3-D Depth Reconstruction from a Single Still Image

Ashutosh Saxena, Sung H. Chung, Andrew Y. Ng

Computer Science Department
Stanford University, Stanford, CA 94305
{asaxena,codedeft,ang}@cs.stanford.edu

Abstract

We consider the task of 3-d depth estimation from a single still image. We take a supervised learning approach to this problem, in which we begin by collecting a training set of monocular images (of unstructured indoor and outdoor environments which include forests, sidewalks, trees, buildings, etc.) and their corresponding ground-truth depthmaps. Then, we apply supervised learning to predict the value of the depthmap as a function of the image. Depth estimation is a challenging problem, since local features alone are insufficient to estimate depth at a point, and one needs to consider the global context of the image. Our model uses a hierarchical, multiscale Markov Random Field (MRF) that incorporates multiscale local- and global-image features, and models the depths and the relation between depths at different points in the image. We show that, even on unstructured scenes, our algorithm is frequently able to recover fairly accurate depthmaps. We further propose a model that incorporates both monocular cues and stereo (triangulation) cues, to obtain significantly more accurate depth estimates than is possible using either monocular or stereo cues alone.

1 Introduction

Recovering 3-d depth from images is a basic problem in computer vision, and has important applications in robotics, scene understanding and 3-d reconstruction. Most work on visual 3-d reconstruction has focused on binocular vision (stereopsis) [1] and on other algorithms that require multiple images, such as structure from motion [2] and depth from defocus [3]. These algorithms consider only the geometric (triangulation) differences. Beyond stereo/triangulation cues, there are also numerous *monocular* cues—such as texture variations and gradients, defocus, color/haze, etc.—that contain useful and important depth information. Even though humans perceive depth by seamlessly combining many of these stereo and monocular cues, most work on depth estimation has focused on stereovision.

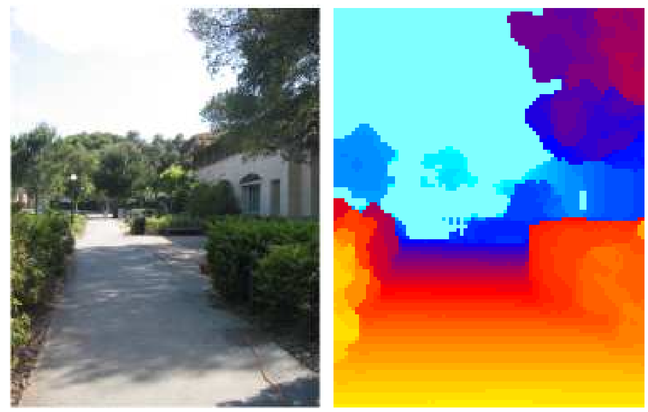

Figure 1: (a) A single still image, and (b) the corresponding (ground-truth) depthmap. Colors in the depthmap indicate estimated distances from the camera.

Depth estimation from a *single* still image is a difficult task, since depth typically remains ambiguous given only local image features. Thus, our algorithms must take into account the global structure of the image, as well as use prior knowledge about the scene. We also view depth estimation as a small but crucial step towards the larger goal of image understanding, in that it will help in tasks such as understanding the spatial layout of a scene, finding walkable areas in a scene, detecting objects, etc. In this paper, we apply supervised learning to the problem of estimating depthmaps (Fig. 1b) from a single still image (Fig. 1a) of a variety of unstructured environments, both indoor and outdoor, containing forests, sidewalks, buildings, people, bushes, etc.

Our approach is based on modeling depths and relationships between depths at multiple spatial scales using a hierarchical, multiscale Markov Random Field (MRF). Taking a supervised learning approach to the problem of depth estimation, we used a 3-d scanner to collect training data, which comprised a large set of images and their corresponding ground-truth depthmaps. (This data has been made publically available on the internet.) Using this training set, we model the conditional distribution of the depths given the monocular image features. Though learning in our MRF model is approximate, MAP inference is tractable via linear programming.

We further consider how monocular cues from a single image can be incorporated into a stereo system. We believe that monocular cues and (purely geometric) stereo cues give largely orthogonal, and therefore complementary, types of information about depth. We show that combining both monocular and stereo cues gives better depth estimates than is obtained with either alone.

We also apply these ideas to autonomous obstacle avoidance. Using a simplified version of our algorithm, we drive a small remote-controlled car at high speeds through various unstructured outdoor environments containing both man-made and natural obstacles.

This paper is organized as follows. Section 2 gives an overview of various methods used for 3-d depth reconstruction. Section 3 describes some of the visual cues used by humans for depth perception, and Section 4 describes the image features used to capture monocular cues. We describe our probabilistic model in Section 5. In Section 6.1, we describe our setup for collecting aligned image and laser data. The results of depth prediction on single images are presented in Section 6.2. Section 6.2 also describes the use of a simplified version of our algorithm in driving a small remote-controlled car autonomously. We describe how we incorporate monocular and stereo cues into our model in Section 7. Finally, we conclude in Section 8.

2 Related Work

Although our work mainly focuses on depth estimation from a single still image, there are many other 3-d reconstruction techniques, such as: explicit measurements with laser or radar sensors [4], using two (or more than two) images [1], and using video sequences [5]. Among the vision-based approaches, most work has focused on stereovision (see [1] for a review), and on other algorithms that require multiple images, such as optical flow [6], structure from motion [2] and depth from defocus [3]. Frueh and Zakhor [7] constructed 3d city models by merging ground-based and airborne views. A large class of algorithms reconstruct the 3-d shape of known objects, such as human bodies, from images and laser data [8; 9]. Structured lighting [10] offers another method for depth reconstruction

There are some algorithms that can perform depth reconstruction from single images in very specific settings. Nagai et al. [11] performed surface reconstruction from single images for known, fixed, objects such as hands and faces. Methods such as shape from shading [12; 13] and shape from texture [14; 15; 16] generally assume uniform color and/or texture,[1] and hence would perform very poorly on the complex, unconstrained, highly textured images that we consider. Hertzmann et al. [17] reconstructed high quality 3-d models from several images, but they required that the images also contain "assistant" objects of known shapes next to the target object. Torresani et al. [18] worked on reconstructing non-rigid surface shapes from video sequences. Torralba and Oliva [19] studied the Fourier spectrum of the images to compute the mean depth of a scene. Michels, Saxena & Ng [20] used supervised learning to estimate 1-d distances to obstacles, for the application of autonomously driving a small car. Delage, Lee and Ng [21; 22] generated 3-d models of indoor environments containing only walls and floor, from single monocular images. Single view metrology [23] assumes that vanishing lines and points are known in a scene, and calculates angles between parallel lines to infer 3-d structure from Manhattan images.

We presented a method for learning depths from a single image in [24] and extended our method to improve stereo vision using monocular cues in [25]. In work that is contemporary to ours, Hoiem, Efros and Herbert [26; 27] built a simple "pop-up" type 3-d model from an image by classifying the image into ground, vertical and sky. Their method, which assumes a simple "ground-vertical" structure of the world, fails on many environments that do not satisfy this assumption and also does not give accurate metric depthmaps. Building on these concepts of single image 3-d reconstruction, Hoiem, Efros and Hebert [28] and Sudderth et al. [29] integrated learning-based object recognition with 3-d scene representations. Saxena et al. [30] extended these ideas to create 3-d models that are both visually pleasing as well as quantitatively accurate.

Our approach draws on a large number of ideas from computer vision such as feature computation and multiscale representation of images. A variety of image features and representations have been used by other authors, such as Gabor filters [31], wavelets [32], SIFT features [33], etc. Many of these image features are used for purposes such as recognizing objects [34; 35], faces [36], facial expressions [37], grasps [38]; image segmentation [39], computing the visual gist of a scene [40] and computing sparse representations of natural images [41]. Stereo and monocular image features have been used together for object recognition and image segmentation [42].

Our approach is based on learning a Markov Random Field (MRF) model. MRFs are a workhorse of machine learning, and have been successfully applied to numerous problems in which local features were insufficient and more contextual information had to be used. Examples include image denoising [43], stereo vision and image segmentation [1], text segmentation [44], object classification [34], and image labeling [45]. For the application of identifying man-made structures in natural images, Kumar and Hebert used a discriminative random fields algorithm [46]. Since MRF learning is intractable in general, most of these models are trained using pseudo-likelihood; sometimes the models' parameters are also hand-tuned.

[1] Also, most of these algorithms assume Lambertian surfaces, which means the appearance of the surface does not change with viewpoint.

3 Visual Cues for Depth Perception

Humans use numerous visual cues to perceive depth. Such cues are typically grouped into four distinct categories: monocular, stereo, motion parallax, and focus cues [47; 48]. Humans combine these cues to understand the 3-d structure of the world [49; 50; 51; 47]. Below, we describe these cues in more detail. Our probabilistic model will attempt to capture a number of monocular cues (Section 5), as well as stereo triangulation cues (Section 7).

3.1 Monocular Cues

Humans use monocular cues such as texture variations, texture gradients, interposition, occlusion, known object sizes, light and shading, haze, defocus, etc. [52] For example, many objects' texture will look different at different distances from the viewer. Texture gradients, which capture the distribution of the direction of edges, also help to indicate depth [16]. For example, a tiled floor with parallel lines will appear to have tilted lines in an image. The distant patches will have larger variations in the line orientations, and nearby patches with almost parallel lines will have smaller variations in line orientations. Similarly, a grass field when viewed at different distances will have different texture gradient distributions. Haze is another depth cue, and is caused by atmospheric light scattering [53].

