Piecewise Planar and Non-Planar Stereo for Urban Scene Reconstruction

David Gallup[1], Jan-Michael Frahm[1], and Marc Pollefeys[2]

Department of Computer Science[1]
The University of North Carolina
Chapel Hill, NC, USA
{gallup,jmf}@cs.unc.edu

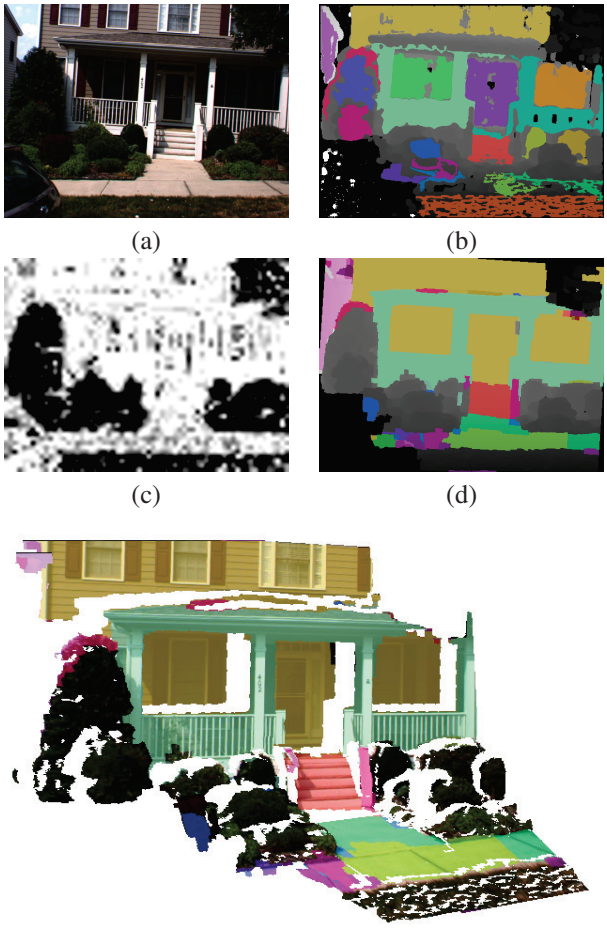
Department of Computer Science[2]
ETH Zürich
Zürich, Switzerland
marc.pollefeys@inf.ethz.ch

Abstract

Piecewise planar models for stereo have recently become popular for modeling indoor and urban outdoor scenes. The strong planarity assumption overcomes the challenges presented by poorly textured surfaces, and results in low complexity 3D models for rendering, storage, and transmission. However, such a model performs poorly in the presence of non-planar objects, for example, bushes, trees, and other clutter present in many scenes. We present a stereo method capable of handling more general scenes containing both planar and non-planar regions. Our proposed technique segments an image into piecewise planar regions as well as regions labeled as non-planar. The non-planar regions are modeled by the results of a standard multi-view stereo algorithm. The segmentation is driven by multi-view photoconsistency as well as the result of a color- and texture-based classifier, learned from hand-labeled planar and non-planar image regions. Additionally our method links and fuses plane hypotheses across multiple overlapping views, ensuring a consistent 3D reconstruction over an arbitrary number of images. Using our system, we have reconstructed thousands of frames of street-level video. Results show our method successfully recovers piecewise planar surfaces alongside general 3D surfaces in challenging scenes containing large buildings as well as residential houses.

1. Introduction

Automatic dense 3D reconstruction from images and video has long been a challenge in computer vision. Recently, fitting a scene with a piecewise planar model has become popular for reconstructing urban scenes [5, 17, 22], as it has several advantages. The strong planarity assumption overcomes the challenges presented by poorly textured or specular surfaces that are often characteristic of man-made planar structures, and the resulting models are low complexity, which is important for rendering, storage, and transmission. Furthermore, simplified geometry can often *look better*, even if the overall surface accuracy is lower,



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

Figure 1. (a) Original image. (b) Planes found by RANSAC in depthmap. (c) Planar class probability. (d) Final plane labeling overlaid on depthmap. Colors = planes, gray = no plane, and black = discarded. (e) Resulting 3D model with planes highlighted.

since piecewise planar surfaces better resemble man-made urban structures.

