3-D Reconstruction from Sparse Views using Monocular Vision

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 task of creating a 3-d model of a large novel environment, given only a small number of images of the scene. This is a difficult problem, because if the images are taken from very different viewpoints or if they contain similar-looking structures, then most geometric reconstruction methods will have great difficulty finding good correspondences. Further, the reconstructions given by most algorithms include only points in 3-d that were observed in two or more images; a point observed only in a single image would not be reconstructed. In this paper, we show how monocular image cues can be combined with triangulation cues to build a photo-realistic model of a scene given only a few images—even ones taken from very different viewpoints or with little overlap. Our approach begins by oversegmenting each image into small patches (superpixels). It then simultaneously tries to infer the 3-d position and orientation of every superpixel in every image. This is done using a Markov Random Field (MRF) which simultaneously reasons about monocular cues and about the relations between multiple image patches, both within the same image and across different images (triangulation cues). MAP inference in our model is efficiently approximated using a series of linear programs, and our algorithm scales well to a large number of images.

1. Introduction

We consider the task of creating 3-d models of large novel environments, given only a few images of each scene. Most prior work has focused on using triangulation (geometric) cues for this task, e.g. [1–3], and monocular (single image) cues have been exploited poorly. However, [4–7] showed that even a single image has many cues which can be exploited to obtain rich 3-d information. Even so, a 3-d model built from a single image will almost invariably be an incomplete model of the scene, because many portions of the scene will be missing or occluded. In this paper, we will use both the monocular cues and multi-view triangulation cues to create better and larger 3-d models.

Given a sparse set of images of a scene, it is sometimes possible to construct a 3-d model, using techniques such as structure from motion (SFM) [2, 3], which start by taking two or more photographs, then find correspondences between the images, and finally uses triangulation to obtain 3-d locations of the points. If the images are taken from nearby cameras (i.e., if the baseline distance is small), then these methods often suffer from large triangulation errors for points far-away from the camera.[1] If, conversely, one chooses images taken far apart, then often the change of viewpoint causes the images to become very different, so that finding correspondences becomes extremely difficult, leading either to spurious or missed correspondences. (Worse, the large baseline also means that there may be little overlap between the images, so that few correspondences may even exist.) These difficulties make purely geometric 3-d reconstruction algorithms work unreliably in practice, when given only a small set of images. When tens of thousands of pictures are available—for example, for frequently-photographed tourist attractions such as national monuments—one can discard images that have only few correspondence matches. Doing so, one can use only a small subset of the images available ($\sim$15%), and still obtain a "3-d point cloud" for points that were matched using SFM. This approach has been very successfully applied to a few famous buildings such as the Notre Dame; the computational cost of this algorithm was significant, and required about a week on a cluster of computers [8]. Other 3-d reconstruction methods include using other types of sensing hardware, such as lasers [9] or calibrated stereo video streams [10], to create a dense colored 3-d point cloud. This paper address the much harder problem of 3-d reconstruction given only a small set of images.

The reason that many geometric "triangulation-based" methods fail is that they do not make use of the information present in a single image. Saxena et al. [4, 5], Delage et al. [11] and Hoiem et al. [7] showed that given even a single image, we can automatically infer a significant portion of the scene's 3-d structure. Building on these ideas, we will develop an MRF model that seamlessly combines triangulation cues and monocular image cues, while also reasoning about 3-d properties of the world such as occlusion, to build a full photo-realistic 3-d model of a scene. We also describe how single-image monocular cues can be used to make 3-d fea-

[1] I.e., the depth estimates will tend to be inaccurate for objects at large distances, because even small errors in triangulation will result in large errors in depth.

ture matching (for triangulation) significantly more robust.

To the best of our knowledge, our work represents the first algorithm capable of automatically creating a full photo-realistic 3-d model from a sparse set of images. Using our approach, we were able to create 3-d models of several large environments.

2. Prior Work

The last few decades have seen a significant amount of work in stereovision and structure from motion. Space constraints prevent us from doing justice to this literature, but examples include [3, 8], and [1, 2] provide detailed surveys.

In some specific settings, there has been some work on depth estimation from single images. Criminisi et al. [12] provided an interactive method for computing 3-d geometry, where the user can specify the object segmentation, 3-d coordinates of some points, and reference height of an object. Methods such as shape-from-shading [13] assume uniform texture on surfaces to estimate depths. Torralba and Oliva [14] studied the relationship between the Fourier spectrum of an image and its mean depth.

In recent work, Saxena, Chung and Ng (SCN) [4, 15] presented an algorithm for predicting depth from monocular image features, and applied it to tasks such as robot driving [16]. Delage, Lee and Ng (DLN) [6, 11] and Hoiem, Efros and Hebert (HEH) [7, 17] 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 horizontal/ground and vertical regions (also sky in the case of HEH) to produce a simple "pop-up" type fly-through from an image. HEH focused on creating "visually-pleasing" fly-throughs, but did not produce quantitatively accurate results.

In [5], Saxena, Sun and Ng presented an algorithm that uses a Markov Random Field to infer both the 3-d location and the 3-d orientation of small patches in the image. They produced models that capture rich and detailed 3-d structure, and that are both visually-pleasing and quantitatively correct. In detail, they tested their approach on 588 images downloaded from the internet, and were able to produce qualitatively correct models for 64.9% of them. These methods for producing 3-d models from a single picture give only incomplete models, because many parts of the scene are almost invariably missing or occluded.