Many monocular cues are "contextual information," in the sense that they are global properties of an image and cannot be inferred from small image patches. For example, occlusion cannot be determined if we look at just a small portion of an occluded object. Although local information such as the texture and color of a patch can give some information about its depth, this is usually insufficient to accurately determine its absolute depth. For another example, if we take a patch of a clear blue sky, it is difficult to tell if this patch is infinitely far away (sky), or if it is part of a blue object. Due to ambiguities like these, one needs to look at the *overall* organization of the image to determine depths.

3.2 Stereo Cues

Each eye receives a slightly different view of the world and stereo vision combines the two views to perceive 3-d depth [54]. An object is projected onto different locations on the two retinae (cameras in the case of a stereo system), depending on the distance of the object. The retinal (stereo) disparity varies with object distance, and is inversely proportional to the distance of the object. Disparity is typically not an effective cue for estimationg small depth variations of objects that are far away.

3.3 Motion Parallax and Focus Cues

As an observer moves, closer objects appear to move more than further objects. By observing this phenomenon, called motion parallax, one can estimate the relative distances in a scene [55]. Humans have the ability to change the focal lengths of the eye lenses by controlling the curvature of lens, thus helping them to focus on objects at different distances. The focus, or accomodation, cue refers to the ability to estimate the distance of an object from known eye lens configuration and the sharpness of the image of the object [56].

Figure 2: The convolutional filters used for texture energies and gradients. The first nine are 3x3 Laws' masks. The last six are the oriented edge detectors spaced at 30^0 intervals. The nine Laws' masks are used to perform local averaging, edge detection and spot detection.

4 Feature Vector

In our approach, we divide the image into small rectangular patches, and estimate a single depth value for each patch. We use two types of features: *absolute* depth features—used to estimate the absolute depth at a particular patch—and *relative* features, which we use to estimate relative depths (magnitude of the difference in depth between two patches). These features try to capture two processes in the human visual system: local feature processing (absolute features), such as that the sky is far away; and continuity features (relative features), a process by which humans understand whether two adjacent patches are physically connected in 3-d and thus have similar depths.[2]

We chose features that capture three types of local cues: texture variations, texture gradients, and color. Texture information is mostly contained within the image intensity channel [54],[3] so we apply Laws' masks [57; 20] to this channel to compute the texture energy (Fig. 2). Haze is reflected in the low frequency information in the color channels, and we capture this by applying a local averaging filter (the first Laws' mask) to the color channels. Lastly, to compute an estimate of texture gradient that is robust to noise, we convolve the intensity channel with six oriented edge filters (shown in Fig. 2).

One can envision including more features to capture other cues. For example, to model atmospheric effects such as fog and haze, features computed from the physics of light scattering [53] could also be included. Similarly, one can also include features based on surface-shading [13].

4.1 Features for absolute depth

We first compute summary statistics of a patch i in the image $I(x, y)$ as follows. We use the output of each of the 17 (9 Laws' masks, 2 color channels and 6 texture gradients) filters F_n, $n = 1, ..., 17$ as: $E_i(n) = \sum_{(x,y)\in \text{patch}(i)} |I * F_n|^k$, where $k \in \{1, 2\}$ give the sum

[2]If two neighboring patches of an image display similar features, humans would often perceive them to be parts of the same object, and therefore to have similar depth values.

[3]We represent each image in YCbCr color space, where Y is the intensity channel, and Cb and Cr are the color channels.

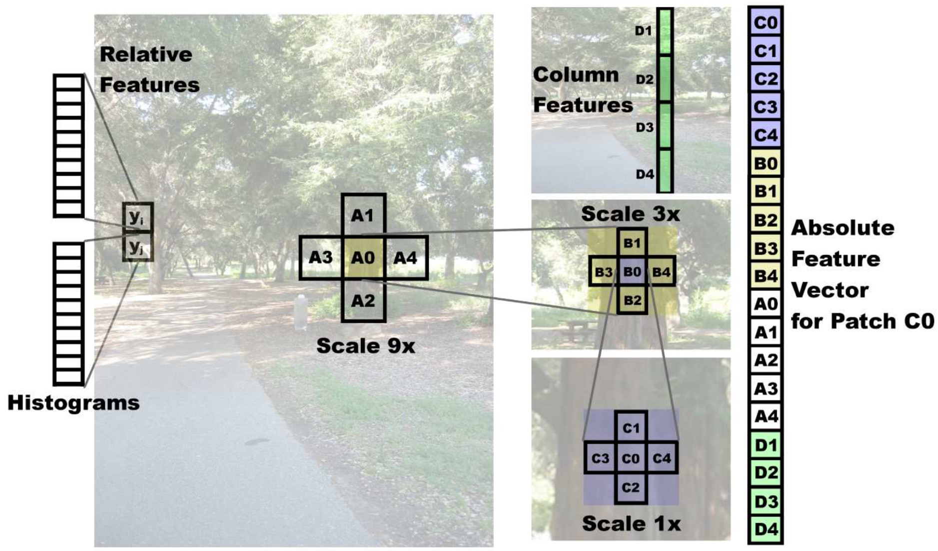

Figure 3: The absolute depth feature vector for a patch, which includes features from its immediate neighbors and its more distant neighbors (at larger scales). The relative depth features for each patch use histograms of the filter outputs.

absolute energy and sum squared energy respectively.[4] This gives us an initial feature vector of dimension 34.

To estimate the absolute depth at a patch, local image features centered on the patch are insufficient, and one has to use more global properties of the image. We attempt to capture this information by using image features extracted at multiple spatial scales (image resolutions).[5] (See Fig. 3.) Objects at different depths exhibit very different behaviors at different resolutions, and using multiscale features allows us to capture these variations [58]. For example, blue sky may appear similar at different scales, but textured grass would not. In addition to capturing more global information, computing features at multiple spatial scales also helps to account for different relative sizes of objects. A closer object appears larger in the image, and hence will be captured in the larger scale features. The same object when far away will be small and hence be captured in the small scale features. Features capturing the scale at which an object appears may therefore give strong indicators of depth.

To capture additional global features (e.g. occlusion relationships), the features used to predict the depth of a particular patch are computed from that patch as well as the four neighboring patches. This is repeated at each of the three scales, so that the feature vector at a patch includes features of its immediate neighbors, its neighbors at a larger spatial scale (thus capturing image features that are slightly further away in the image plane), and again its neighbors at an even larger spatial scale; this is illustrated in Fig. 3. Lastly, many structures (such as trees and buildings) found in outdoor scenes show vertical structure, in the sense that they are vertically connected to themselves (things cannot hang in empty air). Thus, we also add to the features of a patch additional summary features of the column it lies in.

For each patch, after including features from itself and its 4 neighbors at 3 scales, and summary features for its 4 column patches, our absolute depth feature vector x is $19 * 34 = 646$ dimensional.

4.2 Features for relative depth

We use a different feature vector to learn the dependencies between two neighboring patches. Specifically, we compute a 10-bin histogram of each of the 17 filter outputs $|I * F_n|$, giving us a total of 170 features y_{is} for each patch i at scale s. These features are used to estimate how the depths at two different locations are related. We believe that learning these estimates requires less global information than predicting absolute depth, but more detail from the individual patches. For example, given two adjacent patches of a distinctive, unique, color and texture, we may be able to safely conclude that they are part of the same object, and thus that their depths are close, even without more global features. Hence, our relative depth features y_{ijs} for two neighboring patches i and j at scale s will be the differences between their histograms, i.e., $y_{ijs} = y_{is} - y_{js}$.

5 Probabilistic Model

Since local images features are by themselves usually insufficient for estimating depth, the model needs to reason

[4]Our experiments using $k \in \{1, 2, 4\}$ did not improve performance noticeably.

[5]The patches at each spatial scale are arranged in a grid of equally sized non-overlapping regions that cover the entire image. We use 3 scales in our experiments.

$$P_G(d|X;\theta,\sigma) = \frac{1}{Z_G}\exp\left(-\sum_{i=1}^{M}\frac{(d_i(1)-x_i^T\theta_r)^2}{2\sigma_{1r}^2} - \sum_{s=1}^{3}\sum_{i=1}^{M}\sum_{j\in N_s(i)}\frac{(d_i(s)-d_j(s))^2}{2\sigma_{2rs}^2}\right) \quad (1)$$

$$P_L(d|X;\theta,\lambda) = \frac{1}{Z_L}\exp\left(-\sum_{i=1}^{M}\frac{|d_i(1)-x_i^T\theta_r|}{\lambda_{1r}} - \sum_{s=1}^{3}\sum_{i=1}^{M}\sum_{j\in N_s(i)}\frac{|d_i(s)-d_j(s)|}{\lambda_{2rs}}\right) \quad (2)$$