However, such a model performs poorly in the presence of highly non-planar objects such as trees, cars, bushes, and other clutter present in urban scenes. Recent work such as Furukawa *et al*. [5] and Sinha *et al*. [17], produce very convincing 3D reconstructions for man-made architectural scenes, as far as the scene is piecewise planar. But for non-planar objects and clutter, the reconstructions can appear unnatural or even completely incorrect. To address this problem, we present a stereo method capable of handling more general scenes containing both planar and non-planar regions. Our proposed technique segments an image into piecewise planar regions as well as regions which are labeled as non-planar. The non-planar regions are modeled by the output of a standard multi-view stereo algorithm. Thus, our method maintains the advantages of piecewise planar stereo, while also having the ability to fall-back to a general representation to handle non-planar surfaces.

The inputs to our algorithm are a video sequence or collection of images, intrinsic and extrinsic camera calibration for each image, and a dense depthmap for a subset of the views. The output is a refined set of piecewise planar and non-planar depthmaps that are then used to generate a textured polygonal mesh representation of the scene. The initial camera poses and depthmaps can be obtained with a variety of structure from motion (SfM) [14, 18] and dense stereo techniques [8, 13].

From the initial set of depthmaps, a number of plane hypotheses are found using a RANSAC method (Figure 1b). Similiar to Furukawa *et al*. [5] and Sinha *et al*. [17], for each input depthmap, we set up an MRF problem where each pixel is assigned a label corresponding to one of the previously obtained plane hypotheses. The key difference in our approach is the addition of a *non-plane label* which represents the input stereo depthmap. Label likelihoods are defined as the photoconsistency of the plane, in case of a plane label, or of the depthmap, in case of the non-plane label. In the spirit of model selection, the non-plane label incurs an additional penalty, due to the higher degrees of freedom in the depthmap surface. A smoothness prior is defined that penalizes label transitions and is weighted by surface continuity and image gradients. The resulting energy functional is minimized using graph-cuts [2, 3, 11, 19].

To further help distinguish planar and non-planar surfaces, we have trained a classifier based on image color and texture features. The training set includes image segments that have been hand-labeled as either planar or non-planar. A k-nearest neighbor classifier then produces a planar class membership probability for each segment of the oversegmented input images (Figure 1c), and the probability is included in the label likelihood. The reason for this additional constraint derives from our piecewise planar assumption. It may very well be that a plane fits a bush or sloping ground, at least within the uncertainty of the stereo reconstruction. It is in fact the appearance of these image regions that indicate they are non-planar. This constraint also helps to ensure the correct plane label for specular surfaces such as windows which may have poor photoconsistency.

Additionally, our method links and fuses the initial plane hypotheses across overlapping views, ensuring a consistent 3D reconstruction over an arbitrary number of images. Plane segments for which the point-to-plane distances fall within a certain threshold are linked, and the plane estimates are fused. Because overlapping views can share the same plane hypotheses, the image labeling can be performed independently for each view, and the resulting 3D model will be consistent, *i.e*. a single planar surface can be extended indefinitely. Also, because each image is processed separately, our algorithm is *out-of-core*, needing only enough memory for one view at a time, and making it highly scalable. Using our system, we have reconstructed thousands of frames of street-level video. Results show our method sucessfully recovers piecewise planar surfaces alongside general 3D surfaces in challenging scenes containing large buildings as well as residential houses.

2. Related Work

Furukawa *et al*. [5] use a very specific Manhattan-world model, where all planes must be orthogonal, and Sinha *et al*. [17] use a general piecewise planar model. Non-planar surfaces are not handled well and are either reconstructed with a staircase appearance or are flattened to nearby planes. The work of Zebedin *et al*. [22], focuses on aerial imagery, and in addition to planar rooftops allows for a surface of revolution representation to handle domes and spires. Our model allows for a general depthmap reconstruction as an alternative to planes, which handles any non-planar surface.

Zebedin *et al*. [22] require each building segmented as input, and each building is processed independently, making it trivial to scale to large datasets. Furukawa *et al*. [6] present a Manhattan-world fusion technique for the purpose of generating floor plans for indoor scenes. Multiple views must be fused in a single volumetric representation, limiting the overall size of the reconstruction. We use a multi-view plane linking approach which allows images to be processed separately (out-of-core), and can produce consistent reconstructions over datasets of arbitrary size.

