Learning 3-D Scene Structure from a Single Still Image

Ashutosh Saxena, Min Sun and Andrew Y. Ng
Computer Science Department, Stanford University, Stanford, CA 94305
{asaxena,aliensun,ang}@cs.stanford.edu

Abstract

We consider the problem of estimating detailed 3-d structure from a single still image of an unstructured environment. Our goal is to create 3-d models which are both quantitatively accurate as well as visually pleasing.

For each small homogeneous patch in the image, we use a Markov Random Field (MRF) to infer a set of "plane parameters" that capture both the 3-d location and 3-d orientation of the patch. The MRF, trained via supervised learning, models both image depth cues as well as the relationships between different parts of the image. Inference in our model is tractable, and requires only solving a convex optimization problem. Other than assuming that the environment is made up of a number of small planes, our model makes no explicit assumptions about the structure of the scene; this enables the algorithm to capture much more detailed 3-d structure than does prior art (such as Saxena et al., 2005, Delage et al., 2005, and Hoiem et el., 2005), and also give a much richer experience in the 3-d flythroughs created using image-based rendering, even for scenes with significant non-vertical structure.

Using this approach, we have created qualitatively correct 3-d models for 64.9% of 588 images downloaded from the internet, as compared to Hoiem et al.'s performance of 33.1%. Further, our models are quantitatively more accurate than either Saxena et al. or Hoiem et al.

1. Introduction

When viewing an image such as that in Fig. 1a, a human has no difficulty understanding its 3-d structure (Fig. 1b). However, inferring the 3-d structure remains extremely challenging for current computer vision systems—there is an intrinsic ambiguity between local image features and the 3-d location of the point, due to perspective projection.

Most work on 3-d reconstruction has focused on using methods such as stereovision [16] or structure from motion [6], which require two (or more) images. Some methods can estimate 3-d models from a single image, but they make strong assumptions about the scene and work in specific settings only. For example, shape from shading [18], relies on purely photometric cues and is difficult to apply to surfaces that do not have fairly uniform color and texture. Criminisi, Reid and Zisserman [1] used known vanishing points to determine an affine structure of the image.

Figure 1. (a) A single image. (b) A screenshot of the 3-d model generated by our algorithm.

In recent work, Saxena, Chung and Ng (SCN) [13, 14] presented an algorithm for predicting depth from monocular image features. However, their depthmaps, although useful for tasks such as a robot driving [12] or improving performance of stereovision [15], were not accurate enough to produce visually-pleasing 3-d fly-throughs. Delage, Lee and Ng (DLN) [4, 3] and Hoiem, Efros and Hebert (HEH) [9, 7] assumed that the environment is made of a flat ground with vertical walls. DLN considered indoor images, while HEH considered outdoor scenes. They classified the image into ground and vertical (also sky in case of HEH) to produce a simple "pop-up" type fly-through from an image. HEH focussed on creating "visually-pleasing" fly-throughs, but do not produce quantitatively accurate results. More recently, Hoiem et al. (2006) [8] also used geometric context to improve object recognition performance.

In this paper, we focus on inferring the detailed 3-d structure that is both quantitatively accurate as well as visually pleasing. Other than "local planarity," we make no explicit assumptions about the structure of the scene; this enables our approach to generalize well, even to scenes with significant non-vertical structure. We infer both the 3-d location and the orientation of the small planar regions in the image using a Markov Random Field (MRF). We will learn the relation between the image features and the location/orientation of the planes, and also the relationships between various parts of the image using supervised learning. For comparison, we also present a second MRF, which models only the location of points in the image. Although quantitatively accurate, this method is unable to give visually pleasing 3-d models. MAP inference in our models is efficiently performed by solving a linear program.

Using this approach, we have inferred qualitatively cor-

rect and visually pleasing 3-d models automatically for 64.9% of the 588 images downloaded from the internet, as compared to HEH performance of 33.1%. "Qualitatively correct" is according to a metric that we will define later. We further show that our algorithm predicts quantitatively more accurate depths than both HEH and SCN.

2. Visual Cues for Scene Understanding

Images are the projection of the 3-d world to two dimensions—hence the problem of inferring 3-d structure from an image is degenerate. An image might represent an infinite number of 3-d models. However, not all the possible 3-d structures that an image might represent are valid; and only a few are likely. The environment that we live in is reasonably structured, and hence allows humans to infer 3-d structure based on prior experience.

Humans use various monocular cues to infer the 3-d structure of the scene. Some of the cues are local properties of the image, such as texture variations and gradients, color, haze, defocus, etc. [13, 17]. Local image cues alone are usually insufficient to infer the 3-d structure. The ability of humans to "integrate information" over space, i.e., understanding the relation between different parts of the image, is crucial to understanding the 3-d structure. [17, chap. 11]

Both the relation of monocular cues to the 3-d structure, as well as relation between various parts of the image is learned from prior experience. Humans remember that a structure of a particular shape is a building, sky is blue, grass is green, trees grow above the ground and have leaves on top of them, and so on.

3. Image Representation

We first find small homogeneous regions in the image, called "Superpixels," and use them as our basic unit of representation. (Fig. 6b) Such regions can be reliably found using over-segmentation [5], and represent a coherent region in the image with all the pixels having similar properties. In most images, a superpixel is a small part of a structure, such as part of a wall, and therefore represents a plane.

In our experiments, we use algorithm by [5] to obtain the superpixels. Typically, we over-segment an image into about 2000 superpixels, representing regions which have similar color and texture. Our goal is to infer the location and orientation of each of these superpixels.

4. Probabilistic Model

It is difficult to infer 3-d information of a region from local cues alone, (see Section 2) and one needs to infer the 3-d information of a region in relation to the 3-d information of other region.