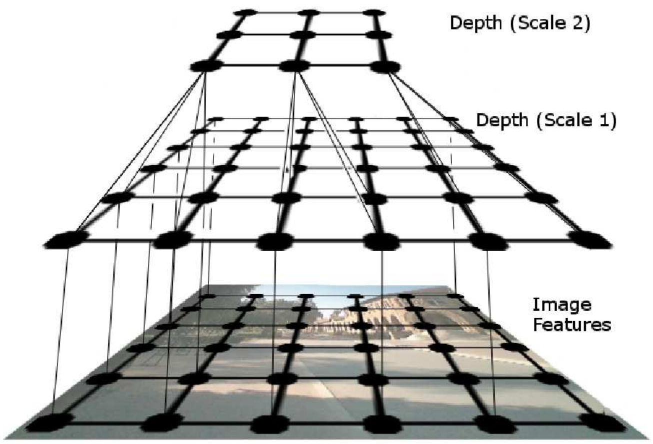

Figure 4: The multiscale MRF model for modeling relation between features and depths, relation between depths at same scale, and relation between depths at different scales. (Only 2 out of 3 scales, and a subset of the edges, are shown.)

more globally about the spatial structure of the scene. We capture the spatial structure of the image by modeling the relationships between depths in different parts of the image. Although the depth of a particular patch depends on the features of the patch, it is also related to the depths of other parts of the image. For example, the depths of two adjacent patches lying in the same building will be highly correlated. We will use a hierarchical multiscale Markov Random Field (MRF) to model the relationship between the depth of a patch and the depths of its neighboring patches (Fig. 4). In addition to the interactions with the immediately neighboring patches, there are sometimes also strong interactions between the depths of patches which are not immediate neighbors. For example, consider the depths of patches that lie on a large building. All of these patches will be at similar depths, even if there are small discontinuities (such as a window on the wall of a building). However, when viewed at the smallest scale, some adjacent patches are difficult to recognize as parts of the same object. Thus, we will also model interactions between depths at multiple spatial scales.

5.1 Gaussian Model

Our first model will be a jointly Gaussian Markov Random Field (MRF) as shown in Eq. 1. To capture the multiscale depth relations, we will model the depths $d_i(s)$ for multiple scales $s = 1, 2, 3$. In our experiments, we enforce a hard constraint that depths at a higher scale are the average of the depths at the lower scale.[6] More formally, we define $d_i(s+1) = (1/5)\sum_{j\in N_s(i)\cup\{i\}} d_j(s)$. Here, $N_s(i)$ are the 4 neighbors of patch i at scale s.[7]

In Eq. 1, M is the total number of patches in the image (at the lowest scale); Z is the normalization constant for the model; x_i is the absolute depth feature vector for patch i; and θ and σ are parameters of the model. In detail, we use different parameters (θ_r, σ_{1r}, σ_{2r}) 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. For example, a blue patch might represent sky if it is in upper part of image, and might be more likely to be water if in the lower part of the image.

Our model is a conditionally trained MRF, in that its model of the depths d is always conditioned on the image features X; i.e., it models only $P(d|X)$. We first estimate the parameters θ_r in Eq. 1 by maximizing the conditional log likelihood $\ell(d) = \log P(d|X;\theta_r)$ of the training data. Since the model is a multivariate Gaussian, the maximum likelihood estimate of parameters θ_r is obtained by solving a linear least squares problem.

The first term in the exponent above models depth as a function of multiscale features of a single patch i. The second term in the exponent places a soft "constraint" on the depths to be smooth. If the variance term σ_{2rs}^2 is a fixed constant, the effect of this term is that it tends to smooth depth estimates across nearby patches. However, in practice the dependencies between patches are not the same everywhere, and our expected value for $(d_i - d_j)^2$ may depend on the features of the local patches.

Therefore, to improve accuracy we extend the model to capture the "variance" term σ_{2rs}^2 in the denominator of the second term as a linear function of the patches i and j's relative depth features y_{ijs} (discussed in Section 4.2). We model the variance as $\sigma_{2rs}^2 = u_{rs}^T|y_{ijs}|$. This helps determine which neighboring patches are likely to have similar depths; for example, the "smoothing" effect is much stronger if neighboring patches are

[6] One can instead have soft constraints relating the depths at higher scale to depths at lower scale. One can also envision putting more constraints in the MRF, such as that points lying on a long straight edge in an image should lie on a straight line in the 3-d model, etc.

[7] Our experiments using 8-connected neighbors instead of 4-connected neighbors yielded minor improvements in accuracy at the cost of a much longer inference time.

similar. This idea is applied at multiple scales, so that we learn different σ^2_{2rs} for the different scales s (and rows r of the image). The parameters u_{rs} are learned to fit σ^2_{2rs} to the expected value of $(d_i(s) - d_j(s))^2$, with a constraint that $u_{rs} \geq 0$ (to keep the estimated σ^2_{2rs} non-negative), using a quadratic program (QP).

Similar to our discussion on σ^2_{2rs}, we also learn the variance parameter $\sigma^2_{1r} = v_r^T x_i$ as a linear function of the features. Since the absolute depth features x_i are non-negative, the estimated σ^2_{1r} is also non-negative. The parameters v_r are chosen to fit σ^2_{1r} to the expected value of $(d_i(r) - \theta_r^T x_i)^2$, subject to $v_r \geq 0$. This σ^2_{1r} term gives a measure of the uncertainty in the first term, and depends on the features. This is motivated by the observation that in some cases, depth cannot be reliably estimated from the local features. In this case, one has to rely more on neighboring patches' depths, as modeled by the second term in the exponent.

After learning the parameters, given a new test-set image we can find the MAP estimate of the depths by maximizing Eq. 1 in terms of d. Since Eq. 1 is Gaussian, $\log P(d|X;\theta,\sigma)$ is quadratic in d, and thus its maximum is easily found in closed form (taking at most 1-2 seconds per image). More details are given in Appendix A.

5.2 Laplacian model

We now present a second model (Eq. 2) that uses Laplacians instead of Gaussians to model the posterior distribution of the depths. Our motivation for doing so is three-fold. First, a histogram of the relative depths $(d_i - d_j)$ is empirically closer to Laplacian than Gaussian (Fig. 5, see [59] for more details on depth statistics), which strongly suggests that it is better modeled as one.[8] Second, the Laplacian distribution has heavier tails, and is therefore more robust to outliers in the image features and to errors in the training-set depthmaps (collected with a laser scanner; see Section 6.1). Third, the Gaussian model was generally unable to give depthmaps with sharp edges; in contrast, Laplacians tend to model sharp transitions/outliers better.

This model is parametrized by θ_r (similar to Eq. 1) and by λ_{1r} and λ_{2rs}, the *Laplacian spread* parameters. Maximum-likelihood parameter estimation for the Laplacian model is not tractable (since the partition function depends on θ_r). However, by analogy to the Gaussian case, we approximate this by solving a linear system of equations $X_r\theta_r \approx d_r$ to minimize L_1 (instead of L_2) error, i.e., $\min_{\theta_r} ||d_r - X_r\theta_r||_1$. Here X_r is the matrix of absolute depth features. Following the Gaussian model, we also learn the Laplacian spread parameters in the denominator in the same way, except that the instead of estimating the expected values of $(d_i - d_j)^2$ and $(d_i(r) - \theta_r^T x_i)^2$, we estimate the expected values of $|d_i - d_j|$ and $|d_i(r) - \theta_r^T x_i|$, as a linear function of u_{rs} and v_r respectively. This is done using a Linear Program (LP), with $u_{rs} \geq 0$ and $v_r \geq 0$.

Even though maximum likelihood (ML) parameter estimation for θ_r is intractable in the Laplacian model, given a new test-set image, MAP inference for the depths d is tractable and convex. Details on solving the inference problem as a Linear Program (LP) are given in Appendix B.

Remark. We can also extend these models to combine Gaussian and Laplacian terms in the exponent, for example by using a L_2 norm term for absolute depth, and a L_1 norm term for the interaction terms. MAP inference remains tractable in this setting, and can be solved using convex optimization as a QP (quadratic program).

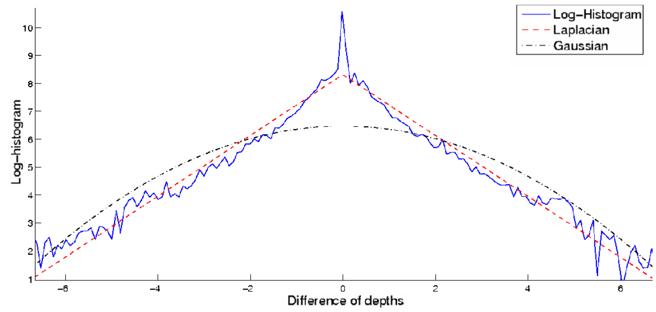

Figure 5: The log-histogram of relative depths. Empirically, the distribution of relative depths is closer to Laplacian than Gaussian.

6 Experiments

6.1 Data collection

We used a 3-d laser scanner to collect images and their corresponding depthmaps (Fig. 7). The scanner uses a laser device (SICK LMS-291) which gives depth readings in a vertical column, with a 1.0° resolution. To collect readings along the other axis (left to right), the SICK laser was mounted on a panning motor. The motor rotates after each vertical scan to collect laser readings for another vertical column, with a 0.5° horizontal angular resolution. We reconstruct the depthmap using the vertical laser scans, the motor readings and known relative position and pose of the laser device and the camera. We also collected data of stereo pairs with corresponding depthmaps (Section 7), by mounting the laser range finding equipment on a LAGR (Learning Applied to Ground Robotics) robot (Fig. 8). The LAGR vehicle is equipped with sensors, an onboard computer, and Point Grey Research Bumblebee stereo cameras, mounted with a baseline distance of 11.7cm. [25]

We collected a total of 425 image+depthmap pairs, with an image resolution of 1704x2272 and a depthmap resolution of 86x107. In the experimental results reported here, 75% of the images/depthmaps were used for training, and the remaining 25% for hold-out testing.