Hoiem *et al*. [9] and Saxena *et al*. [16] use color, texture, and other image features to infer geometric context. They are able to create a plausible 3D representation from a single view, however no depth measurements are made. We use many of the same features as [9], although our classification problem is much simpler. Our planar versus non-planar classifier is used in addition to photoconsistency and smoothness constraints in the image labeling task. Xiao *et*

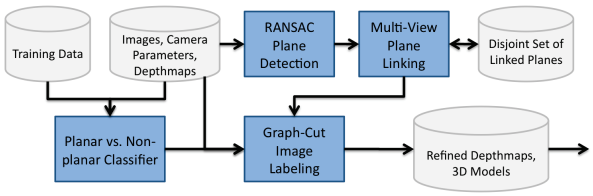


Figure 2. Our piecewise planar and non-planar stereo system.

al. [21] use trained classifiers and multi-view constraints to segment street-side images into ground, tree, building, and sky regions. 3D models are then fit to the buildings. We also combine learned image appearance with multi-view constraints, but no hard decision is made until the the final plane labeling.

3. Piecewise Planar and Non-Planar Stereo

The steps of our algorithm are layed out in Figure 2. The input to our algorithm is a collection of images, camera poses, and depthmaps. Depthmaps can be computed using any standard stereo method. This reconstruction is typically poor for weakly textured and specular surfaces, but the result is sufficient to initialize our algorithm.

3.1. Plane Hypothesis Generation

We first obtain a set of plane hypotheses for every image using a RANSAC method. Typically one seeks to find a single model to fit all the data, but our objective is to find multiple locally fit models. In this regard, there are several important aspects of our method which are crucial to achieving a good set of planes.

- **Sampling.** A plane model can be obtained from three points in the depthmap sampled at random. The first point is selected from a uniform distribution over the image. The second two points are selected from normal distributions centered at the first point with a standard deviation of σ.

- **Scoring.** Each model is evaluated against only points nearby the original samples. Only points within M pixels of the first sample are considered. Instead of scoring simply by the inlier count (number of points within a threshold distance to the plane), we score by the liklihood of each point fitting the plane, according to the MLE-sac method [20].

- **Contiguity.** After RANSAC returns a plane, the inlier set is determined by computing the distance of each point to the plane. Additionally, the inlier set is restricted to points which are connected (contiguous) to the initial sample, according to the image graph. A new plane is obtained as the least-squares fit to the inlier points. Inliers are again determined, and the process is repeated for several iterations. (This contiguity constraint is not used inside the RANSAC sampling loop for performance reasons.)

The final set of inliers is then removed from the image, and RANSAC is again repeated on the remaining points. For each image we obtain a set of N planes $\Pi = \{\pi_1 \cdots \pi_N\}$. This includes most of the major planes in the scene, as well as some spurious planes which happen to fit well to non-planar or quasi-planar structures. We add to each set the plane at infinity, denoted π_∞, which is useful for labeling sky or distant surfaces which are not reconstructed by stereo. At this point, an initial labeling of each image can be performed, simply by assigning each inlier set from RANSAC to its respective plane.

For all of our experiments we use $\sigma = 8$ pixels, $M = 100$ pixels, and $N = 20$ planes. Note that our plane detection method is much simpler than that of Sinha *et al*. [17]. One reason is that [17] operates on points and lines while our method operates on depthmaps.

3.2. Multi-View Plane Linking

For multi-view reconstructions, it is imperative to obtain consistent plane hypotheses across overlapping views. A planar surface visible in several images will generate slightly varying plane hypotheses, due to small variations in the depthmaps. Also, it is intractable to consider every plane for every image when processing large datasets. Thus we perform a single pass over all images and establish links across nearby views between mutually supporting planes. All linked plane hypotheses are fused to give a single multi-view estimate for the plane.

Planes are linked as follows. For every plane π_i, the set of all planes in nearby views, including the planes in the same view as π_i, is considered for linking. For every plane π_j in that set, if a sufficient number of points (90%) belonging to π_i falls within a threshold distance (1% of the camera-to-point distance) of π_j, then π_i and π_j are linked. A new plane is fit to the combined set of points belonging to all the linked planes. A global disjoint set data structure is created which maintains each set of linked planes. The disjoint set can be held in memory at all times, since only a few bytes are required to identify a plane. This ensures that surfaces seen in multiple images have the exact same plane hypothesis. It also serves to link similar planes from repeated structures, or single planes which appear disjoint in the images due to occlusion. See Figure 3.