Several methods have been used to improve the performance of purely triangulation based methods; examples include local smoothing in 2-d, or using image segmentation information to improve stereo depth maps [1]. Priors on camera locations were exploited in [18] to obtain more robust matching and triangulation. Recently, monocular depth cues were used in [19] to improve performance of stereovision algorithms. However, most multi-view reconstruction algorithms for producing large scale 3-d models ignore monocular information.

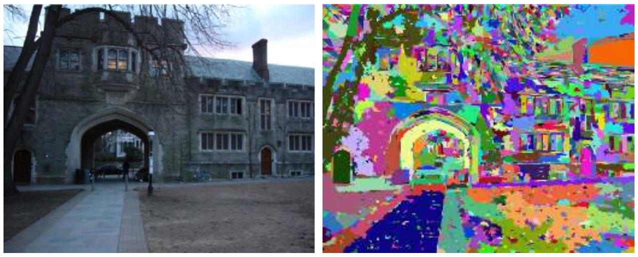


Figure 1. (a) An image of a scene. (c) Over-segmented image. Each small segment (superpixel) lies on a plane in the 3d scene.

3. Visual Cues for Scene Understanding

Humans understand the 3-d structure of a scene by "integrating information" available from different sources [20]. From a single image, they use a variety of monocular cues, such as texture variations and gradients, color, haze, defocus, etc. [4,21] From the stereo pair of images from the eyes, they use stereo (triangulation) cues to estimate depth [19, 22]. They also capture multiple views of the scene by moving their head to different places to build a consistent 3-d structure of the world around them.

Humans can infer 3-d structure even when only a single view is available of parts of a scene, by using their prior experience. Although an image might represent an infinite number of possible 3-d structures because of projective ambiguity, the environment that we live in is reasonably structured; thus out of all the possible 3-d structures that an image might represent, only a few are likely. This allows a human to infer 3-d structure by learning the relations between the image features and depth, and the relations between different parts of the scene. [23, chap. 11]

4. Representation

Our goal is to create a full photo-realistic 3-d model from an image. Following most work on 3-d models in computer graphics and other related fields, we will use a polygonal mesh representation of the 3-d model, in which we assume the world is made of a set of small planes.[2] In detail, given an image of the scene, we first find small homogeneous regions in the image, called "Superpixels" [24]. Such regions represent a coherent region in the scene with all the pixels having similar properties. Our basic unit of representation will be these small planes in the world, and our goal is to infer the location and orientation of each of these small planes.

More formally, we parametrize both the 3-d location and orientation of the infinite plane on which a superpixel lies by using a set of plane parameters $\alpha \in \mathbb{R}^3$. (Fig. 2) (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

[2] This assumption is reasonably accurate for most artificial structures, such as buildings. Some natural structures such as trees could perhaps be better represented by a cylinder. However, since our models are quite detailed, e.g., about 2500 planes for a small scene, the planar assumption works quite well in practice.

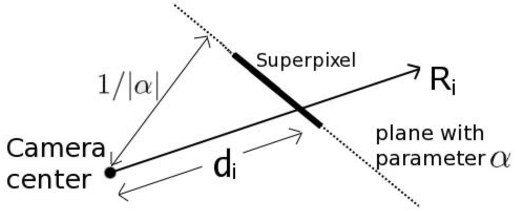


Figure 2. A 2-d illustration to explain the plane parameter α and rays R from the camera.

$\hat{\alpha} = \frac{\alpha}{|\alpha|}$ 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. (See [5] for more details.)

Given two small plane (superpixel) segmentations of two images, there is no guarantee that the two segmentations are "consistent," in the sense of the small planes (on a specific object) in one image having a one-to-one correspondence to the planes in the second image of the same object. Thus, at first blush it appears non-trivial to build a 3-d model using these segmentations, since it is impossible to associate the planes in one image to those in another. We address this problem by using our MRF to reason simultaneously about the position and orientation of every plane in every image. If two planes lie on the same object, then the MRF will (hopefully) infer that they have exactly the same 3-d position. More formally, in our model, the plane parameters α_i^n of each small i^{th} plane in the n^{th} image are represented by a node in our Markov Random Field (MRF). Because our model uses L_1 penalty terms (see Section 5 for details), our algorithm will be able to infer models for which $\alpha_i^n = \alpha_j^m$, which results in the two planes exactly overlapping each other.

5. Probabilistic Model

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

- **Image Features and depth**: The image features bear some relation to the depth/orientation of the superpixel.
- **Co-linearity**: Long straight lines in the image represent straight lines in the 3-d model. For example, edges of buildings, sidewalk, windows.
- **Connected structure and Co-planarity**: Neighboring superpixels are more likely to be connected and coplanar if they look similar.
- **Correspondences across images**: Two points in two images are more likely to occupy the same physical location in the 3-d scene if they look very similar.
- **Depths from Triangulation**: If an estimate of the depth at a point is available from triangulation (SFM), then the depth of the point is more likely to be close to the estimated depth.

The first three of these properties were used in [5] for single image 3-d reconstruction. Note that no single one of these 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.