[8] Although the Laplacian distribution fits the log-histogram of multiscale relative depths reasonably well, there is an unmodeled peak near zero. A more recent model [30] attempts to model this peak, which arises due to the fact that the neighboring depths at the finest scale frequently lie on the same object.

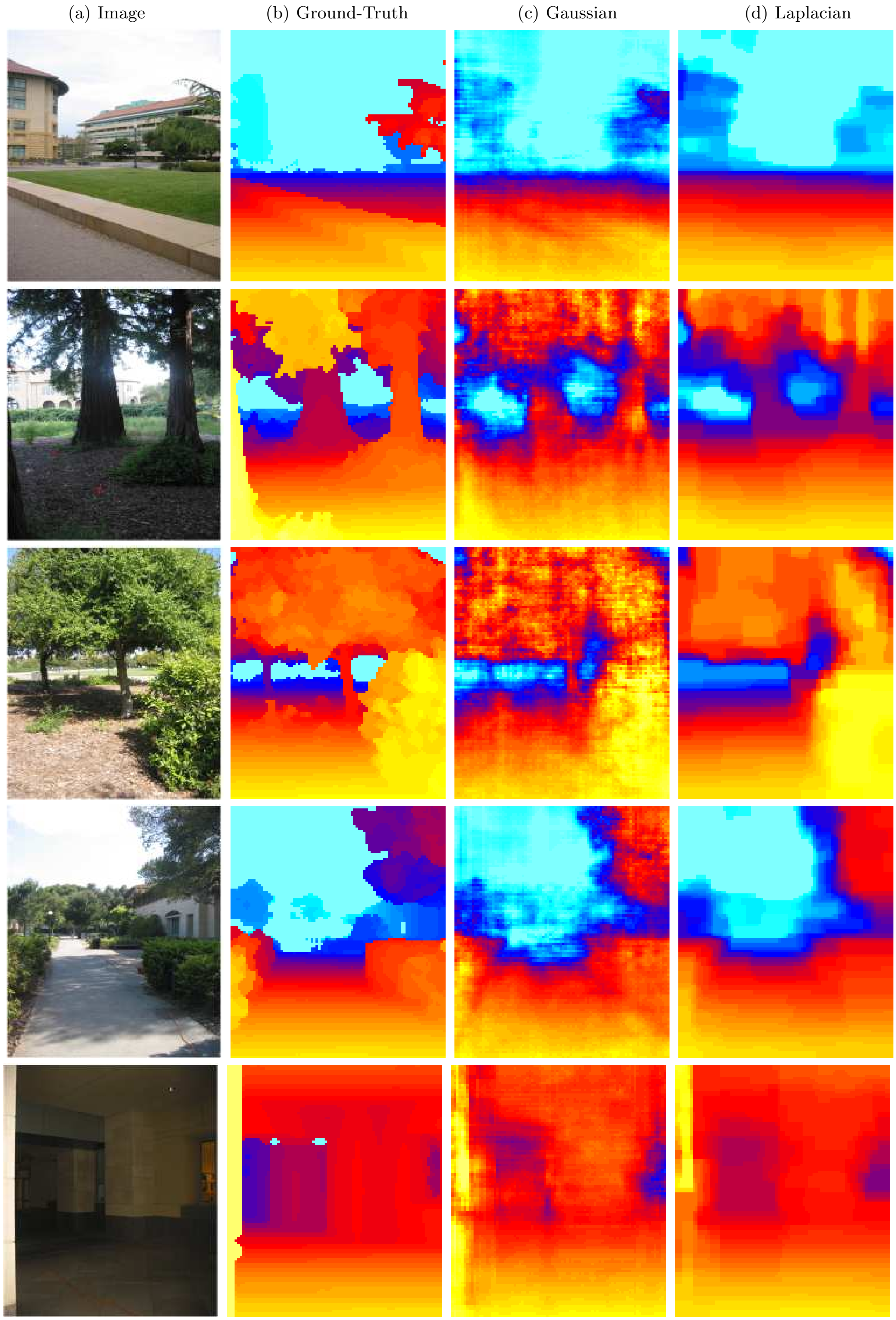

Figure 6: Results for a varied set of environments, showing (a) original image, (b) ground truth depthmap, (c) predicted depthmap by Gaussian model, (d) predicted depthmap by Laplacian model. (**Best viewed in color**).

Figure 7: The 3-d scanner used for collecting images and the corresponding depthmaps.

Figure 8: The custom built 3-d scanner for collecting depthmaps with stereo image pairs, mounted on the LAGR robot.

The images comprise a wide variety of scenes including natural environments (forests, trees, bushes, etc.), man-made environments (buildings, roads, sidewalks, trees, grass, etc.), and purely indoor environments (corridors, etc.). Due to limitations of the laser, the depthmaps had a maximum range of 81m (the maximum range of the laser scanner), and had minor additional errors due to reflections, missing laser scans, and mobile objects. Prior to running our learning algorithms, we transformed all the depths to a log scale so as to emphasize multiplicative rather than additive errors in training. Data used in the experiments is available at: **http://ai.stanford.edu/∼asaxena/learningdepth/**

6.2 Results

We tested our model on real-world test-set images of forests (containing trees, bushes, etc.), campus areas (buildings, people, and trees), and indoor scenes (such as corridors).

Table 1 shows the test-set results with different feature combinations of scales, summary statistics, and neighbors, on three classes of environments: forest, campus, and indoor. The *Baseline* model is trained without any features, and predicts the mean value of depth in the training depthmaps. We see that multiscale and column features improve the algorithm's performance. Including features from neighboring patches, which help capture more global information, reduces the error from .162 orders of magnitude to .133 orders of magnitude.[9] We also note that the Laplacian model performs better than the Gaussian one, reducing error to .084 orders of magnitude for indoor scenes, and .132 orders of magnitude when averaged over all scenes. Empirically, the Laplacian model does indeed give depthmaps with significantly sharper boundaries (as in our discussion in Section 5.2; also see Fig. 6).

Fig. 9 shows that modeling the spatial relationships in the depths is important. Depths estimated without using the second term in the exponent of Eq. 2, i.e., depths predicted using only image features with row-sensitive parameters θ_r, are very noisy (Fig. 9d).[10] Modeling the relations between the neighboring depths at multiple scales through the second term in the exponent of Eq. 2 also gave better depthmaps (Fig. 9e). Finally, Fig. 9c shows the model's "prior" on depths; the depthmap shown reflects our model's use of image-row sensitive parameters. In our experiments, we also found that many features/cues were given large weights; therefore, a model trained with only a few cues (e.g., the top 50 chosen by a feature selection method) was not able to predict reasonable depths.

[9]Errors are on a $\log_{10}$ scale. Thus, an error of ε means a multiplicative error of 10^{ε} in actual depth. E.g., $10^{.132} = 1.355$, which thus represents an 35.5% multiplicative error.

[10]This algorithm gave an overall error of .181, compared to our full model's error of .132.

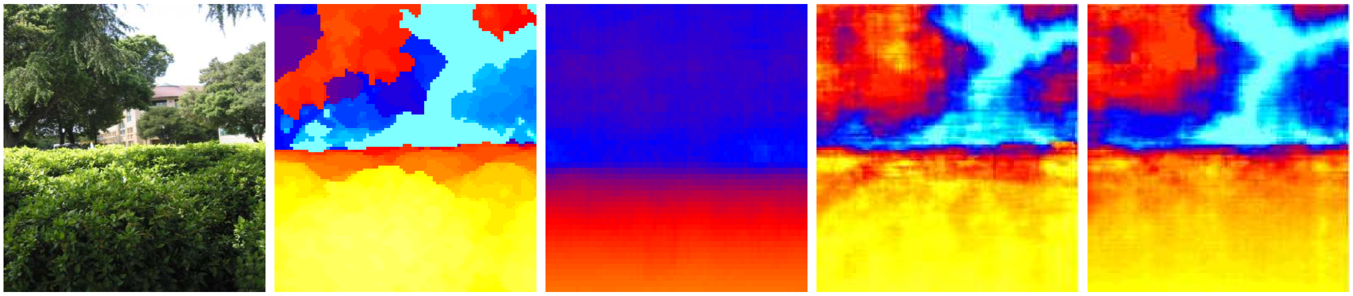

Figure 9: (a) original image, (b) ground truth depthmap, (c) "prior" depthmap (trained with no features), (d) features only (no MRF relations), (e) Full Laplacian model. (**Best viewed in color**).

Table 1: Effect of multiscale and column features on accuracy. The average absolute errors (RMS errors gave very similar trends) are on a log scale (base 10). H_1 and H_2 represent summary statistics for $k = 1, 2$. S_1, S_2 and S_3 represent the 3 scales. C represents the column features. Baseline is trained with only the bias term (no features).