3.3. Graph-Cut Labeling

Once the plane hypotheses have been established, the next step is to perform a pixel-wise labeling of each image. Each image is solved independently, but since plane hypotheses have been fused, the resulting depthmaps will

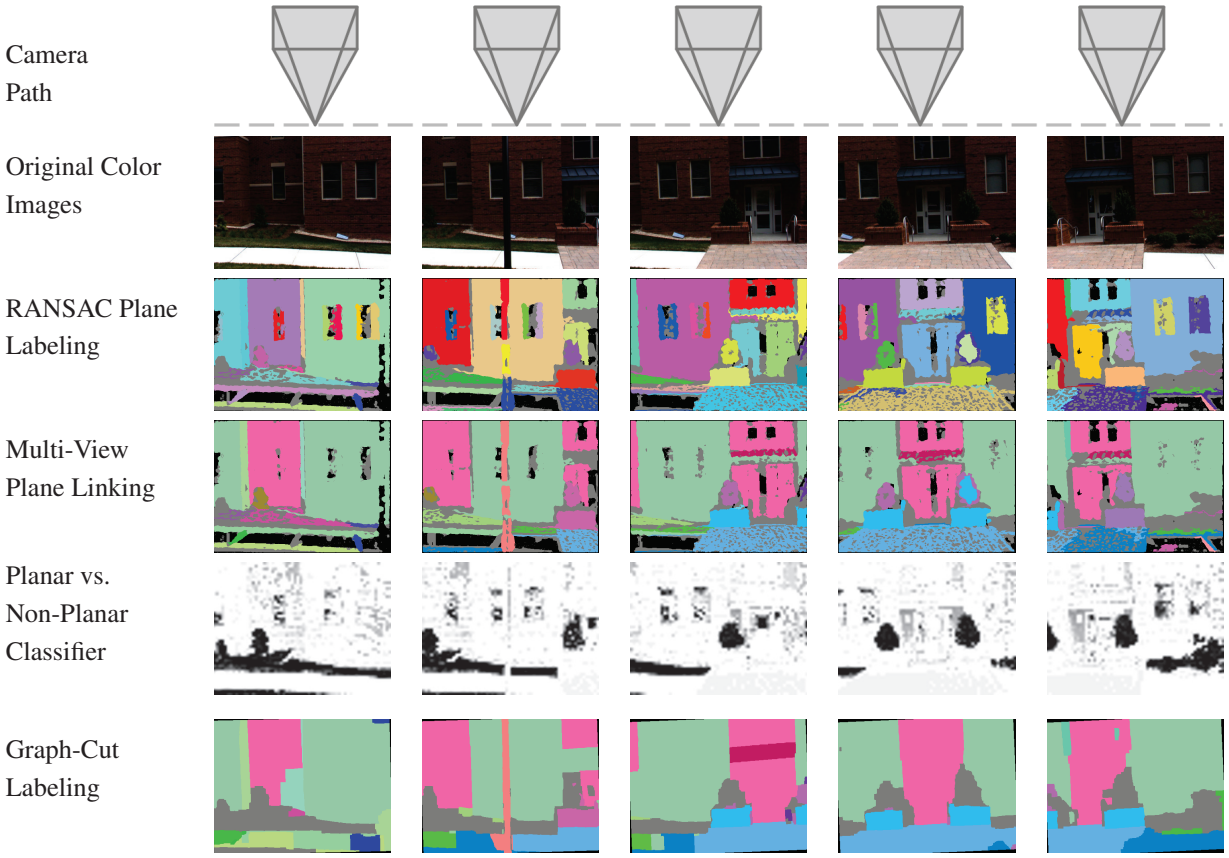


Figure 3. The second row shows a subset of the images used to compute the depth maps. The third row illustrates the the plane detection results of our modified RANSAC technique. The fourth row shows the plane labels after linking. Note that the same scene plane now has a consistent label. The fifth row shows the classification results of the image into planar or non-planar structure (planar class probability black=0, white=1). The sixth row shows the results of the graph cut based plane assignment. See Figure 6 for the resulting 3D model.

be globally consistent. For each image, an MRF is defined which leads to the standard energy minimization problem involving data and smoothness terms. Our goal is to obtain a labeling so as to minimize the energy functional

$$E(L) = \sum_{p \in \mathcal{I}} E_{data}(L(p)) + \sum_{p,q \in \mathcal{N}} \lambda_{smooth} E_{smooth}(L(p), L(q)) \quad (1)$$

where $\mathcal{I}$ is the set of pixels in the image, $\mathcal{N}$ is the standard 4-neighborhood, and L is the labeling.