Fractional depth error: For 3-d reconstruction, the fractional (or relative) error in depths is perhaps the most meaningful performance metric, because the 3-d structure of an object is better predicted by an image taken closer to the object than one taken far-away. Fractional error is a popular metric used in structure for motion, stereo reconstruction, etc. [1, 25] For ground-truth depth d and estimated depth $\hat{d}$, fractional error is defined as $(\hat{d} - d)/d = \hat{d}/d - 1$. In our model, we will be penalizing fractional errors in the distances.

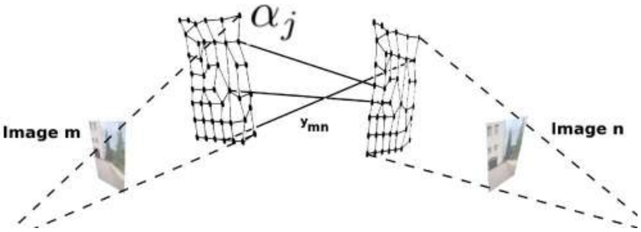


Figure 3. An illustration of the Markov Random Field (MRF) for inferring 3-d structure. (Only a subset of edges shown.)

5.1. Plane Parameter MRF

In our 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.

Let Ψ be the set of pairs of images which have correspondence matches available, let $Q^n = [\text{Rotation}, \quad \text{Translation}] \in \mathbb{R}^{3\times4}$ (technically SE(3)) be the camera pose when image n was taken (w.r.t. a fixed reference, such as the camera pose of the first image), and let d_T be the depths obtained by triangulation (see Section 5.2). We formulate our MRF as

$$P(\alpha|X, Y, d_T; \theta) \propto \prod_n P_1(\alpha^n|X^n, Y^n, Q^n; \theta^n)$$
$$\prod_n P_2(\alpha^n|X^n, Y^n, Q^n)$$
$$\prod_{n,m\in\Psi} P_3(\alpha^n, \alpha^m|Q^n, Q^m, Y^{mn})$$
$$\prod_n P_4(\alpha^n|Q^n, d_T^n, Y_T^n)$$

where, the superscript n is an index over the images, For an image n, α_i^n is the plane parameter of superpixel i in image n. Sometimes, we will drop the superscript for brevity, and write α in place of α^n when it is clear that we are referring to a particular image.

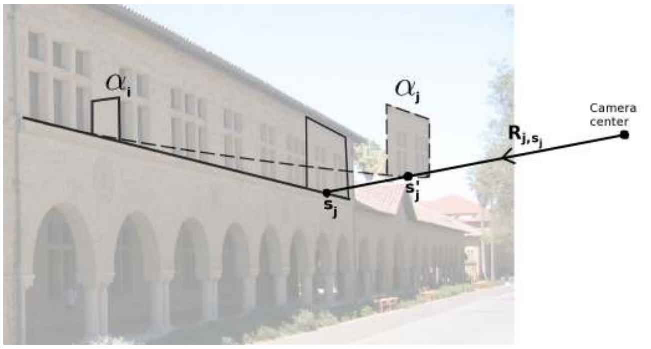


Figure 4. Illustration explaining the choice of s_j and s'_j for enforcing colinearity. A long straight line in the image is more likely to be a straight line in 3-d; therefore we penalize distance between s_j and s'_j.

The first term $P_1(\cdot)$ models the plane parameters α^n as a function of the monocular image features X^n, and penalizes the fractional errors in depths. It is parametrized by the parameters $\theta_r \in \mathbb{R}^{524}, r = 1, ..., 11$, and are learned from ground-truth laser data. Y^n denotes our confidence in the estimate of plane parameters from the monocular features. (See [5] for more details.)

The second term $P_2(\cdot)$ models the co-linearity, the connected structure, and the coplanarity properties by capturing 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:

$$P_2(\cdot) = \prod_{\{s_i, s_j\} \in N} h_{s_i,s_j}(.) \tag{1}$$

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

Co-linearity: We enforce the 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. In detail, we choose two superpixels α_i and α_j that lie on different portions of the straight line; we then choose a point p in the image lying on the long straight line, and let s_j be the 3-d position of p if it were to lie on the (infinite) plane parameterized by α_i, and let s'_j be the 3-d position of p if it were to lie the plane parameterized by α_j. Our model then penalizes the (fractional) distance between s_j and s'_j. (See Fig. 4.)

$$h_{s_j}(\alpha_i, \alpha_j, y_{ij}, R_{j,s_j}) = \exp\left(-y_{ij}|(R_{j,s_j}^T \alpha_i - R_{j,s_j}^T \alpha_j)\hat{d}|\right) \tag{2}$$

with $h_{s_i,s_j}(.) = h_{s_i}(.)h_{s_j}(.)$.

In detail, $R_{j,s_j}^T \alpha_j = 1/d_{j,s_j}$ and $R_{j,s_j}^T \alpha_i = 1/d'_{j,s_j}$; therefore, the term $(R_{j,s_j}^T \alpha_i - R_{j,s_j}^T \alpha_j)\hat{d}$ gives the fractional distance $\left|(d_{j,s_j} - d'_{j,s_j})/\sqrt{d_{j,s_j} d'_{j,s_j}}\right|$ for $\hat{d} = \sqrt{\hat{d}_{j,s_j} \hat{d}'_{j,s_j}}$. The "confidence" y_{ij} depends on the length of the line and its curvature—a long straight line in 2-d is more likely to be a straight line in 3-d.