In our MRF model, we try to capture the following properties of the images:

- **Image Features and depth**: The image features of a superpixel bear some relation to the depth (and orientation) of the superpixel.
- **Connected structure**: Except in case of occlusion, neighboring superpixels are more likely to be connected to each other.
- **Co-planar structure**: Neighboring superpixels are more likely to belong to the same plane, if they have similar features and if there are no edges between them.
- **Co-linearity**: Long straight lines in the image represent straight lines in the 3-d model. For example, edges of buildings, sidewalk, windows.

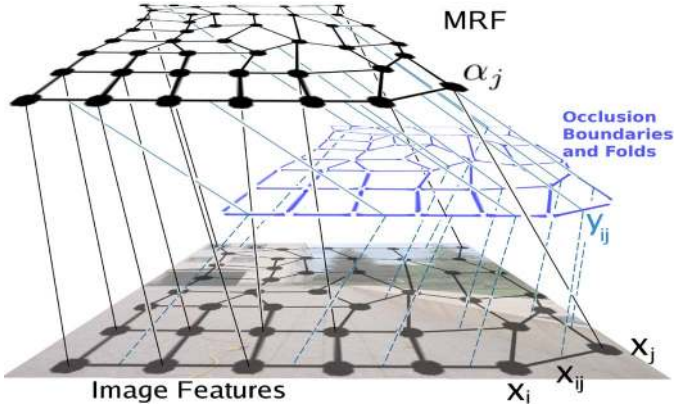

Figure 2. An illustration of the Markov Random Field (MRF) for inferring 3-d structure. (Only a subset of edges and scales shown.)

Note that no single one of these four properties is enough, by itself, to predict the 3-d structure. For example, in some cases, local image features are not strong indicators of the depth (and orientation). Thus, our approach will combine these properties in an MRF, in a way that depends on our "confidence" in each of these properties. Here, the "confidence" is itself estimated from local image cues, and will vary from region to region in the image.

Concretely, we begin by determining the places where there is no *connected* or *co-planar* structure, by inferring variables y_{ij} that indicate the presence or absence of occlusion boundaries and folds in the image (Section 4.1). We then infer the 3-d structure using our "Plane Parameter MRF," which uses the variables y_{ij} to selectively enforce coplanar and connected structure property (Section 4.2). This MRF models the 3-d location and orientation of the superpixels as a function of image features.

For comparison, we also present an MRF that only models the 3-d location of the points in the image ("Point-wise MRF," Section 4.3) We found that our Plane Parameter MRF outperforms our Point-wise MRF (both in quantitative and visually pleasing aspects); therefore we will discuss Point-wise MRF only briefly.

4.1. Occlusion Boundaries and Folds

We will infer the location of occlusion boundaries and folds (places where two planes are connected but not co-planar). We use the variables $y_{ij} \in \{0, 1\}$ to indicate

whether an "edgel" (the edge between two neighboring superpixels) is an occlusion boundary/fold or not. The inference of these boundaries is typically not completely accurate; therefore we will infer *soft* values for y_{ij}. More formally, for an edgel between two superpixels i and j, $y_{ij} = 0$ indicates an occlusion boundary/fold, and $y_{ij} = 1$ indicates none (i.e., a planar surface). We model y_{ij} using a logistic response as $P(y_{ij} = 1|x_{ij}; \psi) = 1/(1 + \exp(-\psi^T x_{ij}))$. where, x_{ij} are features of the superpixels i and j (Section 5.2), and ψ are the parameters of the model. During inference (Section 4.2), we will use a mean field-like approximation, where we replace y_{ij} with its mean value under the logistic model.

4.2. Plane Parameter MRF

In this MRF, each node represents a superpixel in the image. We assume that the superpixel lies on a plane, and we will infer the location and orientation of that plane.

Representation: We parameterize both the location and orientation of the infinite plane on which the superpixel lies by using plane parameters $\alpha \in \mathbb{R}^3$. (Fig. 3) (Any point $q \in \mathbb{R}^3$ lying on the plane with parameters α satisfies $\alpha^T q = 1$.) The value $1/|\alpha|$ is the distance from the camera center to the closest point on the plane, and the normal vector $\hat{\alpha} = \frac{\alpha}{|\alpha|}$ gives the orientation of the plane. If R_i is the unit vector from the camera center to a point i lying on a plane with parameters α, then $d_i = 1/R_i^T \alpha$ is the distance of point i from the camera center.

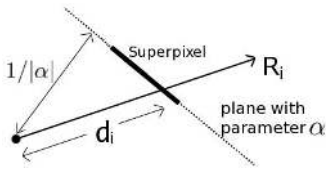

Figure 3. A 2-d illustration to explain the plane parameter α and rays R from the camera.

Fractional depth error: For 3-d reconstruction, the fractional (or relative) error in depths is most meaningful; and is used in structure for motion, stereo reconstruction, etc. [10, 16] For ground-truth depth d, and estimated depth $\hat{d}$, fractional error is defined as $(\hat{d} - d)/d = \hat{d}/d - 1$. Therefore, we would be penalizing fractional errors in our MRF.

Model: To capture the relation between the plane parameters and the image features, and other properties such as co-planarity, connectedness and co-linearity, we formulate our MRF as

$$P(\alpha|X, Y, R; \theta) = \frac{1}{Z} \prod_i f_\theta(\alpha_i, X_i, y_i, R_i) \prod_{i,j} g(\alpha_i, \alpha_j, y_{ij}, R_i, R_j) \quad (1)$$

where, α_i is the plane parameter of the superpixel i. For a total of S_i points in the superpixel i, we use x_{i,s_i} to denote the features for point s_i in the superpixel i. $X_i = \{x_{i,s_i} \in \mathbb{R}^{524} : s_i = 1, ..., S_i\}$ are the features for the superpixel i. (Section 5.1) Similarly, $R_i = \{R_{i,s_i} : s_i = 1, ..., S_i\}$ is the set of rays for superpixel i.