The set of labels is the union of all planes, a non-plane label, and a discard label. The labeling function $L : \mathcal{R}^2 \rightarrow \{\pi_1, \cdots, \pi_N, \pi_\infty, \textit{non-plane}, \textit{discard}\}$ identifies a label for every pixel. The *non-plane* label indicates the original stereo depthmap, and the *discard* label indicates no reliable reconstruction could be obtained.

The E_{data} function is defined as

$$E_{data}(l) = \begin{cases} min(\rho(l), \rho_{max}) & \text{if } l \in \{\pi_1 \cdots \pi_\infty\} \\ min(\rho(l), \rho_{max}) + \rho_{bias} & \text{if } l = \textit{non-plane} \\ \alpha \rho_{max} & \text{if } l = \textit{discard} \end{cases} \quad (2)$$

where ρ is a photoconsistency (dissimilarity) measure between the pixels in nearby views put into correspondence by the assigned plane, or by the original stereo depthmap. For photoconsistency we use the Birchfield-Tomasi pixel-to-pixel dissimilarity measure [1]. For occlusion handling we use the multi-view technique of [10]. For the *non-plane* label, a penalty ρ_{bias} is given in order to penalize the model with more degrees of freedom. The dissimilarity measures have been truncated to ρ_{max} in order to handle poorly matching specular or reflective surfaces such as windows. The *discard* label receives slightly less penalty than maximum. Thus small poorly matching regions will be labeled according to their surroundings due to the smoothness term, but large poorly matching regions will incur enough cost to be discarded. For all our experiments we set $\rho_{max} = 6$, $\rho_{bias} = 0.5$, and $\alpha = 0.9$. (See Figure 3.)

The E_{smooth} function is defined as

$$E_{smooth}(l_p, l_q) = g \cdot \begin{cases} 0 & \text{if } l_p = l_q \\ d_{max} & \text{if } l_p \text{ or } l_q \in \{\pi_\infty, \textit{discard}\} \\ d' & \text{otherwise} \end{cases} \quad (3)$$

$$d' = min(d, d_{max}) + d_{min} \quad (4)$$

$$g = \frac{1}{\gamma \|\partial I/\partial u\|^2 + 1} \quad (5)$$

where d is the distance between the 3D neighboring points according to their labels, and g is the image gradient magnitude (color or grayscale) between the two neighbors. Our video sequences were captured along with GPS data, so absolute distances can be measured. Otherwise, distances can be defined relative to the median value in the depthmap for example. d_{min} incurs a minimum penalty in order to prevent spurious transitions between planes that are close in 3D. d_{max} makes the penalty robust to discontinuities. For all our experiments we set $\lambda_{smooth} = 5$, $d_{min} = 2$, $d_{max} = 0.2$ meters, and $\gamma = 10$.

The energy can be minimized using the well-known multi-label graph-cut method [3]. One limitation of graph-cuts, and the discrete MRF in general, is that of metrication, which follows a manhattan distance, not a euclidean one. This leads to stair-case and other artifacts. However, we use this to our advantage in man-made scenes, where the vertical direction and dominant facade normal can be readily obtained [7] from the vanishing points of the scene structure. The image can then be rectified so that the horizontal and vertical vanishing points correspond to the x and y axes. Then the Manhattan distance metrication actually helps to enforce that label boundaries follow vertical and horizontal lines.

3.4. Planar Classifier

Even with the *non-plane* label available, surfaces such as bushes, trees, and grass are occasionally detected and assigned by the graph-cut solution. Ultimately for some regions, within the uncertainty of the stereo depth, a plane may well fit those surfaces. However, this leads to an undesirable result since common experience does not support planes in such natural objects. To this end, we train a classifier based on color and texture features to distinguish between surfaces that appear planar, and those that do not. Features are computed from image patches. Inspired by [9], we use the following color features: mean red, green, and blue (RGB color space), mean hue, saturation, value (HSV color space), and the hue histogram (5 bins). We use the following features computed from the edge orientation histogram [12]: entropy, maximum value, and number of modes. The texture features capture the fact that man-made objects tend to have only a few consistent edge orientations, while natural objects have a less structured appearance.