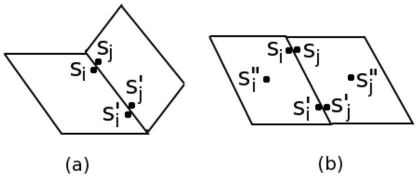


Figure 5. Illustration explaining effect of the choice of s_i and s_j on enforcing (a) Connected structure and (b) Co-planarity.

Connected structure and Co-planarity: 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. 5a, penalizing the (fractional) distance between two such points ensures that they remain fully connected.

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. 5b) 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}. (Please see [5] for more details.)

Correspondences: A point in the 3-d scene could appear in multiple images taken from different viewpoints. Therefore, if two points in two images match (see Section 5.2), then they are more likely to occupy the same 3-d location in the scene. More formally, if two points $p_n = (x_n, y_n, z_n)$ and $p'_n = (x'_n, y'_n, z'_n)$ from two images refer to the same 3-d location (in same coordinate frame) from two different views, then we want to minimize the (fractional) distance between them. The distance between the two points is given as

$$\begin{aligned} p'_n - p_n &= Q^{mn}[p_m; \quad 1] - p_n \\ &= Q^{mn}[R^m/({R^m}^T \alpha^m); \quad 1]) - R^n/({R^n}^T \alpha^n) \end{aligned}$$

Thus, to penalize the (fractional) distance, we have

$$\begin{aligned} & P_3(\alpha^n, \alpha^m | Q^n, Q^m, Y^{mn}) \\ & \propto \prod_{k=1}^{J^{mn}} \exp\Bigg(- y_k^{mn} \Big| (Q^{mn}[({R_{i(k)}^n}^T \alpha_{i(k)}^n) R_{j(k)}^m; \\ & \quad ({R_{i(k)}^n}^T \alpha_{i(k)}^n)({R_{j(k)}^m}^T \alpha_{j(k)}^m)] - ({R_{j(k)}^m}^T \alpha_{j(k)}^m) R_{i(k)}^n) \hat{d} \Big| \Bigg) \end{aligned}$$

where there are J^{mn} correspondences between images m and n. Here, $\hat{d} = \sqrt{\hat{d}_{i(k)}^n \hat{d}_{j(k)}^m}$, and $\hat{d}_{i(k)}^n = 1/({R_{i(k)}^n}^T \alpha_{i(k)}^n)$.

Note that this term penalizes distance between two corresponding points in 3-d in the same coordinate frame; therefore it will tend to bring the points closer even if the camera poses Q are slightly inaccurate, and can be used even if we have only rough camera poses. Specifically, it does not require the 3-d locations of the points (which would be available only if we run bundle adjustment in the triangulation

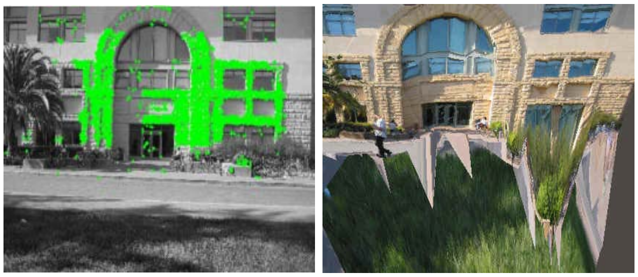


Figure 6. An image showing a few matches, and the resulting 3-d model without estimating the variables y for confidence in the 3-d matching. The noisy 3-d matches reduce the quality of the model. (Note the cones erroneously projecting out from the wall.)

step). Later, we further describe how enforcing a phantom planes constraint (Section 5.3) yields additional correspondences.

Depths from Triangulation: In an image n, there could be some points for which approximate depths d_T are obtained from triangulation (see Section 5.2). Since there could be errors in the triangulated depths, we penalize the (fractional) error in the triangulated depths d_{Ti} and $1/(R_i^T \alpha_i)$. For K^n points for which the triangulated depths are available, we have

$$P_4(\alpha|Q, d_T, Y_T) \propto \prod_{i=1}^{K^n} \exp\left(-y_{Ti} \left|d_{Ti} {R_i}^T \alpha_i - 1\right|\right).$$

This term places a "soft" constraint on a point in the plane to have its depth equal to its triangulated depth.

MAP Inference: For MAP inference of the plane parameters, we need to maximize the conditional log-likelihood $\log P(\alpha|X, Y, d_T; \theta)$. All the terms in $P_1(\cdot)$, $P_2(\cdot)$ and $P_4(\cdot)$ corresponds to L_1 norm terms; thus MAP inference in an MRF that uses only these terms can be efficiently solved using a Linear Program (LP) [5].[3] To solve the LP, we implemented an efficient method that takes advantage of the sparsity in our problem.

5.2. Triangulation Matches

In this section, we will describe how we obtained the correspondences across images, the "confidences" y_k^{mn} in these correspondences, and the triangulated depths d_T, used in the $P_3(\cdot)$ and $P_4(\cdot)$ terms in Section 5.1.