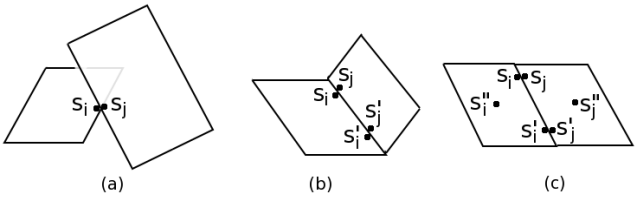

Figure 4. An illustration to explain effect of the choice of s_i and s_j on enforcing the following properties: (a) Partially connected, (b) Fully connected, and (c) Co-planar.

The first term $f(.)$ models the plane parameters as a function of the image features x_{i,s_i}. We have $R_{i,s_i}^T \alpha_i = 1/d_{i,s_i}$ (where R_{i,s_i} is the ray that connects the camera to the 3-d location of point s_i), and if the estimated depth $\hat{d}_{i,s_i} = x_{i,s_i}^T \theta_r$, then the fractional error would be $(R_{i,s_i}^T \alpha_i (x_{i,s_i}^T \theta_r) - 1)$. Therefore, to minimize the aggregate fractional error over all the points in the superpixel, we model the relation between the plane parameters and the image features as

$$f_\theta(\alpha_i, X_i, y_i, R_i) = \exp\left(-\sum_{s_i=1}^{S_i} \nu_{i,s_i} \left|R_{i,s_i}^T \alpha_i (x_{i,s_i}^T \theta_r) - 1\right|\right)$$

The parameters of this model are $\theta_r \in \mathbb{R}^{524}$. We use different parameters (θ_r) for each row r in the image, because the images we consider are taken from a horizontally mounted camera, and thus different rows of the image have different statistical properties. E.g., a blue superpixel might be more likely to be sky if it is in the upper part of image, or water if it is in the lower part of the image. Here, $y_i = \{\nu_{i,s_i} : s_i = 1, ..., S_i\}$ and the variable ν_{i,s_i} indicates the confidence of the features in predicting the depth $\hat{d}_{i,s_i}$ at point s_i.[1] If the local image features were not strong enough to predict depth for point s_i, then $\nu_{i,s_i} = 0$ turns off the effect of the term $\left|R_{i,s_i}^T \alpha_i (x_{i,s_i}^T \theta_r) - 1\right|$.

The second term $g(.)$ models the relation between the plane parameters of two superpixels i and j. It uses pairs of points s_i and s_j to do so:

$$g(.) = \prod_{\{s_i, s_j\} \in N} h_{s_i, s_j}(.) \quad (2)$$

We will capture co-planarity, connectedness and co-linearity, by different choices of $h(.)$ and $\{s_i, s_j\}$.

Connected structure: We enforce this constraint by choosing s_i and s_j to be on the boundary of the superpixels i and j. As shown in Fig. 4b, penalizing the distance between two such points ensures that they remain fully connected. Note that in case of occlusion, the variables $y_{ij} = 0$, and hence the two superpixels will not be forced to be connected. The relative (fractional) distance between points s_i and s_j is penalized by

$$h_{s_i,s_j}(\alpha_i, \alpha_j, y_{ij}, R_i, R_j) = \exp\left(-y_{ij} |(R_{i,s_i}^T \alpha_i - R_{j,s_j}^T \alpha_j) \hat{d}|\right)$$

[1] The variable ν_{i,s_i} is an indicator of how good the image features are in predicting depth for point s_i in superpixel i. We learn ν_{i,s_i} from the monocular image features, by estimating the expected value of $|d_i - x_i^T \theta_r|/d_i$ as $\phi_r^T x_i$ with logistic response, with ϕ_r as the parameters of the model, features x_i and d_i as ground-truth depths.

In detail, $R^T_{i,s_i}\alpha_i = 1/d_{i,s_i}$ and $R^T_{j,s_j}\alpha_j = 1/d_{j,s_j}$; therefore, the term $(R^T_{i,s_i}\alpha_i - R^T_{j,s_j}\alpha_j)\hat{d}$ gives the fractional distance $|(d_{i,s_i} - d_{j,s_j})/\sqrt{d_{i,s_i}d_{j,s_j}}|$ for $\hat{d} = \sqrt{\hat{d}_{s_i}\hat{d}_{s_j}}$.

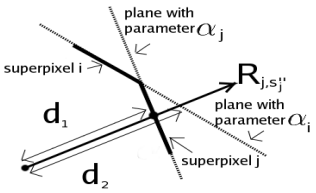

Figure 5. A 2-d illustration to explain the co-planarity term. The distance of the point s_j on superpixel j to the plane on which superpixel i lies along the ray R_{j,s_j}" is given by $d_1 - d_2$.

Co-planarity: We enforce the co-planar structure by choosing a third pair of points s_i'' and s_j'' in the center of each superpixel along with ones on the boundary. (Fig. 4c) To enforce co-planarity, we penalize the relative (fractional) distance of point s_j'' from the plane in which superpixel i lies, along the ray $R_{j,s_j''}$ (See Fig. 5).

$$h_{s_j''}(\alpha_i, \alpha_j, y_{ij}, R_{j,s_j''}) = \exp\left(-y_{ij}|(R^T_{j,s_j''}\alpha_i - R^T_{j,s_j''}\alpha_j)\hat{d}_{s_j''}|\right)$$

with $h_{s_i'',s_j''}(.) = h_{s_i''}(.)h_{s_j''}(.)$. Note that if the two superpixels are coplanar, then $h_{s_i'',s_j''} = 1$. To enforce co-planarity between two distant planes that are not connected, we can choose 3 pairs of points and use the above penalty.