Each image is segmented into a grid of 16×16 pixel cells, and the feature vector is computed for each cell. We experimented with commonly used oversegmentation (superpixel) algorithms [15, 4], but in the end we preferred the regular grid. We have found that there is enough information in the photoconsistency and smoothness penalties to find accurate object boundaries. Therefore we prefer the grid as it ensures segments of a regular size and density. Approximately five thousand segments in five images were hand labeled as planar or non-planar. For a given input image, the planar class probablility for each grid cell is computed using k-nearest-neighbors. In the end we are interested in the class membership probability, as we will defer the final plane labeling decision until the graph-cut.

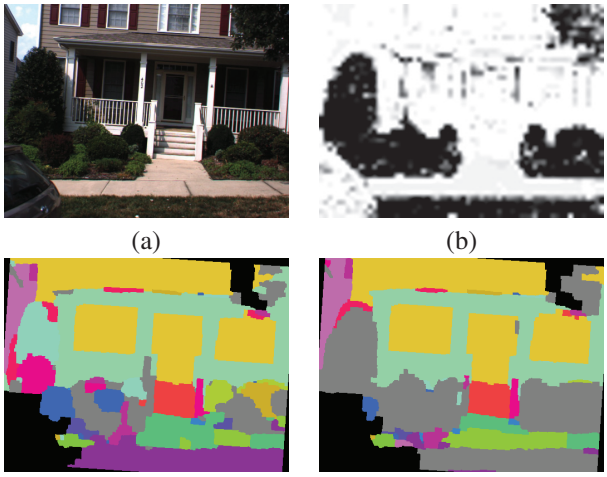


Figure 4. Result of graph-cut labeling with and without the planar class probability term. (a) Original color image. (b) Planar class probability. (c) Graph-cut labeling result without the class probability term. (d) Labeling result with the class probability term, which helps to remove many false planes labeled in the bushes and grass.

Let $a \in [0, 1]$ be the planar class probability for a given segment, and l be the label of pixel within that segment. The data term now becomes

$$E'_{data}(l) = E_{data}(l) + \lambda_{class} \begin{cases} 1-a & \text{if } l \in \{\pi_1, \cdots, \pi_\infty\} \\ a & \text{if } l = \textit{non-plane} \\ 0 & \text{if } l = \textit{discard}. \end{cases} \quad (6)$$

We have set $\lambda_{class} = 2$ for all our experiments. Figure 4 demonstrates the effect the class penalty has on the labeling result.

4. Results

To test our system, we have processed street-side video captured by two vehicle-mounted Point Grey Flea2 1024x768 color cameras. The cameras are aimed perpendicular to the driving direction, with one camera pointed horizontally and the other pointed upwards at 30 degrees. The composite camera system has a horizontal field of view of 120 degrees, and a vertical field of view of 60 degrees. The captured data contains a variety of street-level scenes

