measurements was 81m, we truncate our predictions to the range $[0, 81]$. Table 4 shows our results compared against previous published results. We compare both the average log-error and average relative error, defined as $|\log_{10} g_p - \log_{10} d_p|$ and $\frac{|g_p - d_p|}{g_p}$, respectively, where g_p is the ground truth depth for pixel p. We also compare our results to our own baseline implementation which does not use any semantic information.

METHOD	$\log_{10}$	REL.
SCN [18]†	0.198	0.530
HEH [11]†	0.320	1.423
Pointwise MRF [19]†	0.149	0.458
PP-MRF [19]†	0.187	**0.370**
Pixel MRF Baseline	0.206	0.464
Pixel MRF Model (§3.4.1)	0.149	0.375
Superpixel MRF Baseline	0.209	0.471
Superpixel MRF Model (§3.4.2)	**0.148**	0.379

† Results reported in Saxena *et al.* [19].

Figure 4. Quantitative results comparing variants of our "semantic-aware" approach with strong baselines and other state-of-the-art methods. Baseline models do not use semantic class information.

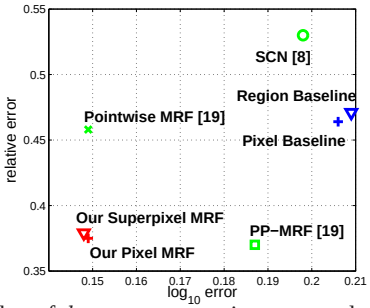


Figure 5. Plot of log_{10} error metric versus relative error metric comparing algorithms from Table 4 (HEH [11] not shown). Bottom-left indicates better performance.

Figure 7. Example 3d reconstructions from our model.

Both our pixel- and superpixel-based models achieve state-of-the-art performance for the $\log_{10}$ metric and comparable performance to state-of-the-art for the relative error metric. Importantly, they achieve good results on both metrics unlike the previous results which perform well at either one or the other. This can be clearly seen in Figure 5 where we have plotted the performance metrics on the same graph.

Having semantic labels allows us to break down our results by (predicted) class. Our best performing results are the ground plane classes (especially road), which are easily identified by our semantic model and tightly constrained geometrically. We achieve poor performance on the foreground class which we attribute to the lack of foreground objects in the training set (less than 1% of the pixels).

Unexpectedly, we also perform poorly on sky pixels which are easy to predict and should always be positioned at the maximum depth. This is due, in part, to errors in the groundtruth measurements (caused by sensor misalignment) and the occasional misclassification of the reflective surfaces of buildings as sky by our semantic model. Note that the nature of the relative error metric is to magnify these mistakes since the ground truth measurement in these cases is always closer than the maximum depth.

Given our heavy reliance on inferred semantic evidence at the depth estimation stage of both algorithms, it is also important to consider the robustness of our approach to errors in the semantic labeling. Quantitatively, 8% of pixels were incorrectly classified. The depth estimation accuracy over these pixels was comparable between our model and the baseline model (0.209/0.507 versus 0.226/0.499 $\log_{10}$/relative error) showing that the depth-estimation stage of our approach is able to partially overcome mistakes made in the semantic classification stage.

Finally, we show some qualitative results in Figure 6 and example 3D reconstructions in Figure 7. The results show that we correctly model co-planarity of the ground plane and building surfaces. Notice our accurate prediction of the sky (which is sometimes penalized by misalignment in the groundtruth, e.g., bottom-right example). Our algorithm also makes mistakes, such as positioning the building too close in the middle-left example and missing the ledge in the foreground (a mistake that many human would also make). We also miss some low contrast objects such as the post in the top-right figure.

5. Discussion

This work addresses the problem of depth perception from a single monocular image through the incorporation of predicted semantic information. The inclusion of semantic information allowed us to model appearance and geometry constraints that were not possible in previous works (e.g., [19]). With semantic reasoning, we achieved state-of-the-art results using a geometrically plausible model and simpler image features. Importantly, our method can use any multi-class semantic labeling model.

There are a number of interesting extensions suggested by our approach. First, it would be valuable to divide our foreground class into subcategories to allow the inclusion of additional modeling constraints (such as the average height of a person or the average width of a car). Furthermore, such constraints also inform upon camera parameters such as camera height, rotation, and focal length, and incorporating these constraints could enable the model to infer those parameters automatically and with greater accuracy.

Second, our geometric modeling currently makes strong assumptions about the location of the supporting pixel for vertical objects (i.e., the topmost ground pixel below the object). However, this assumption breaks for overhanging objects (such as outstretched arms, building arches, and tree limbs). A model which can more accurately determine a pixel's "support point" will allow the geometry priors to be strengthened and likely result in better performance.