Feature	All	Forest	Campus	Indoor
Baseline	.295	.283	.343	.228
Gaussian (S_1,S_2,S_3, H_1,H_2,no neighbors)	.162	.159	.166	.165
Gaussian (S_1, H_1,H_2)	.171	.164	.189	.173
Gaussian (S_1,S_2, H_1,H_2)	.155	.151	.164	.157
Gaussian (S_1,S_2,S_3, H_1,H_2)	.144	.144	.143	.144
Gaussian (S_1,S_2,S_3, C, H_1)	.139	.140	.141	.122
Gaussian (S_1,S_2,S_3, C, H_1,H_2)	.133	.135	**.132**	.124
Laplacian	**.132**	**.133**	.142	**.084**

Our algorithm works well in a varied set of environments, as shown in Fig. 6 (last column). A number of vision algorithms based on "ground finding" (e.g., [60]) appear to perform poorly when there are discontinuities or significant luminance variations caused by shadows, or when there are significant changes in the ground texture. In contrast, our algorithm appears to be robust to luminance variations, such as shadows (Fig. 6, 4th row) and camera exposure (Fig. 6, 2nd and 5th rows).

Some of the errors of the algorithm can be attributed to errors or limitations of the training set. For example, the maximum value of the depths in the training and test set is 81m; therefore, far-away objects are all mapped to the distance of 81m. Further, laser readings are often incorrect for reflective/transparent objects such as glass; therefore, our algorithm also often estimates depths of such objects incorrectly. Quantitatively, our algorithm appears to incur the largest errors on images which contain very irregular trees, in which most of the 3-d structure in the image is dominated by the shapes of the leaves and branches. However, arguably even human-level performance would be poor on these images.

We note that monocular cues rely on prior knowledge, learned from the training set, about the environment. This is because monocular 3-d reconstruction is an inherently ambiguous problem. Thus, the monocular cues may not generalize well to images very different from ones in the training set, such as underwater images or aerial photos.

To test the generalization capability of the algorithm, we also estimated depthmaps of images downloaded from the Internet (images for which camera parameters are not known).[11] The model (using monocular cues only) was able to produce reasonable depthmaps on most of the images (Fig. 10). Informally, our algorithm appears to predict the relative depths quite well (i.e., their relative distances to the camera);[12] even for scenes very different from the training set, such as a sunflower field, an oil-painting scene, mountains and lakes, a city skyline photographed from sea, a city during snowfall, etc.

Car Driving Experiments: Michels, Saxena and Ng [20] used a simplified version of the monocular depth estimation algorithm to drive a remote-controlled car (Fig. 11a). The algorithm predicts (1-d) depths from single still images, captured from a web-camera with 320x240 pixel resolution. The learning algorithm can be trained on either real camera images labeled with ground-truth ranges to the closest obstacle in each direction, or on a training set consisting of synthetic graphics images. The resulting algorithm, trained on a combination of real and synthetic data, was able to learn monocular visual cues that accurately estimate the relative depths of obstacles in a scene (Fig. 11b). We tested

[11]Since we do not have ground-truth depthmaps for images downloaded from the Internet, we are unable to give a quantitative comparisons on these images. Further, in the extreme case of orthogonal cameras or very wide angle perspective cameras, our algorithm would need to be modified to take into account the field of view of the camera.

[12]For most applications such as object recognition using knowledge of depths, robotic navigation, or 3-d reconstruction, relative depths are sufficient. The depths could be rescaled to give accurate absolute depths, if the camera parameters are known or are estimated.

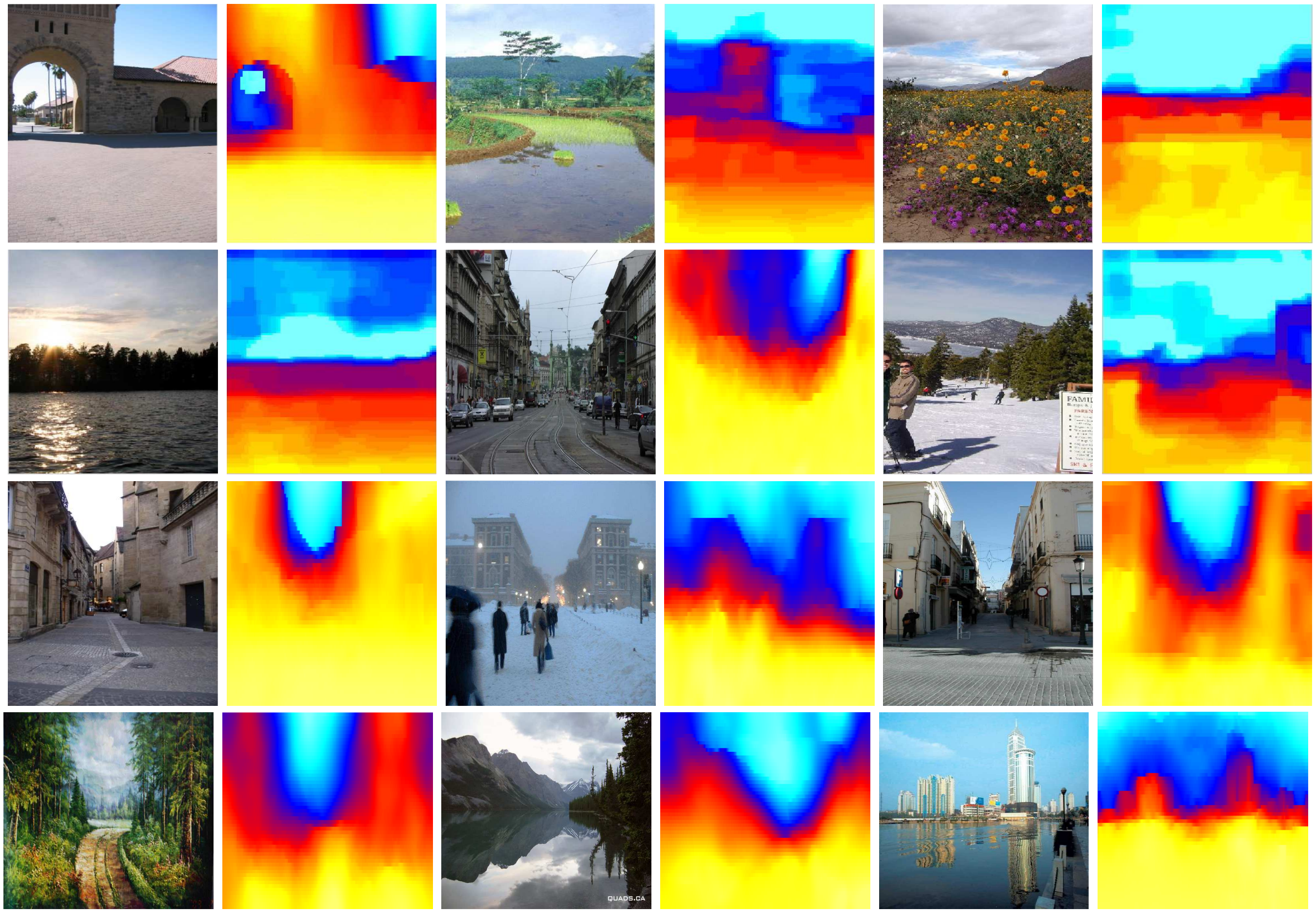

Figure 10: Typical examples of the predicted depthmaps for images downloaded from the internet. (**Best viewed in color.**)

the algorithm by driving the car at four different locations, ranging from man-made environments with concrete tiles and trees, to uneven ground in a forest environment with rocks, trees and bushes where the car is almost never further than 1m from the nearest obstacle. The mean timebefore crash ranged from 19 to more than 200 seconds, depending on the density of the obstacles [20]. The unstructured testing sites were limited to areas where no training or development images were taken. Videos of the algorithm driving the car autonomously are available at:
http://ai.stanford.edu/∼asaxena/rccar

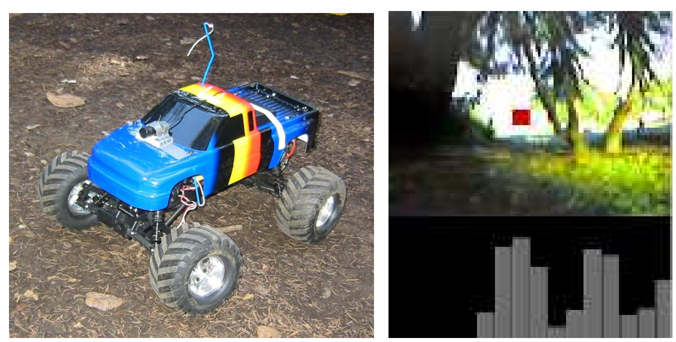

Figure 11: (a) The remote-controlled car driven autonomously in various cluttered unconstrained environments, using our algorithm. (b) A view from the car, with the chosen steering direction indicated by the red square; the estimated distances to obstacles in the different directions are shown by the bar graph below the image.

7 Improving Performance of Stereovision using Monocular Cues

Consider the problem of estimating depth from two images taken from a pair of stereo cameras (Fig. 12). The most common approach for doing so is stereopsis (stereovision), in which depths are estimated by triangulation using the two images. Over the past few decades, researchers have developed very good stereovision systems (see [1] for a review). Although these systems work well

Figure 12: Two images taken from a stereo pair of cameras, and the depthmap calculated by a stereo system.

in many environments, stereovision is fundamentally limited by the baseline distance between the two cameras. Specifically, their depth estimates tend to be inaccurate when the distances considered are large (because even very small triangulation/angle estimation errors translate to very large errors in distances). Further, stereovision also tends to fail for textureless regions of images where correspondences cannot be reliably found.