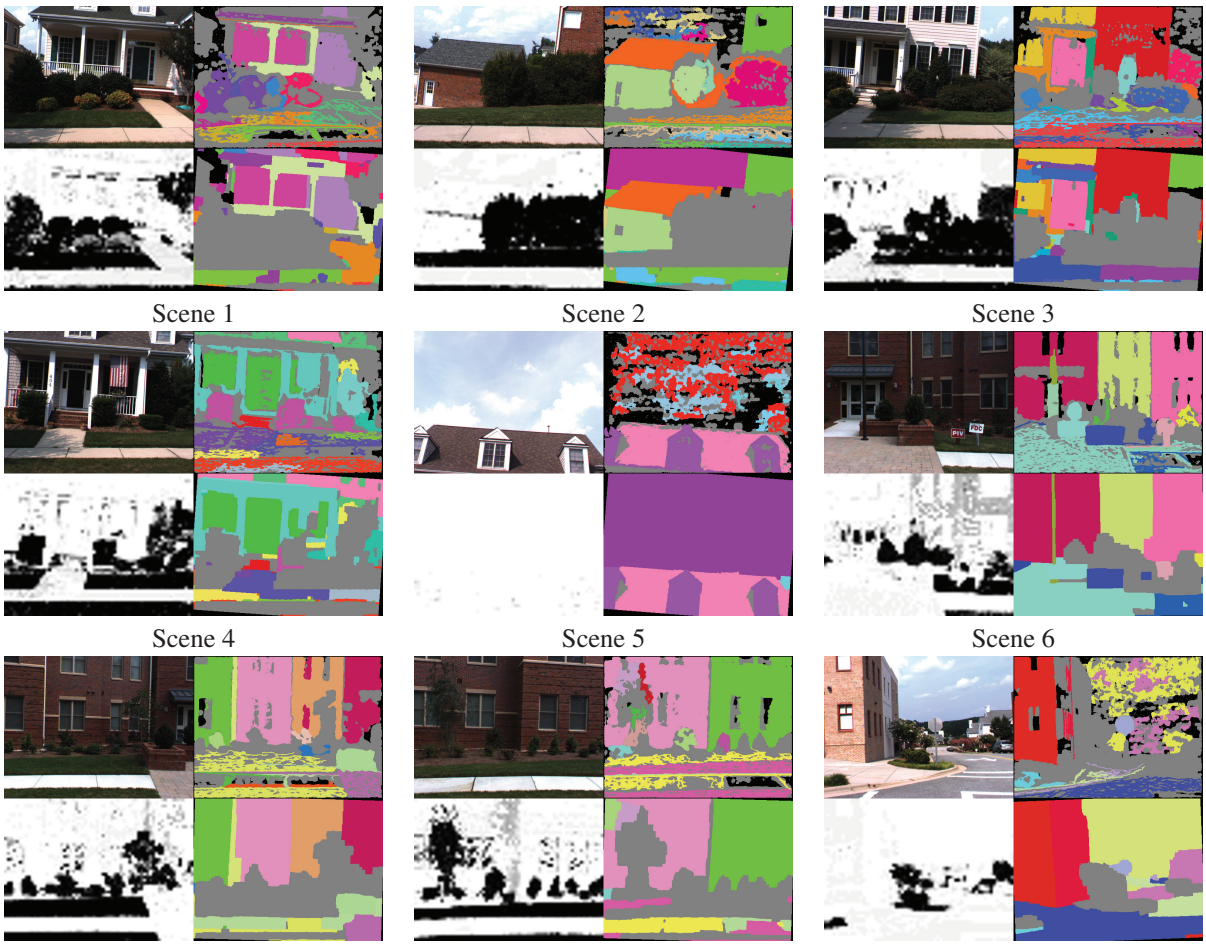


Figure 5. Results from the various steps of our algorithm for several scenes. Each of the four panes is as follows. Top Left: original color image. Top Right: RANSAC planes. Bottom Left: Planar class probability. Bottom Right: Final graph-cut labeling.

including large buildings and residential houses. These scenes contain large planar facade surfaces, but also contain many non-planar objects such as bushes, trees, and cars.

For all our experiments we have used the same parameters which have been given throughout Section 3. Parameters were chosen empirically and without much difficulty. The fact that we used the same set of parameters for several diverse datasets indicates that the parameters are not overly sensitive.

Although the end goal of our approach is to produce depthmaps, we have evaluated the final graph-cut labels for accuracy against hand-labeled test set of 22700 planar and non-planar image segments in 28 images. Since the graph-cut labels are computed on the full resolution, the segment label is determined by majority vote. Any of $\{\pi_1, \cdots, \pi_N, \pi_\infty\}$ count as planar, and either of $\{$*non-plane*, *discard*$\}$ count as non-planar. 94.7% of the planar segments and 97.2% of the non-planar segments were labeled correctly.

Figure 5 shows 9 images sampled from our results. For each image, the results of the RANSAC plane detection, planar classification, and graph-cut labeling are shown. In each scene, most of the major planes are found by our RANSAC method, although some planes are occasionally missed, especially when they occupy only a small portion of the image. The planar classifier performs well, despite its simplicity, and provides a good cue for the final graph-cut labeling to select between plane labels and the *non-plane* label. The final graph-cut labeling recovers broad planar surfaces while also identifying non-planar surfaces. Note that even though the planar classifier is performed on a coarse grid, the graph-cut result recovers fine object boundaries due to the photoconsistency constraint.

The number of input video frames ranges from 200 to 800 (see Figures 6-8 for exact numbers), and a refined depthmap is computed for every 10th frame. Our unoptimized C++ implementation takes about 1 to 2 minutes per depthmap for all steps.

Final 3D Model

3D Model with Highlighted Planes

Before

After

Before

After

Figure 6. 3D model produced by our piecewise planar and non-planar stereo algorithm from 400 images (40 depthmaps). The before and after images show the improvements of a piecewise planar model: textureless and specular surfaces (windows) are recovered, straight lines remain straight, and 3D model complexity is reduced. Also note that the reconstruction is able to preserve non-planar surfaces as well.