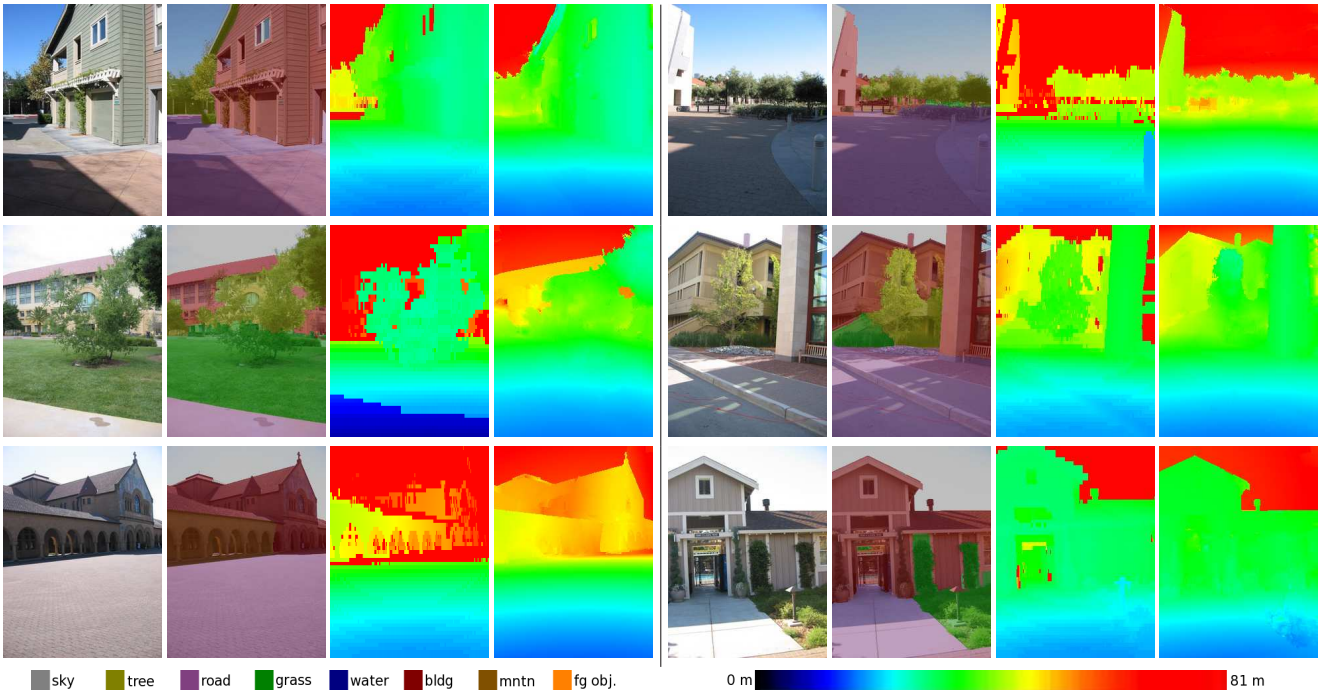


Figure 6. Some qualitative depth reconstructions from our model showing (from left to right) the image, semantic overlay, ground truth depth measurements, and our predicted depths. Legend shows semantic color labels and depth scale.

Finally, our current approach relies on accurate ground truth data for learning the parameters of our linear regression model and prior strengths. This is hampered by the limitation imposed by real-world depth sensors and the quality of existing datasets. We aim to extend our model to learn from richer data sources including synthetic data (e.g., from ray-traced scenes) and weakly labeled images (e.g., with the height of only a few objects labeled).

Acknowledgments. This work was supported by NSF (ISS 0917151), MURI (N000140710747), and the Boeing company.

References

[1] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. In *CVPR*, 2003.

[2] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *PAMI*, 2002.

[3] A. Criminisi, I. Reid, and A. Zisserman. Single view metrology. *IJCV*, 2000.

[4] E. Delage, H. Lee, and A. Y. Ng. A dynamic Bayesian network model for autonomous 3d reconstruction from a single indoor image. In *CVPR*, 2006.

[5] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 2004.

[6] S. Gould, R. Fulton, and D. Koller. Decompsing a scene into geometric and semantically consistent regions. *ICCV*, 2009.

[7] X. He, R. Zemel, and M. Carreira-Perpinan. Multiscale CRFs for image labeling. In *CVPR*, 2004.

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

[9] G. Heitz, S. Gould, A. Saxena, and D. Koller. Cascaded classification models: Combining models for holistic scene understanding. In *NIPS*, 2008.

[10] D. Hoiem, A. A. Efros, and M. Hebert. Automatic photo pop-up. In *SIGGRAPH*, 2005.

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

[12] D. Hoiem, A. A. Efros, and M. Hebert. Recovering surface layout from an image. *IJCV*, 2007.

[13] D. Hoiem, A. A. Efros, and M. Hebert. Closing the loop on scene interpretation. *CVPR*, 2008.

[14] L.-J. Li, R. Socher, and L. Fei-Fei. Towards total scene understanding: Classification, annotation and segmentation in an automatic framework. In *CVPR*, 2009.

[15] D. Liu and J. Nocedal. On the limited memory method for large scale optimization. In *Math. Prog. B*, 1989.

[16] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry. *An Invitation to 3-D Vision*. Springer, 2005.

[17] B. C. Russell and A. Torralba. Building a database of 3D scenes from user annotations. In *CVPR*, 2009.

[18] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In *NIPS*, 2005.

[19] A. Saxena, M. Sun, and A. Y. Ng. Make3D: Learning 3-D scene structure from a single still image. In *PAMI*, 2008.

[20] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE PAMI*, 2000.

[21] J. Shotton, J. Winn, C. Rother, and A. Criminisi. TextonBoost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV*, 2006.