On the other hand, humans perceive depth by seamlessly combining monocular cues with stereo cues. We believe that monocular cues and (purely geometric) stereo cues give largely orthogonal, and therefore complementary, types of information about depth. Stereo cues are based on the difference between two images and do not depend on the content of the image. Even if the images are entirely random, it would still generate a pattern of disparities (e.g., random dot stereograms [52]). On the other hand, depth estimates from monocular cues are entirely based on the evidence about the environment presented in a single image. In this section, we investigate how monocular cues can be integrated with any reasonable stereo system, to obtain better depth estimates than the stereo system alone.

7.1 Disparity from stereo correspondence

Depth estimation using stereovision from two images (taken from two cameras separated by a baseline distance) involves three steps: First, establish correspondences between the two images. Then, calculate the relative displacements (called "disparity") between the features in each image. Finally, determine the 3-d depth of the feature relative to the cameras, using knowledge of the camera geometry.

Stereo correspondences give reliable estimates of disparity, except when large portions of the image are featureless (i.e., correspondences cannot be found). Further, for a given baseline distance between cameras, the accuracy decreases as the depth values increase. In the limit of very distant objects, there is no observable disparity, and depth estimation generally fails. Empirically, depth estimates from stereo tend to become unreliable when the depth exceeds a certain distance.

Our stereo system finds good feature correspondences between the two images by rejecting pixels with little texture, or where the correspondence is otherwise ambiguous. More formally, we reject any feature where the best match is not significantly better than all other matches within the search window. We use the sum-of-absolute-differences correlation as the metric score to find correspondences [2]. Our cameras (and algorithm) allow sub-pixel interpolation accuracy of 0.2 pixels of disparity. Even though we use a fairly basic implementation of stereopsis, the ideas in this paper can just as readily be applied together with other, perhaps better, stereo systems.

7.2 Modeling Uncertainty in Stereo

The errors in disparity are often modeled as either Gaussian [3] or via some other, heavier-tailed distribution (e.g., [61]). Specifically, the errors in disparity have two main causes: (a) Assuming unique/perfect correspondence, the disparity has a small error due to image noise (including aliasing/pixelization), which is well modeled by a Gaussian. (b) Occasional errors in correspondence cause larger errors, which results in a heavy-tailed distribution for disparity [61].

If the standard deviation is σ_g in computing disparity g from stereo images (because of image noise, etc.), then the standard deviation of the depths[13] will be $\sigma_{d,\mathrm{stereo}} \approx \sigma_g/g$. For our stereo system, we have that σ_g is about 0.2 pixels;[14] this is then used to estimate $\sigma_{d,\mathrm{stereo}}$. Note therefore that $\sigma_{d,\mathrm{stereo}}$ is a function of the estimated depth, and specifically, it captures the fact that variance in depth estimates is larger for distant objects than for closer ones.

7.3 Probabilistic Model

We use our Markov Random Field (MRF) model, which models relations between depths at different points in the image, to incorporate both monocular and stereo cues. Therefore, the depth of a particular patch depends on the monocular features of the patch, on the stereo disparity, and is also related to the depths of other parts of the image.

In our Gaussian and Laplacian MRFs (Eq. 3 and 4), we now have an additional term $d_{i,stereo}$, which is the depth estimate obtained from disparity.[15] This term models the relation between the depth and the estimate from stereo disparity. The other terms in the models are similar to Eq. 1 and 2 in Section 5.

7.4 Results on Stereo

For these experiments, we collected 257 stereo pairs+depthmaps in a wide-variety of outdoor and indoor environments, with an image resolution of 1024x768

[13] Using the delta rule from statistics: $\mathrm{Var}(f(x)) \approx (f'(x))^2\mathrm{Var}(x)$, derived from a second order Taylor series approximation of $f(x)$. The depth d is related to disparity g as $d = \log(C/g)$, with camera parameters determining C.

[14] One can also envisage obtaining a better estimate of σ_g as a function of a match metric used during stereo correspondence [62], such as normalized sum of squared differences; or learning σ_g as a function of disparity/texture based features.

[15] In this work, we directly use $d_{i,stereo}$ as the stereo cue. In [63], we use a library of features created from stereo depths as the cues for identifying a grasp point on objects.

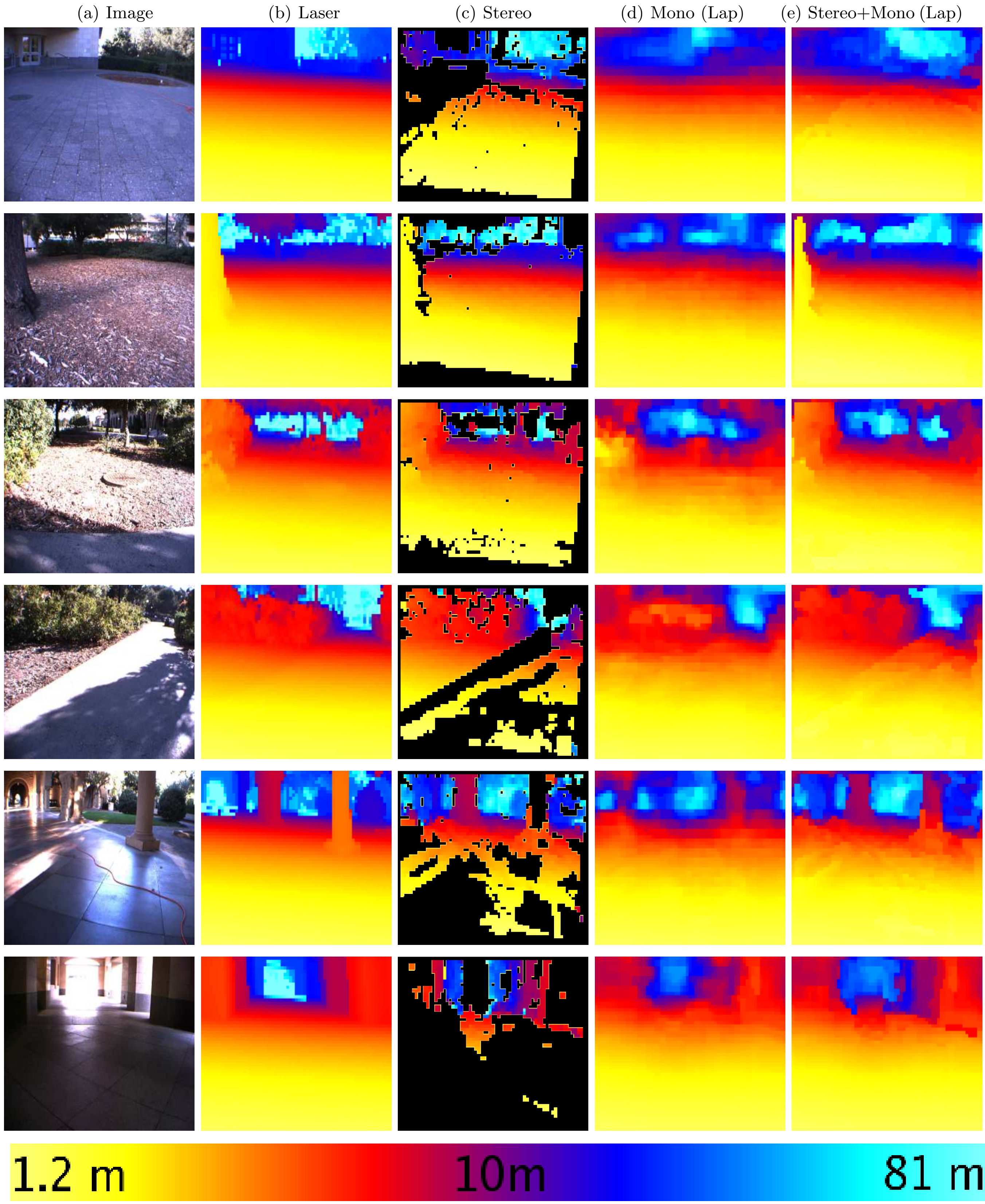

Figure 13: Results for a varied set of environments, showing one image of the stereo pairs (column 1), ground truth depthmap collected from 3-d laser scanner (column 2), depths calculated by stereo (column 3), depths predicted by using monocular cues only (column 4), depths predicted by using both monocular and stereo cues (column 5). The bottom row shows the color scale for representation of depths. Closest points are 1.2 m, and farthest are 81m. (**Best viewed in color**)

$$P_G(d|X;\theta,\sigma) = \frac{1}{Z_G}\exp\left(-\frac{1}{2}\sum_{i=1}^{M}\left(\frac{(d_i(1)-d_{i,\mathrm{stereo}})^2}{\sigma_{i,\mathrm{stereo}}^2} + \frac{(d_i(1)-x_i^T\theta_r)^2}{\sigma_{1r}^2} + \sum_{s=1}^{3}\sum_{j\in N_s(i)}\frac{(d_i(s)-d_j(s))^2}{\sigma_{2rs}^2}\right)\right) \quad (3)$$

$$P_L(d|X;\theta,\lambda) = \frac{1}{Z_L}\exp\left(-\sum_{i=1}^{M}\left(\frac{|d_i(1)-d_{i,\mathrm{stereo}}|}{\lambda_{i,\mathrm{stereo}}} + \frac{|d_i(1)-x_i^T\theta_r|}{\lambda_{1r}} + \sum_{s=1}^{3}\sum_{j\in N_s(i)}\frac{|d_i(s)-d_j(s)|}{\lambda_{2rs}}\right)\right) \quad (4)$$

Table 2: The average errors (RMS errors gave very similar trends) for various cues and models, on a log scale (base 10).