Final 3D Model

3D Model with Highlighted Planes

Before

After

Before

After

Figure 7. 3D model produced by our piecewise planar and non-planar stereo algorithm from 800 images (80 depthmaps).

Figures 6-8 show the final 3D models produced by our system. Many textureless and specular surfaces that were missed in the original reconstruction were recovered by our system due to the piecewise planar model. Also, because our model enforces planes, straight lines on planar surfaces remain straight in the 3D models. Especially note that the non-planar surfaces are preserved, and are not flattened to planes as in other piecewise planar stereo methods.

5. Conclusion

Results have shown that our piecewise planar and non-planar model can successfully recover planar surfaces alongside non-planar surfaces, even in highly cluttered scenes. One of the weakness of our reconstructions is the lack of completeness. Many planar surfaces that occupy only a small part of the image are missed by our system, and other surfaces are simply not seen in any of the cameras. This can be addressed by adding a more complete set of views to the dataset. Note that these scenes are significantly more cluttered than those addressed by previous piecewise planar stereo methods.

Acknowledgements: This work was funded by the David and Lucille Packard Foundation Fellowship and the Department of Energy under Award DE-FG52-08NA28778.

Before After Highlighted Planes

Figure 8. 3D model produced by our piecewise planar and non-planar stereo algorithm from 200 images (20 depthmaps).

References

[1] S. Birchfield and C. Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *Pattern Analysis and Machine Intelligence (PAMI)*, 1998.

[2] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *Pattern Analysis and Machine Intelligence (PAMI)*, 2004.

[3] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence (PAMI)*, 2001.

[4] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision (IJCV)*, 2004.

[5] Y. Furukawa, B. Curless, S. M. Seitz, , and R. Szeliski. Manhattan-world stereo. In *Computer Vision and Pattern Recognition (CVPR)*, 2009.

[6] Y. Furukawa, B. Curless, S. M. Seitz, , and R. Szeliski. Reconstructing building interiors from images. In *International Conference on Computer Vision (ICCV)*, 2009.

[7] D. Gallup, J.-M. Frahm, P. Mordohai, Y. Qingxiong, and M. Pollefeys. Real-time plane-sweeping stereo with multiple sweeping directions. In *Computer Vision and Pattern Recognition (CVPR)*, 2007.

[8] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz. Multi-view stereo for community photo collections. In *International Conference on Computer Vision (ICCV)*, 2007.

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

[10] S. Kang, R. Szeliski, and J. Chai. Handling occlusions in dense multi-view stereo. In *Computer Vision and Pattern Recognition (CVPR)*, 2001.

[11] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *Pattern Analysis and Machine Intelligence (PAMI)*, 2002.

[12] S. Kumar and M. Hebert. Discriminative random fields: A discriminative framework for contextual interaction in classification. In *Proceedings of the Ninth IEEE International Conference on Computer Vision*, 2003.

[13] M. Pollefeys and *et al*. Detailed real-time urban 3d reconstruction from video. *Int. Journal of Computer Vision (IJCV)*, 2008.

[14] M. Pollefeys, L. V. Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual modeling with a hand-held camera. *International Journal of Computer Vision (IJCV)*, 2004.

[15] X. Ren and J. Malik. Learning a classification model for segmentation. In *International Conference on Computer Vision (ICCV)*, 2003.

[16] A. Saxena, S. H. Chung, and A. Y. Ng. 3-d depth reconstruction from a single still image. *International Journal of Computer Vision (IJCV)*, 2007.

[17] S. N. Sinha, D. Steedly, and R. Szeliski. Piecewise planar stereo for image-based rendering. In *International Conference on Computer Vision (ICCV)*, 2009.

[18] N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision (IJCV)*, 2008.

[19] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for markov random fields. In *European Conference on Computer Vision (ECCV)*, 2006.

[20] P. H. S. Torr and A. Zisserman. Mlesac: a new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding (CVIU)*, 2000.

[21] J. Xiao and L. Quan. Image-based street-side city modeling. In *SIGGRAPH Asia*, 2009.

[22] L. Zebedin, J. Bauer, K. Karner, and H. Bischof. Fusion of feature- and area-based information for urban buildings modeling from aerial imagery. In *European Conference on Computer Vision (ECCV)*, 2008.