Co-linearity: Finally, we enforce co-linearity constraint using this term, by choosing points along the sides of *long* straight lines. This also helps to capture relations between regions of the image that are not immediate neighbors.

Parameter Learning and MAP Inference: Exact parameter learning of the model is intractable; therefore, we use Multi-Conditional Learning (MCL) for approximate learning, where we model the probability as a product of multiple conditional likelihoods of individual densities. [11] We estimate the θ_r parameters by maximizing the conditional likelihood $\log P(\alpha|X, Y, R; \theta_r)$ of the training data, which can be written as a Linear Program (LP).

MAP inference of the plane parameters, i.e., maximizing the conditional likelihood $P(\alpha|X, Y, R; \theta)$, is efficiently performed by solving a LP. To solve the LP, we implemented an efficient method that uses the sparsity in our problem allowing inference in a few seconds.

4.3. Point-wise MRF

We present another MRF, in which we use points in the image as basic unit, instead of superpixels; and infer only their 3-d location. The nodes in this MRF are a dense grid of points in the image, where the value of each node represents its depth. The depths in this model are in log scale to emphasize fractional (relative) errors in depth. Unlike SCN's fixed rectangular grid, we use a deformable grid, aligned with structures in the image such as lines and corners to improve performance. Further, in addition to using the connected structure property (as in SCN), our model also captures co-planarity and co-linearity. Finally, we use logistic response to identify occlusion and folds, whereas SCN learned the variances.

In the MRF below, the first term $f(.)$ models the relation between depths and the image features as $f_\theta(d_i, x_i, y_i) = \exp\left(-y_i|d_i - x_i^T\theta_{r(i)}|\right)$. The second term $g(.)$ models connected structure by penalizing differences in depth of neighboring points as $g(d_i, d_j, y_{ij}, R_i, R_j) = \exp\left(-y_{ij}|(R_i d_i - R_j d_j)|\right)$. The third term $h(.)$ depends on three points i,j and k, and models co-planarity and co-linearity. (Details omitted due to space constraints; see full paper for details.)

$$P(d|X, Y, R; \theta) = \frac{1}{Z}\prod_i f_\theta(d_i, x_i, y_i) \prod_{i,j\in N} g(d_i, d_j, y_{ij}, R_i, R_j)$$
$$\prod_{i,j,k\in N} h(d_i, d_j, d_k, y_{ijk}, R_i, R_j, R_k)$$

where, $d_i \in \mathbb{R}$ is the depth at a point i. x_i are the image features at point i. MAP inference of depths, i.e. maximizing $\log P(d|X, Y, R; \theta)$ is performed by solving a linear program (LP). However, the size of LP in this MRF is larger than in the Plane Parameter MRF.

5. Features

For each superpixel, we compute a battery of features to capture some of the monocular cues discussed in Section 2. We also compute features to predict meaningful boundaries in the images, such as occlusion. Note that this is in contrast with some methods that rely on very specific features, e.g. computing parallel lines on a plane to determine vanishing points. Relying on a large number of different types of features helps our algorithm to be more robust and generalize to images that are very different from the training set.

5.1. Monocular Image Features

For each superpixel at location i, we compute texture-based summary statistic features, and superpixel shape and location based features.[2] (See Fig. 6.) We attempt to capture more "contextual" information by also including features from neighboring superpixels (4 in our experiments), and at multiple spatial scales (3 in our experiments). (See Fig. 6.) The features, therefore, contain information from a larger portion of the image, and thus are more expressive

[2]Similar to SCN, we use the output of each of the 17 (9 Laws masks, 2 color channels in YCbCr space and 6 oriented edges) filters $F_n(x, y)$, $n = 1, ..., 17$ as: $E_i(n) = \sum_{(x,y)\in S_i} |I(x, y) * F_n(x, y)|^k$, where k = 2,4 gives the energy and kurtosis respectively. This gives a total of 34 values for each superpixel. We compute features for a superpixel to improve performance over SCN, who computed them for fixed rectangular patches.

Our superpixel shape and location based features included the shape and location based features in Section 2.2 of [9], and also the eccentricity of the superpixel.

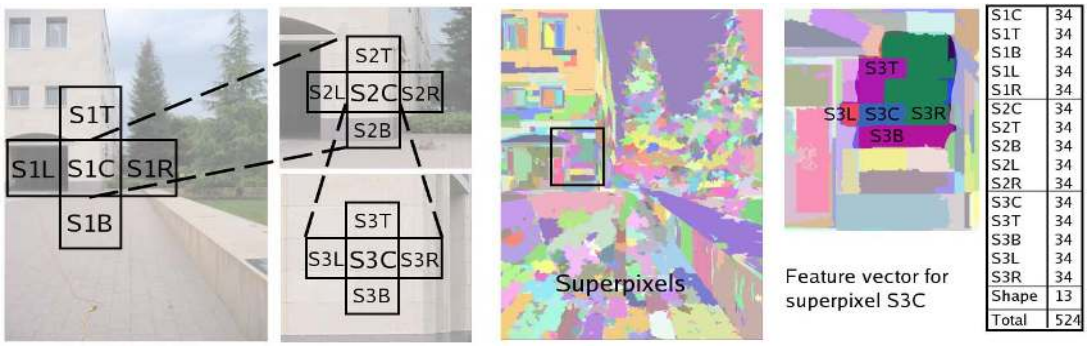

Figure 6. The feature vector for a superpixel, which includes immediate and distant neighbors in multiple scales. (*Best viewed in color.*)

than just local features. This makes the feature vector x_i of a superpixel 524 dimensional.