Algorithm	All	Campus	Forest	Indoor
Baseline	.341	.351	.344	.307
Stereo	.138	.143	.113	.182
Stereo (smooth)	.088	.091	.080	.099
Mono (Gaussian)	.093	.095	.085	.108
Mono (Lap)	.090	.091	.082	.105
Stereo+Mono (Lap)	**.074**	**.077**	**.069**	**.079**

and a depthmap resolution of 67x54. We used 75% of the images/depthmaps for training, and the remaining 25% for hold-out testing.

We quantitatively compare the following classes of algorithms that use monocular and stereo cues in different ways:
(i) **Baseline**: This model, trained without any features, predicts the mean value of depth in the training depthmaps.
(ii) **Stereo**: Raw stereo depth estimates, with the missing values set to the mean value of depth in the training depthmaps.
(iii) **Stereo (smooth)**: This method performs interpolation and region filling; using the Laplacian model without the second term in the exponent in Eq. 4, and also without using monocular cues to estimate λ_2 as a function of the image.
(iv) **Mono (Gaussian)**: Depth estimates using only monocular cues, without the first term in the exponent of the Gaussian model in Eq. 3.
(v) **Mono (Lap)**: Depth estimates using only monocular cues, without the first term in the exponent of the Laplacian model in Eq. 4.
(vi) **Stereo+Mono**: Depth estimates using the full model.

Table 2 shows that although the model is able to predict depths using monocular cues only ("Mono"), the performance is significantly improved when we combine both mono and stereo cues. The algorithm is able to estimate depths with an error of .074 orders of magnitude, (i.e., 18.6% of multiplicative error because $10^{.074} = 1.186$) which represents a significant improvement over stereo (smooth) performance of .088.

Fig. 13 shows that the model is able to predict depthmaps (column 5) in a variety of environments. It also demonstrates how the model takes the best estimates from both stereo and monocular cues to estimate more accurate depthmaps. For example, in row 6 (Fig. 13), the depthmap generated by stereo (column 3) is very inaccurate; however, the monocular-only model predicts depths fairly accurately (column 4). The combined model uses both sets of cues to produce a better depthmap (column 5). In row 3, stereo cues give a better estimate than monocular ones. We again see that our combined MRF model, which uses both monocular and stereo cues, gives an accurate depthmap (column 5) correcting some mistakes of stereo, such as some far-away regions that stereo predicted as close.

In Fig. 14, we study the behavior of the algorithm as a function of the 3-d distance from the camera. At small distances, the algorithm relies more on stereo cues, which are more accurate than the monocular cues in this regime. However, at larger distances, the performance of stereo degrades, and the algorithm relies more on monocular cues. Since our algorithm models uncertainties in both stereo and monocular cues, it is able to combine stereo and monocular cues effectively.

We note that monocular cues rely on prior knowledge, learned from the training set, about the environment. This is because monocular 3-d reconstruction is an inherently ambiguous problem. In contrast, the stereopsis cues we used are are purely geometric, and therefore should work well even on images taken from very different environments. For example, the monocular algorithm fails sometimes to predict correct depths for objects which are only partially visible in the image (e.g., Fig. 13, row 2: tree on the left). For a point lying on such an object, most of the point's neighbors lie outside the image; hence the relations between neighboring depths are less effective here than for objects lying in the middle of an image. However, in many of these cases, the stereo cues still allow an accurate depthmap to be estimated (row 2, column 5).

8 Conclusions

Over the last few decades, stereo and other "triangulation" cues have been successfully applied to many important problems, including robot navigation, building 3-d models of urban environments, and object recognition. Unlike triangulation-based algorithms such as stereopsis and structure from motion, we have developed a class of algorithms that exploit a largely orthogonal set of monocular cues. We presented a hierarchical, multiscale Markov Random Field (MRF) learning model that uses such cues to estimate depth from a single still image. These monocular cues can not only be combined with triangulation ones, but also scale better than most

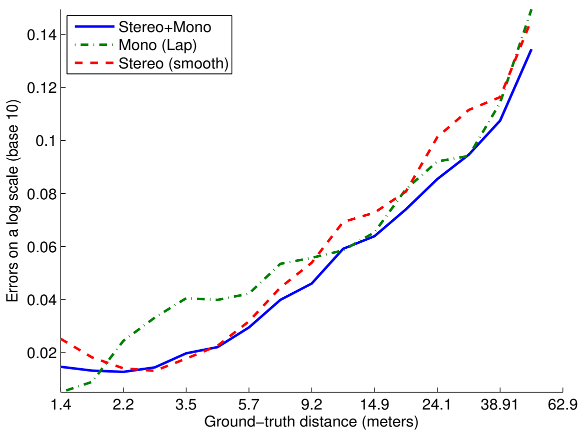

Figure 14: The average errors (on a log scale, base 10) as a function of the distance from the camera.

triangulation-based cues to depth estimation at large distances. Although our work has been limited to depth estimation, we believe that these monocular depth and shape cues also hold rich promise for many other applications in vision.

Acknowledgments

We give warm thanks to Jamie Schulte, who designed and built the 3-d scanner, and to Andrew Lookingbill, who helped us with collecting the data used in this work. We also thank Jeff Michels, Larry Jackel, Sebastian Thrun, Min Sun and Pieter Abbeel for helpful discussions. This work was supported by the DARPA LAGR program under contract number FA8650-04-C-7134.

Appendix

A MAP Inference for Gaussian Model

We can rewrite Eq. 1 as a standard multivariate Gaussian,

$$P_G(d|X;\theta,\sigma) = \frac{1}{Z_G}\exp\left(-\frac{1}{2}(d - X_a\theta_r)^T\Sigma_a^{-1}(d - X_a\theta_r)\right)$$

where $X_a = (\Sigma_1^{-1} + Q^T\Sigma_2^{-1}Q)^{-1}\Sigma_1^{-1}X$, with Σ_1 and Σ_2 representing the matrices of the variances $\sigma_{1,i}^2$ and $\sigma_{2,i}^2$ in the first and second terms in the exponent of Eq. 1 respectively.[16] Q is a matrix such that rows of Qd give the differences of the depths in the neighboring patches at multiple scales (as in the second term in the exponent of Eq. 1). Our MAP estimate of the depths is, therefore, $d^* = X_a\theta_r$.

During learning, we iterate between learning θ and estimating σ. Empirically, $\sigma_1 \ll \sigma_2$, and X_a is very close to X; therefore, the algorithm converges after 2-3 iterations.

[16]Note that if the variances at each point in the image are constant, then $X_a = (I + \sigma_1^2/\sigma_2^2 Q^TQ)^{-1}X$. I.e., X_a is essentially a smoothed version of X.

B MAP Inference for Laplacian Model

Exact MAP inference of the depths $d \in \mathbb{R}^M$ can be obtained by maximizing $\log P(d|X;\theta,\lambda)$ in terms of d (Eq. 2). More formally,

$$\begin{aligned} d^* &= \arg\max_d \log P(d|X;\theta,\lambda) \\ &= \arg\min_d c_1^T|d - X\theta_r| + c_2^T|Qd| \end{aligned}$$

where, $c_1 \in \mathbb{R}^M$ with $c_{1,i} = 1/\lambda_{1,i}$, and $c_2 \in \mathbb{R}^{6M}$ with $c_{2,i} = 1/\lambda_{2,i}$. Our features are given by $X \in \mathbb{R}^{M\text{x}k}$ and the learned parameters are $\theta_r \in \mathbb{R}^k$, which give a naive estimate $\tilde{d} = X\theta_r \in \mathbb{R}^M$ of the depths. Q is a matrix such that rows of Qd give the differences of the depths in the neighboring patches at multiple scales (as in the second term in the exponent of Eq. 2).

We add auxiliary variables ξ_1 and ξ_2 to pose the problem as a Linear Program (LP):

$$\begin{aligned} d^* = \arg\min_{d,\xi_1,\xi_2} \quad & c_1^T\xi_1 + c_2^T\xi_2 \\ \text{s.t.} \quad & -\xi_1 \le d - \tilde{d} \le \xi_1 \\ & -\xi_2 \le Qd \le \xi_2 \end{aligned}$$

In our experiments, MAP inference takes about 7-8 seconds for an image.

References

[1] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision (IJCV)*, 47(1):7–42, 2002.

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

[3] S. Das and N. Ahuja. Performance analysis of stereo, vergence, and focus as depth cues for active vision. *IEEE Trans Pattern Analysis and Machine Intelligence (PAMI)*, 17(12):1213–1219, 1995.

[4] M. Quartulli and M. Datcu. Bayesian model based city reconstruction from high resolution ISAR data. In *IEEE/ISPRS Joint Workshop Remote Sensing and Data Fusion over Urban Areas*, 2001.

[5] N. Cornelis, B. Leibe, K. Cornelis, and L. Van Gool. 3d city modeling using cognitive loops. In *Video Proceedings of CVPR (VPCVPR)*, 2006.

[6] J.L. Barron, D.J. Fleet, and S.S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision (IJCV)*, 12:43–77, 1994.

[7] C. Frueh and A. Zakhor. Constructing 3D city models by merging ground-based and airborne views. In *Computer Vision and Pattern Recognition (CVPR)*, 2003.

[8] S. Thrun and B. Wegbreit. Shape from symmetry. In *International Conf on Computer Vision (ICCV)*, 2005.