Two points that appear to match in two images are somewhat likely to be at the same 3-d location. However, many of these 3-d correspondences are noisy; for example, local structures are often repeated across an image (e.g., Fig. 8a and 6). Therefore, we also model the "confidence" y_k^{mn} in the k^{th} match between images m and n, by estimating the probability $P(y_k^{mn} = 1)$ of the match being correct. To estimate how likely a match is correct, we use neighboring 3-d matches as a cue. For example, a group of spatially consistent 3-d matches is more likely to be correct than a single isolated 3-d match. We capture this by using a feature vector that counts the number of matches found in the present superpixel and in larger surrounding regions (i.e., at multiple spatial scales), as well as measures the relative quality between the best and second best match. We use these correspondences directly in our probabilistic model in term $P_3(\cdot)$ without requiring to explicitly estimate the 3-d locations of the points (Section 5.1).

We can also compute depths from triangulation by first using the monocular approximate depths to remove the scale ambiguity, and then using bundle adjustment [26] to refine our matches. In detail, we start by computing 128 SURF features [27], and then calculate matches based on the Euclidean distances between the features found. Then to compute the camera poses $Q = [\text{Rotation}, \quad \text{Translation}] \in \mathbb{R}^{3\times 4}$ and the depths d_T of the points matched, we use bundle adjustment [26]. These triangulated depths are used in the term $P_4(\cdot)$ in Section 5.1.

Improving matching performance using monocular cues: Increasingly many cameras and camera-phones come equipped with GPS, and sometimes also accelerometers (which measure gravity/orientation). Many photo-sharing sites also offer geo-tagging (where a user can specify the longitude and latitude at which an image was taken). In this section, we will describe how such geo-tags (together with a rough user-specified estimate of camera orientation) can be used, together with monocular cues, to improve the performance of correspondence algorithms.

We compute the approximate depths of the points using monocular image features as $\hat{d} = x^T \theta$; this requires only computing a dot product and hence is fast (see Section 5.1). Now, for each point in an image B for which we are trying to find a correspondence in image A, typically we would search in a band around the corresponding epipolar line in image A. However, given an approximate depth estimated from from monocular cues, we can limit the search to a rectangular window that comprises only a subset of this band (Fig. 7). This reduces the time required for matching, and also improves the accuracy significantly when there are repeated structures in the scene.

To illustrate the applicability of our algorithm even to settings where geo-tags are not available, our experiments (Section 5.4) will report results on models built without using geo-tags (i.e., the camera position/orientation was not known in advance). We also performed additional experiments in which an approximate camera pose was entered by

[3] Actually, the correspondence term in $P_3(\cdot)$ is not convex. These terms occur for correspondences that do not have associated triangulated depths. To address this, one can envisage a variety of algorithms, but we found that a simple approximation method that solves a series of LPs works very well in practice. In detail, we take the estimate of $\hat{d}_{n_{i(k)}}$ from the last LP, and use it to replace the term $({R^n_{i(k)}}^T \alpha^n_{i(k)})({R^m_{j(k)}}^T \alpha^m_{j(k)})$ in $P_3(\cdot)$ with $1/\hat{d}_{n_{i(k)}} \hat{d}_{m_{j(k)}}$, solve the new LP, and iterate a few times. This entire process typically takes about 30 seconds.

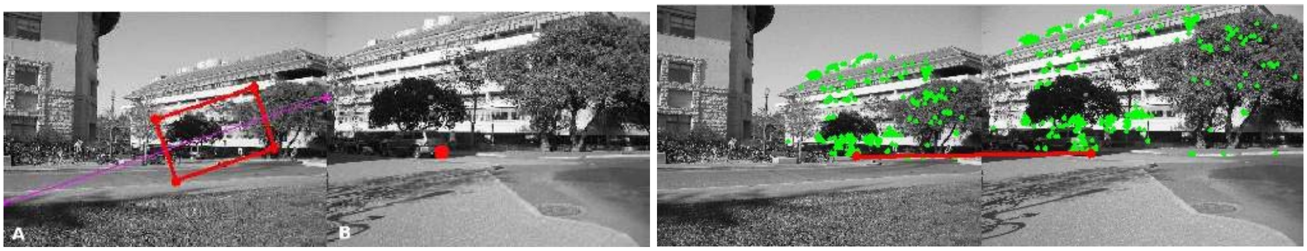


Figure 7. (a) Corresponding region to search for in image A, for a point in image B, (b) Correspondences found using our monocular depth estimates.

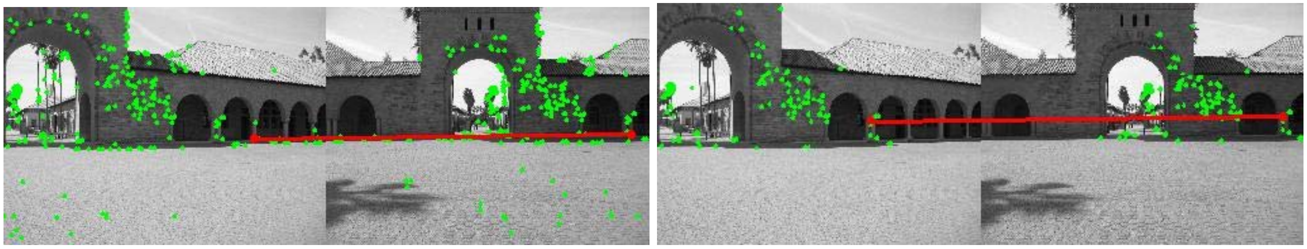


Figure 8. (a) Bad correspondences, caused by repeated structures in the world. (b) Use of monocular depth estimates results in better correspondences. Note the these corresponses are still fairly sparse and slightly noisy, and are therefore insufficient for creating a good 3-d model if we do not additionally use monocular cues.

a user on a 2-d map; this resulted in fairly similar models to the version that did not use geo-tags, but ran about three times faster.