5.2. Features for Boundaries

Another strong cue for 3-d structure perception is boundary information. If two neighbor superpixels of an image display different features, humans would often perceive them to be parts of different objects; therefore an edge between two superpixels with distinctly different features, is a candidate for a occlusion boundary or a fold. To compute the features x_{ij} between superpixels i and j, we first generate 14 different segmentations for each image for 2 different scales for 7 different properties: textures, color, and edges. Each element of our 14 dimensional feature vector x_{ij} is then an indicator if two superpixels i and j lie in the same segmentation. The features x_{ij} are the input to the classifier for the occlusion boundaries and folds. (see Section 4.1)

6. Incorporating Object Information

In this section, we will discuss how our model can also incorporate other information that might be available, for example, from object recognizers. In [8], Hoiem et al. used knowledge of objects and their location to improve the estimate of the horizon. In addition to estimating the horizon, the knowledge of objects and their location in the scene gives strong cues regarding the 3-d structure of the scene. For example, a person is more likely to be on top of the ground, rather than under it.

Here we give some examples of such constraints, and describe how we can encode them in our MRF:

(a) *"Object A is on top of object B"*
This constraint could be encoded by restricting the points $s_i \in \mathbb{R}^3$ on object A to be on top of the points $s_j \in \mathbb{R}^3$ on object B, i.e., $s_i^T \hat{z} \geq s_j^T \hat{z}$ (if $\hat{z}$ denotes the "up" vector). In practice, we actually use a probabilistic version of this constraint. We represent this inequality in plane-parameter space ($s_i = R_i d_i = R_i/(\alpha_i^T R_i)$). To penalize the fractional error $\xi = \left(R_i^T \hat{z} R_j^T \alpha_j - R_j^T \hat{z} R_i \alpha_i\right) \hat{d}$ (the constraint corresponds to $\xi \geq 0$), we choose an MRF potential $h_{s_i,s_j}(.) = \exp\left(-y_{ij}\left(\xi + |\xi|\right)\right)$, where y_{ij} represents the uncertainty in the object recognizer output. Note that for $y_{ij} \to \infty$ (corresponding to certainty in the object recognizer), this becomes a "hard" constraint $R_i^T \hat{z}/(\alpha_i^T R_i) \geq R_j^T \hat{z}/(\alpha_j^T R_j)$.

In fact, we can also encode other similar spatial-relations by choosing the vector $\hat{z}$ appropriately. For example, a constraint *"Object A is in front of Object B"* can be encoded by choosing $\hat{z}$ to be the ray from the camera to the object.

(b) *"Object A is attached to Object B"*
For example, if the ground-plane is known from a recognizer, then many objects would be more likely to be "attached" to the ground plane. We easily encode this by using our connected-structure constraint (Section 4).

(c) *Known plane orientation*
If orientation of a plane is roughly known, e.g. that a person is more likely to be "vertical", then it can be easily encoded by adding to Eq. 1 a term $f(\alpha_i) = \exp\left(-\nu_i |\alpha_i^T \hat{z}|\right)$; here, ν_i represents the confidence, and $\hat{z}$ represents the up vector.

We will describe our results using these constraints in Section 7.3.

7. Experiments

7.1. Data collection

We used a custom-built 3-D scanner to collect images and their corresponding depthmaps using lasers. We collected a total of 534 images+depthmaps, with an image resolution of 2272x1704 and a depthmap resolution of 55x305; and used 400 for training our model.

We tested our model on 134 images collected using our 3-d scanner, and also on 588 internet images. The images on the internet were collected by issuing keywords on Google image search. To collect data and to perform the evaluation of the algorithms in a completely unbiased manner, a person *not* associated with the project was asked to collect images of environments (greater than 800x600 size). The person chose the following keywords to collect the images: campus, garden, park, house, building, college, university, church, castle, court, square, lake, temple, scene.

7.2. Results and Discussion

We performed an extensive evaluation of our algorithm on 588 internet test images, and 134 test images collected using the laser scanner.

In Table 1, we compare the following algorithms:
(a) Baseline: Both for depth-MRF (Baseline-1) and plane parameter MRF (Baseline-2). The Baseline MRF is trained

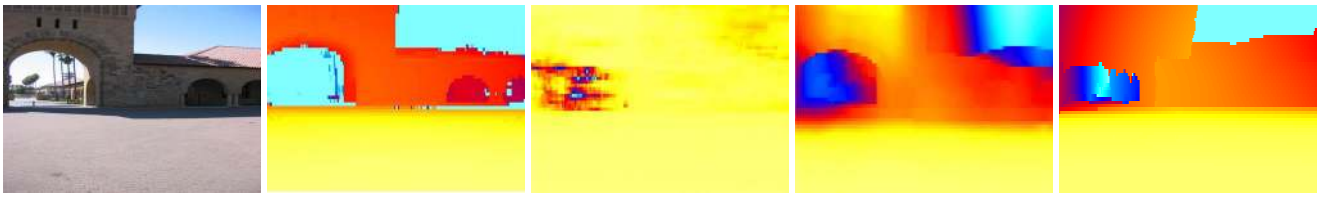

Figure 7. (a) Original Image, (b) Ground truth depthmap, (c) Depth from image features only, (d) Point-wise MRF, (e) Plane parameter MRF. (*Best viewed in Color*)

Table 1. Results: Quantitative comparison of various methods.

METHOD	CORRECT (%)	% PLANES CORRECT	$\log_{10}$	REL
SCN	NA	NA	0.198	0.530
HEH	33.1%	50.3%	0.320	1.423
BASELINE-1	0%	NA	0.300	0.698
NO PRIORS	0%	NA	0.170	0.447
POINT-WISE MRF	23%	NA	**0.149**	0.458
BASELINE-2	0%	0%	0.334	0.516
NO PRIORS	0%	0%	0.205	0.392
CO-PLANAR	45.7%	57.1%	0.191	0.373
PP-MRF	**64.9%**	**71.2%**	0.187	**0.370**

without any image features, and thus reflects a "prior" depthmap of sorts.