[9] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. SCAPE: shape completion and animation of people. *ACM Trans. Graph.*, 24(3):408–416, 2005.

[10] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In *Computer Vision and Pattern Recognition (CVPR)*, 2003.

[11] T. Nagai, T. Naruse, M. Ikehara, and A. Kurematsu. Hmm-based surface reconstruction from single images. In *IEEE Int'l Conf Image Processing (ICIP)*, 2002.

[12] R. Zhang, P.-S. Tsai, J.E. Cryer, and M. Shah. Shape from shading: A survey. *IEEE Trans on Pattern Analysis and Machine Intelligence (PAMI)*, 21(8):690–706, 1999.

[13] A. Maki, M. Watanabe, and C. Wiles. Geotensity: Combining motion and lighting for 3d surface reconstruction. *International Journal of Computer Vision (IJCV)*, 48(2):75–90, 2002.

[14] T. Lindeberg and J. Garding. Shape from texture from a multi-scale perspective. In *International Conference on Computer Vision (ICCV)*, 1993.

[15] J. Malik and R. Rosenholtz. Computing local surface orientation and shape from texture for curved surfaces. *International Journal of Computer Vision (IJCV)*, 23(2):149–168, 1997.

[16] J. Malik and P. Perona. Preattentive texture discrimination with early vision mechanisms. *Journal of the Optical Society of America A*, 7(5):923–932, 1990.

[17] A. Hertzmann and S.M. Seitz. Example-based photometric stereo: Shape reconstruction with general, varying brdfs. *IEEE Trans Pattern Analysis and Machine Intelligence (PAMI)*, 27(8):1254–1264, 2005.

[18] L. Torresani and A. Hertzmann. Automatic non-rigid 3D modeling from video. In *European Conference on Computer Vision (ECCV)*, 2004.

[19] A. Torralba and A. Oliva. Depth estimation from image structure. *IEEE Trans Pattern Analysis and Machine Intelligence (PAMI)*, 24(9):1–13, 2002.

[20] J. Michels, A. Saxena, and A.Y. Ng. High speed obstacle avoidance using monocular vision and reinforcement learning. In *22nd International Conference on Machine Learning (ICML)*, 2005.

[21] E. Delage, H. Lee, and A.Y. Ng. A dynamic bayesian network model for autonomous 3D reconstruction from a single indoor image. In *Computer Vision and Pattern Recognition (CVPR)*, 2006.

[22] E. Delage, H. Lee, and A.Y. Ng. Automatic single-image 3d reconstructions of indoor manhattan world scenes. In *12th International Symposium of Robotics Research (ISRR)*, 2005.

[23] A. Criminisi, I. Reid, and A. Zisserman. Single view metrology. *International Journal of Computer Vision (IJCV)*, 40:123–148, 2000.

[24] A. Saxena, S.H. Chung, and A.Y. Ng. Learning depth from single monocular images. In *Neural Information Processing System (NIPS) 18*, 2005.

[25] A. Saxena, J. Schulte, and A.Y. Ng. Depth estimation using monocular and stereo cues. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2007.

[26] D. Hoiem, A.A. Efros, and M. Herbert. Geometric context from a single image. In *International Conference on Computer Vision (ICCV)*, 2005.

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

[28] D. Hoiem, A.A. Efros, and M. Herbert. Putting objects in perspective. In *Computer Vision and Pattern Recognition (CVPR)*, 2006.

[29] E.B. Sudderth, A. Torralba, W.T. Freeman, and A.S. Willisky. Depth from familiar objects: A hierarchical model for 3D scenes. In *Computer Vision and Pattern Recognition (CVPR)*, 2006.

[30] A. Saxena, M. Sun, R. Agarwal, and A.Y. Ng. Learning 3-d scene structure from a single still image. Unpublished manuscript, 2006.

[31] O. Nestares, R. Navarro, J. Portilia, and A. Tabernero. Efficient spatial-domain implementation of a multiscale image representation based on gabor functions. *Journal of Electronic Imaging*, 7(1):166–173, 1998.

[32] G. Strang and T. Nguyen. *Wavelets and filter banks*. Wellesley-Cambridge Press, Wellesley, MA, 1997.

[33] E.N. Mortensen, H. Deng, and L. Shapiro. A SIFT descriptor with global context. In *Computer Vision and Pattern Recognition (CVPR)*. 2005.

[34] K. Murphy, A. Torralba, and W.T. Freeman. Using the forest to see the trees: A graphical model relating features, objects, and scenes. In *Neural Information Processing Systems (NIPS) 16*, 2003.

[35] T. Serre, L. Wolf, and T. Poggio. Object recognition with features inspired by visual cortex. In *Computer Vision and Pattern Recognition (CVPR)*, 2005.

[36] W. Zhao, R. Chellappa, P.J. Phillips, and A. Rosenfield. Face recognition: A literature survey. *ACM Computing Surveys*, 35:399–458, 2003.

[37] A. Saxena, A. Anand, and A. Mukerjee. Robust facial expression recognition using spatially localized geometric model. In *International Conf Systemics, Cybernetics and Informatics (ICSCI)*, 2004.

[38] A. Saxena, J. Driemeyer, J. Kearns, C. Osondu, and A.Y. Ng. Learning to grasp novel objects using vision. In *10th International Symposium on Experimental Robotics (ISER)*, 2006.

[39] S. Konishi and A. Yuille. Statistical cues for domain specific image segmentation with performance analysis. In *Computer Vision and Pattern Recognition (CVPR)*, 2000.

[40] A. Oliva and A. Torralba. Building the gist of a scene: the role of global image features in recognition. *IEEE Trans Pattern Analysis and Machine Intelligence (PAMI)*, 155:23–36, 2006.

[41] B.A. Olshausen and D.J. Field. Sparse coding with an over-complete basis set: A strategy employed by v1? *Vision Research*, 37:3311–3325, 1997.

[42] V. Kolmogorov, A. Criminisi, A. Blake, G. Cross, and C. Rother. Probabilistic fusion of stereo with color and contrast for bilayer segmentation. *IEEE Pattern Analysis and Machine Intelligence (PAMI)*, 28(9):1480–1492, 2006.

[43] T.M. Moldovan, S. Roth, and M.J. Black. Denoising archival films using a learned bayesian model. In *International Conference on Image Processing (ICIP)*, 2006.

[44] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning (ICML)*, 2001.

[45] X. He, R. Zemel, and M. Perpinan. Multiscale conditional random fields for image labeling. In *Computer Vision and Pattern Recognition (CVPR)*, 2004.

[46] S. Kumar and M. Hebert. Discriminative fields for modeling spatial dependencies in natural images. In *Neural Information Processing Systems (NIPS) 16*, 2003.

[47] J.M. Loomis. Looking down is looking up. *Nature News and Views*, 414:155–156, 2001.

[48] S.H. Schwartz. *Visual Perception 2nd ed.* Appleton and Lange, Connecticut, 1999.

[49] A.E. Welchman, A. Deubelius, V. Conrad, H.H. Blthoff, and Z. Kourtzi. 3D shape perception from combined depth cues in human visual cortex. *Nature Neuroscience*, 8:820–827, 2005.

[50] J. Porrill, J.P. Frisby, W.J. Adams, and D. Buckley. Robust and optimal use of information in stereo vision. *Nature*, 397:63–66, 1999.

[51] B. Wu, T.L. Ooi, and Z.J. He. Perceiving distance accurately by a directional process of integrating ground information. *Letters to Nature*, 428:73–77, 2004.

[52] I. Bulthoff, H. Bulthoff, and P. Sinha. Top-down influences on stereoscopic depth-perception. *Nature Neuroscience*, 1:254–257, 1998.

[53] S.G. Narasimhan and S.K. Nayar. Shedding light on the weather. In *Computer Vision and Pattern Recognition (CVPR)*, 2003.

[54] Brian A. Wandell. *Foundations of Vision*. Sinauer Associates, Sunderland, MA, 1995.

[55] M. Wexler, F. Panerai, I. Lamouret, and J. Droulez. Self-motion and the perception of stationary objects. *Nature*, 409:85–88, 2001.

[56] L. Harkness. Chameleons use accommodation cues to judge distance. *Nature*, 267:346–349, 1977.

[57] E.R. Davies. Laws' texture energy in TEXTURE. In *Machine Vision: Theory, Algorithms, Practicalities 2nd Edition*. Academic Press, San Diego, 1997.

[58] A.S. Willsky. Multiresolution markov models for signal and image processing. *Proc IEEE*, 90(8):1396–1458, 2002.

[59] J. Huang, A.B. Lee, and D. Mumford. Statistics of range images. *Computer Vision and Pattern Recognition (CVPR)*, 2000.

[60] G. Gini and A. Marchi. Indoor robot navigation with single camera vision. In *PRIS*, 2002.

[61] R. Szeliski. Bayesian modeling of uncertainty in low-level vision. In *International Conference on Computer Vision (ICCV)*, 1990.

[62] M.Z. Brown, D. Burschka, and G.D. Hager. Advances in computational stereo. *IEEE Trans on Pattern Analysis and Machine Intelligence (PAMI)*, 25(8):993–1008, 2003.

[63] A. Saxena, J. Driemeyer, J. Kearns, and A.Y. Ng. Robotic grasping of novel objects. In *Neural Information Processing Systems (NIPS) 19*, 2006.