5.3. Phantom Planes

This cue enforces occlusion constraints across multiple cameras. Concretely, each small plane (superpixel) comes from an image taken by a specific camera. Therefore, there must be an unoccluded view between the camera and the 3-d position of that small plane—i.e., the small plane must be visible from the camera location where its picture was taken, and it is not plausible for any other small plane (one from a different image) to have a 3-d position that occludes this view. This cue is important because often the connected structure terms, which informally try to "tie" points in two small planes together, will result in models that are inconsistent with this occlusion constraint, and result in what we call "phantom planes"—i.e., planes that are not visible from the camera that photographed it. We will use a Laplacian model (L_1 penalty) to penalize the distance between the offending phantom plane and the plane that occludes its view from the camera. This tends to make the two planes lie in exactly the same location (i.e., have the same plane parameter), which eliminates the phantom/occlusion problem.

5.4. Experiments

In [5], we described an earlier version of this algorithm that used monocular cues to produce 3-d models from single still images. That algorithm was shown to produce qualitatively correct 3-d models for 64.9% of 588 images downloaded from the internet. Some depthmaps produced by it are shown in Fig. 9. Other than assuming that the environment is made of a number of small planes, the algorithm did not make any explicit assumptions about the structure of the scene, such as the "ground-vertical" assumption by Delage et al. [11] and Hoiem et al. [7]; thus it was able to generalize well, even to scenes with significant non-vertical structure.

In this experiment, we create a photo-realistic 3-d model of a scene given only a few images (with unknown location/pose), even ones taken from very different viewpoints or with little overlap. Fig. 10, 11, 12 and 13 show snapshots of some 3d models output by our algorithm. Using monocular cues, our algorithm is able to create full 3-d models even when large portions of the images have no overlap (Fig. 10, 11 and 12). In Fig. 10, monocular predictions (not shown) from a single image gave approximate 3-d models that failed to capture the arch structure in the images. However, using both monocular and triangulation cues, we were able to capture this 3-d arch structure.

More models are available at:

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

6. Conclusions

We have presented an algorithm that builds large 3-d reconstructions of outdoor environments, given only a small number of images. Our algorithm segments each of the images into a number of small planes, and simultaneously infers the 3-d position and orientation of each plane in every image. Using an MRF, it reasons about both monocular depth cues and triangulation cues, while further taking into account various properties of the real world, such as occlusion, co-planarity, and others.

Acknowledgments: We thank Jeff Michels, Rajiv Agarwal, Marc Pollefeys and Sebastian Thrun 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.

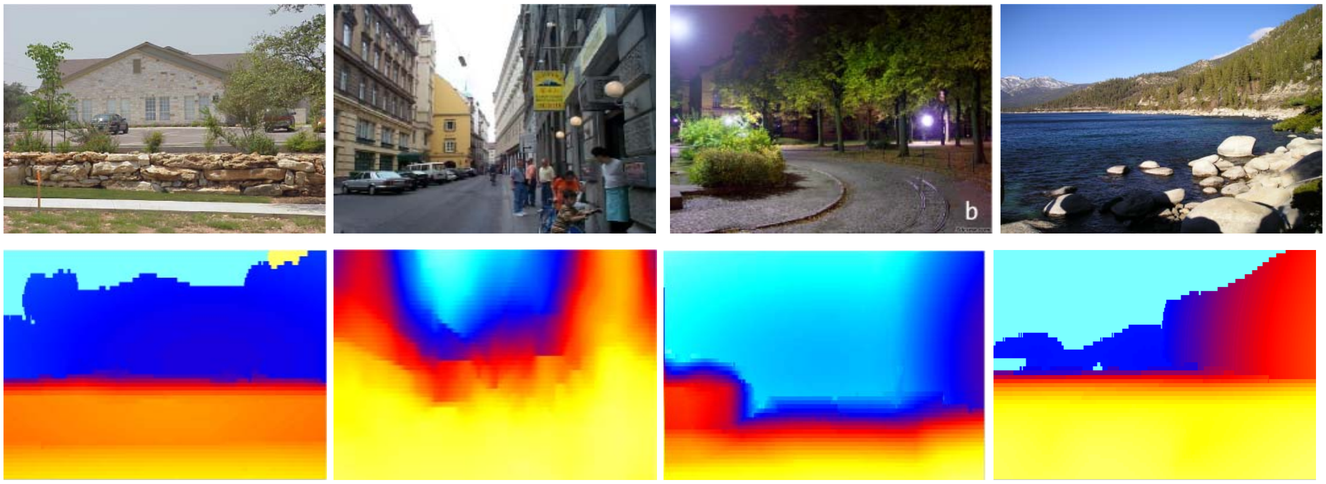
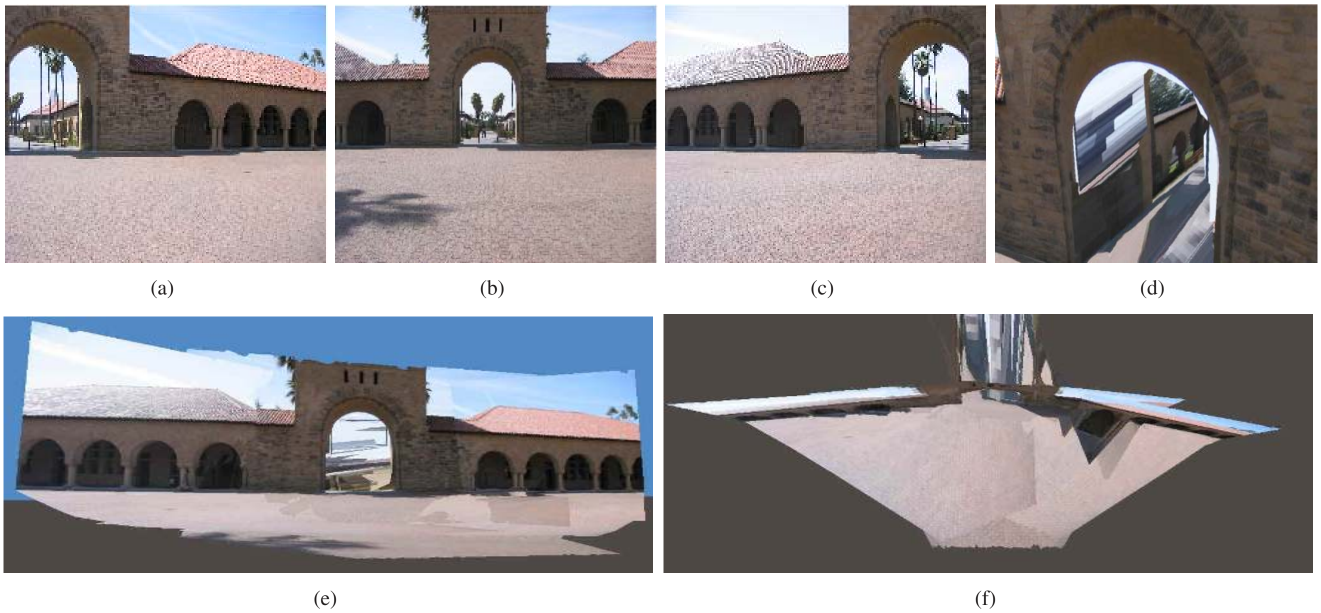