(b) Our Point-wise MRF: with and without constraints (connectivity, co-planar and co-linearity).

(c) Our Plane Parameter MRF (PP-MRF): without any constraint, with co-planar constraint only, and the full model.

(d) Saxena et al. (SCN), applicable for quantitative errors.

(e) Hoiem et al. (HEH). For fairness, we scale and shift their depthmaps before computing the errors to match the global scale of our test images. Without the scaling and shifting, their error is much higher (7.533 for relative depth error).

We compare the algorithms on the following metrics: (a) Average depth error on a log-10 scale, (b) Average relative depth error, (We give these numerical errors on only the 134 test images that we collected, because ground-truth depths are not available for internet images.) (c) % of models qualitatively correct, (d) % of major planes correctly identified.[3]

Table 1 shows that both of our models (Point-wise MRF and Plane Parameter MRF) outperform both SCN and HEH in quantitative accuracy in depth prediction. Plane Parameter MRF gives better relative depth accuracy, and produces sharper depthmaps. (Fig. 7) Table 1 also shows that by capturing the image properties of connected structure, coplanarity and colinearity, the models produced by the algorithm become significantly better. In addition to reducing quantitative errors, PP-MRF does indeed produce significantly better 3-d models. When producing 3-d flythroughs, even a small number of erroneous planes make the 3-d model visually unacceptable, even though the quantitative numbers may still show small errors.

[3]We define a model as correct when for 70% of the major planes in the image (major planes occupy more than 15% of the area), the plane is in correct relationship with its nearest neighbors (i.e., the relative orientation of the planes is within 30 degrees). Note that changing the numbers, such as 70% to 50% or 90%, 15% to 10% or 30%, and 30 degrees to 20 or 45 degrees, gave similar trends in the results.

Table 2. Percentage of images for which HEH is better, our PP-MRF is better, or it is a tie.

ALGORITHM	%BETTER
TIE	15.8%
HEH	22.1%
PP-MRF	**62.1%**

Our algorithm gives qualitatively correct models for 64.9% of images as compared to 33.1% by HEH. The qualitative evaluation was performed by a person not associated with the project following the guidelines in Footnote 3. HEH generate a "photo-popup" effect by folding the images at "ground-vertical" boundaries—an assumption which is not true for a significant number of images; therefore, their method fails in those images. Some typical examples of the 3-d models are shown in Fig. 8. (Note that all the *test* cases shown in Fig. 1, 8 and 9 are from the dataset downloaded from the internet, except Fig. 9a which is from the laser-test dataset.) These examples also show that our models are often more detailed than HEH, in that they are often able to model the scene with a multitude (over a hundred) of planes.

We performed a further comparison to HEH. Even when both HEH and our algorithm is evaluated as qualitatively correct on an image, one result could still be superior. Therefore, we asked the person to compare the two methods, and decide which one is better, or is a tie.[4] Table 2 shows that our algorithm performs better than HEH in 62.1% of the cases. Full documentation describing the details of the unbiased human judgment process, along with the 3-d flythroughs produced by our algorithm and HEH, is available online at:

http://ai.stanford.edu/∼asaxena/reconstruction3d

Some of our models, e.g. in Fig. 9j, have cosmetic defects—e.g. stretched texture; better texture rendering techniques would make the models more visually pleasing. In some cases, a small mistake (i.e., one person being detected as far-away in Fig. 9h) makes the model look bad; and hence be evaluated as "incorrect."

Our algorithm, trained on images taken in a small geographical area in our university, was able to predict

[4]To compare the algorithms, the person was asked to count the number of errors made by each algorithm. We define an error when a major plane in the image (occupying more than 15% area in the image) is in wrong location with respect to its neighbors, or if the orientation of the plane is more than 30 degrees wrong. For example, if HEH fold the image at incorrect place (see Fig. 8, image 2), then it is counted as an error. Similarly, if we predict top of a building as far and the bottom part of building near, making the building tilted—it would count as an error.

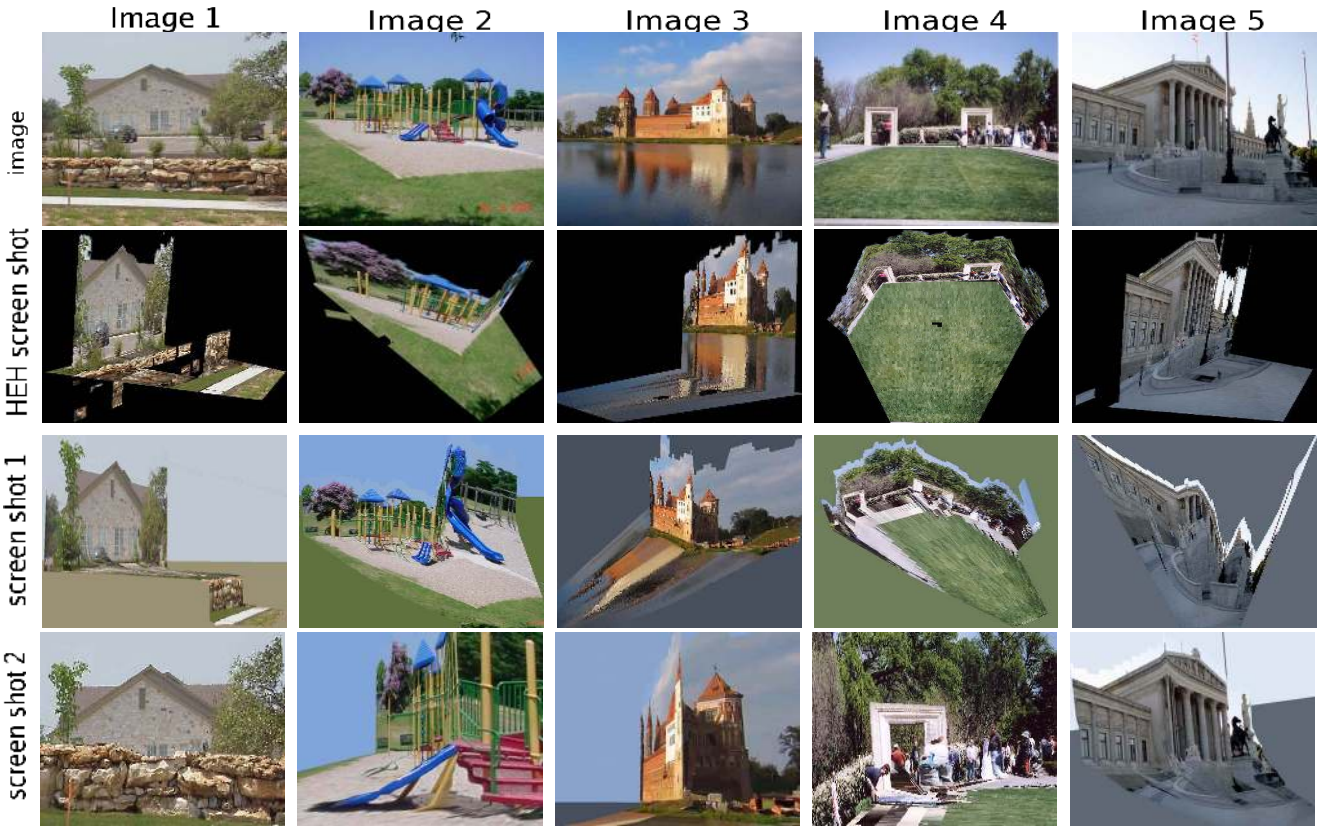

Figure 8. Typical results from HEH and our algorithm. **Row 1**: Original Image. **Row 2**: 3-d model generated by HEH, **Row 3 and 4**: 3-d model generated by our algorithm. (Note that the screenshots cannot be simply obtained from the original image by an affine transformation.) In **image 1**, HEH makes mistakes in some parts of the foreground rock, while our algorithm predicts the correct model; with the rock occluding the house, giving a novel view. In **image 2**, HEH algorithm detects a wrong ground-vertical boundary; while our algorithm not only finds the correct ground, but also captures a lot of non-vertical structure, such as the blue slide. In **image 3**, HEH is confused by the reflection; while our algorithm produces a correct 3-d model. In **image 4**, HEH and our algorithm produce roughly equivalent results—HEH is a bit more visually pleasing and our model is a bit more detailed. In **image 5**, both HEH and our algorithm fail; HEH just predict one vertical plane at a incorrect location. Our algorithm predicts correct depths of the pole and the horse, but is unable to detect their boundary; hence making it qualitatively incorrect.

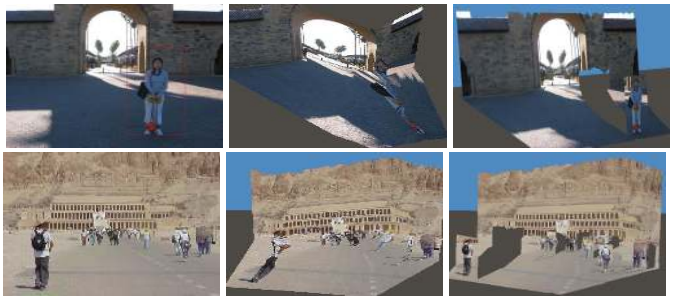

Figure 10. (Left) Original Images, (Middle) Snapshot of the 3-d model without using object information, (Right) Snapshot of the 3-d model that uses object information.

qualitatively correct 3-d models for a large variety of environments—for example, ones that have hills, lakes, and ones taken at night, and even paintings. (See Fig. 9 and the website.) We believe, based on our experiments varying the number of training examples (not reported here), that having a larger and more diverse set of training images would improve the algorithm significantly.

7.3. Results using Object Information

We also performed experiments in which information from object recognizers was incorporated into the MRF for inferring a 3-d model (Section 6). In particular, we implemented a recognizer (based on the features described in Section 5) for ground-plane, and used the Dalal-Triggs Detector [2] to detect pedestrains. For these objects, we encoded the (a), (b) and (c) constraints described in Section 6. Fig. 10 shows that using the pedestrian and ground detector improves the accuracy of the 3-d model. Also note that using "soft" constraints in the MRF (Section 6), instead of "hard" constraints, helps in estimating correct 3-d models even if the object recognizer makes a mistake.

8. Conclusions

We presented an algorithm for inferring detailed 3-d structure from a single still image. Compared to previous approaches, our model creates 3-d models which are both quantitatively more accurate and more visually pleasing. We model both the location and orientation of small