Figure 9. Typical results from our algorithm. (Top row) Original images, (Bottom row) depthmaps (shown in log scale, yellow is closest, followed by blue) generated from the images. Colors indicate depth, see color-scale bottom row.


(a) (b) (c) (d)

(e) (f)

Figure 10. (a,b,c) Three original images from different viewpoints; (d,e,f) Snapshots of the 3-d model predicted by our algorithm. (f) shows a top-down view; the top part of the figure shows portions of the ground correctly modeled as lying either within or beyond the arch.

References

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

[2] R. Hartley and A. Zisserman, *Multiple View Geometry in computer vision*. Cambridge, 2003.

[3] M. Pollefeys, “Visual modeling with a hand-held camera,” *IJCV*, vol. 59, 2004.

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

[5] A. Saxena, M. Sun, and A. Y. Ng, “Learning 3-d scene structure from a single still image,” in *ICCV workshop on 3D Representation for Recognition (3dRR-07)*, 2007.

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

[7] D. Hoiem, A. Efros, and M. Herbert, “Geometric context from a single image,” in *ICCV*, 2005.

[8] N. Snavely, S. M. Seitz, and R. Szeliski, “Photo tourism: Exploring photo collections in 3d,” *SIGGRAPH*, vol. 25, no. 3, 2006.

[9] S. Thrun and M. Montemerlo, “The graphslam algorithm with applications to large-scale mapping of urban structures,” *IJRR*, vol. 25.

[10] A. Akbarzadeh and et al., “Towards urban 3d reconstruction from video,” in *3DPVT*, 2006.

[11] 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.

[12] A. Criminisi, I. Reid, and A. Zisserman, “Single view metrology,” *IJCV*, vol. 40, pp. 123–148, 2000.

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

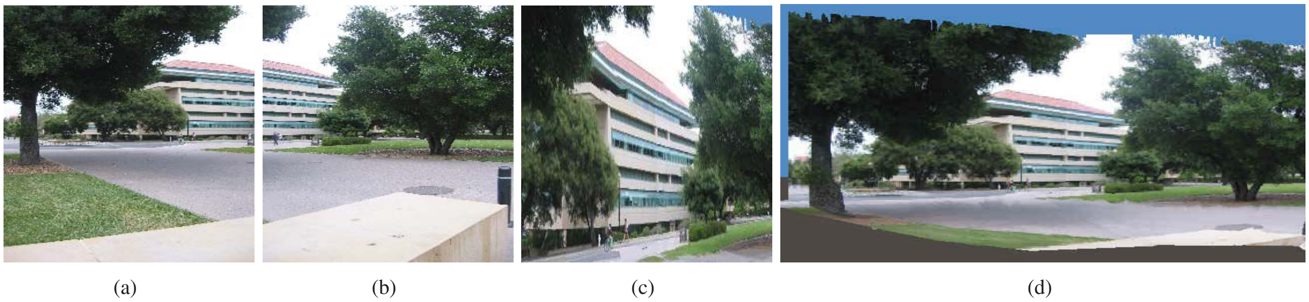

(a) (b) (c) (d)

Figure 11. (a,b) Two original images with only a little overlap. (c,d) Snapshots from our inferred 3-d model.

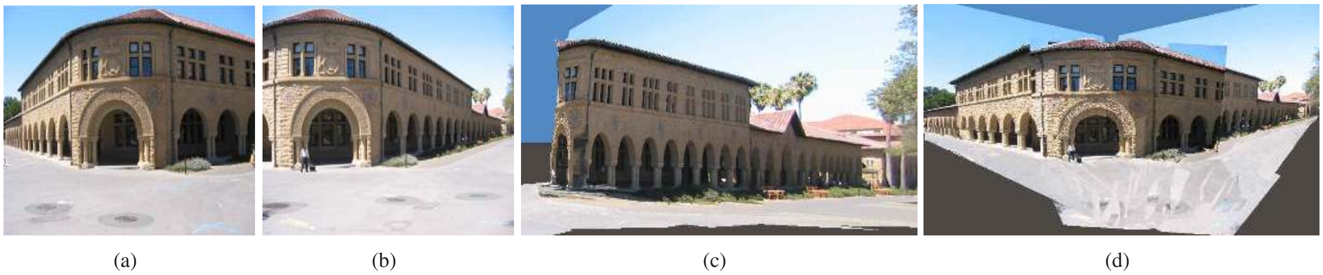
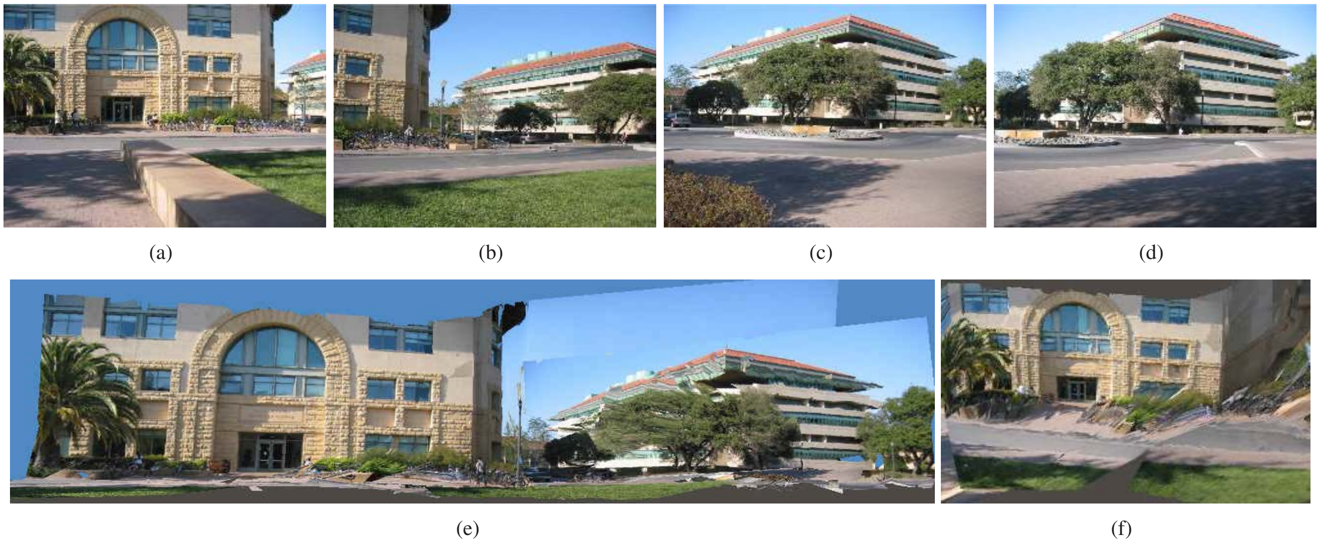

(a) (b) (c) (d)

Figure 12. (a,b) Two original images with many repeated structures; (c,d) Snapshots of the 3-d model predicted by our algorithm.


(a) (b) (c) (d)

(e) (f)

Figure 13. (a,b,c,d) Four original images; (e,f) Two snapshots shown from a larger 3-d model created using our algorithm.

[14] A. Torralba and A. Oliva, "Depth estimation from image structure," *IEEE PAMI*, vol. 24, no. 9, pp. 1–13, 2002.

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

[16] J. Michels, A. Saxena, and A. Y. Ng, "High speed obstacle avoidance using monocular vision and reinforcement learning," in *ICML*, 2005.

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

[18] J. Yan and M. Pollefeys, "Articulated motion segmentation using ransac with priors," in *ICCV Workshop on Dynamic Vision*, 2005.

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

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

[21] J. Loomis, "Looking down is looking up," *Nature News and Views*, vol. 414, pp. 155–156, 2001.

[22] J. Porrill, J. P. Frisby1, W. J. Adams, and D. Buckley, "Robust and optimal use of information in stereo vision," *Nature*, vol. 397, 1999.

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

[24] P. Felzenszwalb and D. Huttenlocher, "Efficient graph-based image segmentation," *IJCV*, vol. 59, 2004.

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

[26] M. Lourakis and A. Argyros, "A generic sparse bundle adjustment c/c++ package based on the levenberg-marquardt algorithm," Foundation for Research and Technology - Hellas, Tech. Rep., 2006.

[27] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," in *ECCV*, 2006.