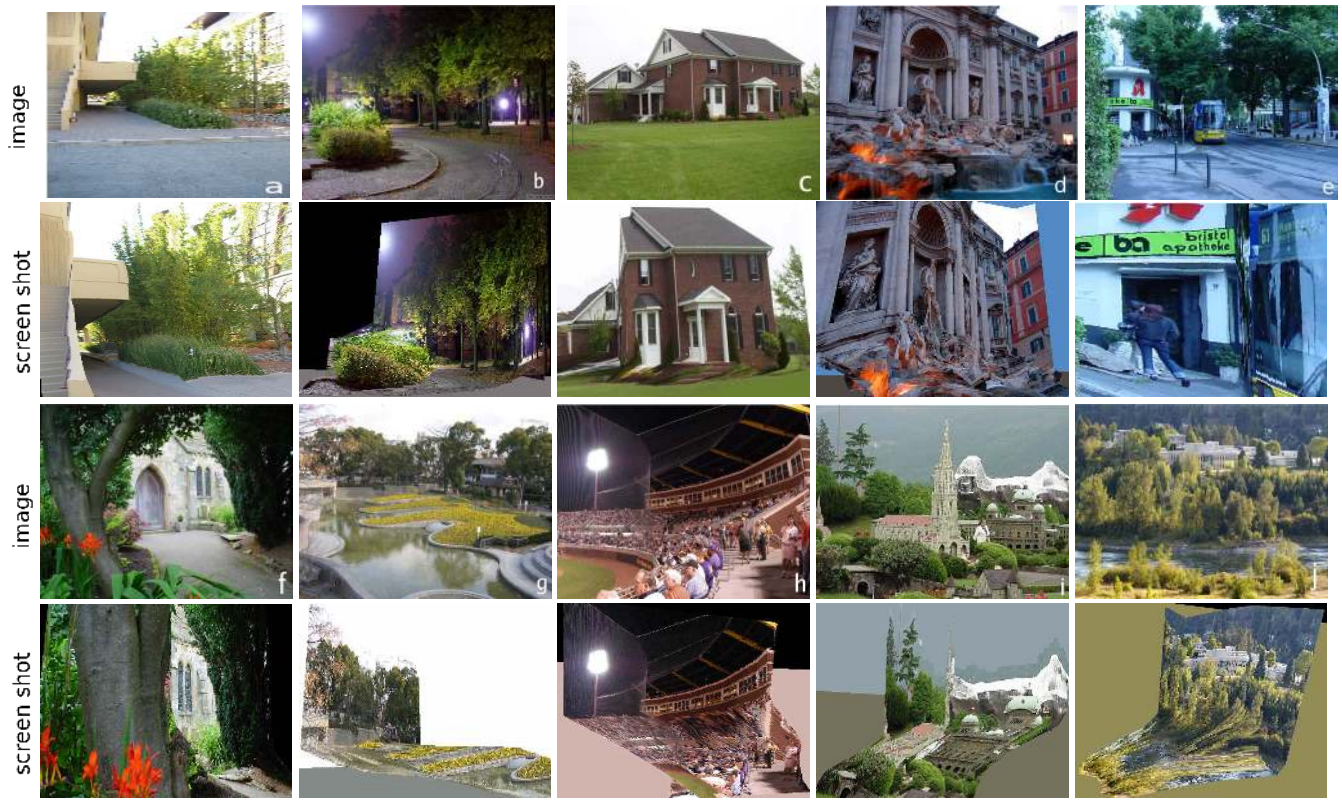

Figure 9. Typical results from our algorithm. Original image (top), and a screenshot of the 3-d flythrough generated from the image (bottom of the image). The first 7 images (a-g) were evaluated as "correct" and the last 3 (h-j) were evaluated as "incorrect."

homogenous regions in the image, called "superpixels," using an MRF. Our model, trained via supervised learning, estimates plane parameters using image features, and also reasons about relationships between various parts of the image. MAP inference for our model is efficiently performed by solving a linear program. Other than assuming that the environment is made of a number of small planes, we do not make any explicit assumptions about the structure of the scene, such as the "ground-vertical" planes assumption by Delage et al. and Hoiem et al.; thus our model is able to generalize well, even to scenes with significant non-vertical structure. We created visually pleasing 3-d models autonomously for 64.9% of the 588 internet images, as compared to Hoiem et al.'s performance of 33.1%. Our models are also quantitatively more accurate than prior art. Finally, we also extended our model to incorporate information from object recognizers to produce better 3-d models.

Acknowledgments: We thank Rajiv Agarwal and Jamie Schulte for help in collecting data. We also thank James Diebel, Jeff Michels and Alyosha Efros for helpful discussions. This work was supported by the National Science Foundation under award CNS-0551737, and by the Office of Naval Research under MURI N000140710747.

References

[1] A. Criminisi, I. Reid, and A. Zisserman. Single view metrology. *IJCV*, 40:123–148, 2000.

[2] N. Dalai and B. Triggs. Histogram of oriented gradients for human detection. In *CVPR*, 2005.

[3] E. Delage, H. Lee, and A. Ng. Automatic single-image 3d reconstructions of indoor manhattan world scenes. In *ISRR*, 2005.

[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. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59, 2004.

[6] D. A. Forsyth and J. Ponce. *Computer Vision : A Modern Approach*. Prentice Hall, 2003.

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

[8] D. Hoiem, A. Efros, and M. Hebert. Putting objects in perspective. In *CVPR*, 2006.

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

[10] R. Koch, M. Pollefeys, and L. V. Gool. Multi viewpoint stereo from uncalibrated video sequences. In *ECCV*, 1998.

[11] A. McCalloum, C. Pal, G. Druck, and X. Wang. Multi-conditional learning: generative/discriminative training for clustering and classification. In *AAAI*, 2006.

[12] J. Michels, A. Saxena, and A. Y. Ng. High speed obstacle avoidance using monocular vision & reinforcement learning. In *ICML*, 2005.

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

[14] A. Saxena, S. H. Chung, and A. Y. Ng. 3-d depth reconstruction from a single still image. *IJCV*, 2007.

[15] A. Saxena, J. Schulte, and A. Y. Ng. Depth estimation using monocular and stereo cues. In *IJCAI*, 2007.

[16] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47, 2002.

[17] B. A. Wandell. *Foundations of Vision*. Sinauer Associates, Sunderland, MA, 1995.

[18] R. Zhang, P. Tsai, J. Cryer, and M. Shah. Shape from shading: A survey. *IEEE PAMI*, 21:690–706, 1999.