Piecewise Planar Stereo for Image-based Rendering

Sudipta N. Sinha
Microsoft Research
sudipsin@microsoft.com

Drew Steedly
Microsoft
steedly@microsoft.com

Richard Szeliski
Microsoft Research
szeliski@microsoft.com

Abstract

We present a novel multi-view stereo method designed for image-based rendering that generates piecewise planar depth maps from an unordered collection of photographs.

First a discrete set of 3D plane candidates are computed based on a sparse point cloud of the scene (recovered by structure from motion) and sparse 3D line segments reconstructed from multiple views. Next, evidence is accumulated for each plane using 3D point and line incidence and photo-consistency cues. Finally, a piecewise planar depth map is recovered for each image by solving a multi-label Markov Random Field (MRF) optimization problem using graph-cuts. Our novel energy minimization formulation exploits high-level scene information. It incorporates geometric constraints derived from vanishing directions, enforces free space violation constraints based on ray visibility of 3D points and 3D lines and imposes smoothness priors specific to planes that intersect.

We demonstrate the effectiveness of our approach on a wide variety of outdoor and indoor datasets. The view interpolation results are perceptually pleasing, as straight lines are preserved and holes are minimized even for challenging scenes with non-Lambertian and textureless surfaces.

1. Introduction

Significant progress has recently been made in solving the problem of automatic feature matching and structure from motion robustly, which allows us to recover camera calibration and a sparse 3D structure of a scene from an unordered collection of photographs [24]. However, the problem of recovering a dense, photorealistic 3D model—the multi-view stereo problem—arguably still remains unresolved. While fully automatic stereo reconstruction systems such as [15, 13] have shown great promise, the quality of the generated models often suffer from various drawbacks. Textureless and non-Lambertian surfaces in the scene give rise to holes in the depth maps which must be interpolated in some manner. This causes flat surfaces with straight lines to appear bumpy and jaggies may also be present due to unreliable matching in the presence of non-Lambertian surfaces, occlusions *etc*. These problems frequently occur in architectural, urban scenes, or in scenes containing man-made objects where planar surfaces are quite common.

In this paper, we propose a new stereo method aimed at recovering a dense, piecewise planar reconstruction of the scene. For predominantly planar scenes, our piecewise planar depth maps are accurate, compact, and plausible enough for view interpolation between cameras with wide baselines. During view interpolation, humans are sensitive to the motion of high-contrast edges and straight lines in the scene. Our approach aims at preserving such features and minimizing parallax error, which produces perceptible ghosting. The lack of surface detail is rarely noticeable during viewpoint transitions between cameras.

1.1. Overview

Our approach starts by automatically matching features and performing structure from motion on the input photographs using an approach similar to [24], which recovers the camera calibration and produces a sparse 3D point cloud of the scene (Figure 1). This is followed by joint multi-image vanishing point detection and reconstruction of sparse 3D line segments. Next, a set of plane candidates is estimated by robustly fitting planes to the sparse set of 3D points and lines while using vanishing point cues for inferring salient plane orientations (Section 3). Piecewise planar depth maps are then recovered for each image by solving a multi-label Markov Random Field (MRF) optimization problem that involves assigning each pixel to one of the candidate planes detected earlier (Section 4). Our graph-cut based energy minimization takes into account various geometric constraints that have not previously been explored within MRF-based multi-view stereo, such as plane intersection boundaries and free-space constraints. Our piecewise planar depth maps contain planar polygonal segments that are mostly free of discretization errors that are usually present in regular grid based MRFs. The resulting lightweight and compact geometric proxies are effective for view interpolation between fairly wide baselines.

2009 IEEE 12th International Conference on Computer Vision (ICCV)
978-1-4244-4419-9/09/$25.00 ©2009 IEEE

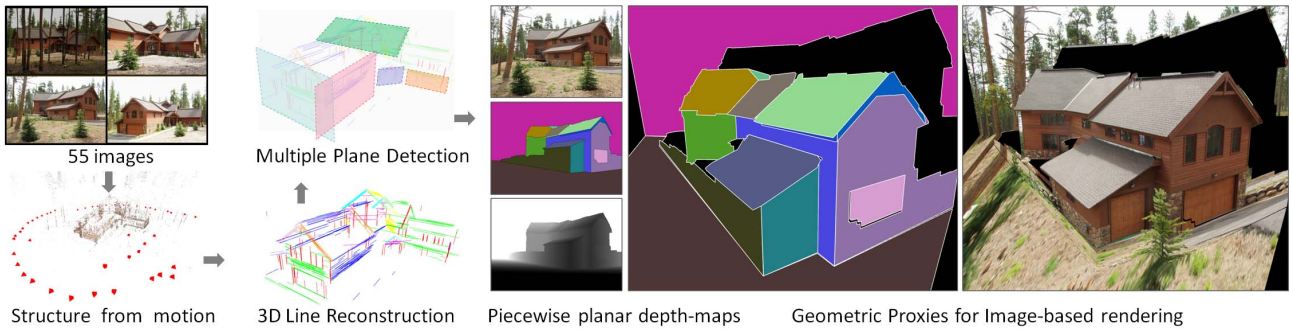


Figure 1. Overview: Multiple scene planes are robustly extracted from sparse 3D points and lines and piecewise planar depth maps are then generated by graph-cut based energy minimization.

1.2. Contributions

The main contribution of this paper is two-fold. First, we show how to recover a fairly exhaustive set of dominant scene planes based on robust plane-fitting of 3D points and lines while utilizing strong vanishing point cues to infer their orientations. These are not limited to just three orthogonal scene directions (Manhattan scenes), as is most common in the computer vision literature [2, 12, 14, 18].

Secondly, a novel Markov Random Field (MRF) formulation for generating piecewise planar depth maps is proposed. It incorporates geometric constraints derived from the 3D arrangement of multiple scene planes and vanishing directions in the scene. This allows us to enforce C_0 continuity between planes that meet and preserves straight boundaries at depth discontinuities for polygonal and rectilinear shapes. Furthermore, ray visibility of 3D points and lines [26] is used to enforce free space constraints in the energy functional. Such global information has not previously been used by state-of-the-art multi-view MRF methods.

2. Related Work

MRF-based stereo [5] traditionally focuses on first order smoothness priors by assuming fronto-parallel surfaces, which limits the quality of piecewise planar reconstructions that can be obtained. An iterative approach for fitting planes to disparity maps was proposed by [3] but this approach could only find the most dominant planes in each image. In contrast, our plane detection step is performed using global scene information recovered from all the images. This allows us to recover smaller planar structures as well as those at grazing angles to the camera.

Our work is related to the plane sweep stereo approach of [7, 25], an idea that was extended by [14] to sweeping multiple directions. The key difference is that in our MRF, we consider a small discrete set of plane hypotheses for each pixel, instead of finely discretizing the disparity space.

Various approaches for robustly detecting multiple planes in the scene [6, 10] and recoving piecewise planar reconstructions [1, 2, 11, 23, 27] have been proposed. Unfortunately, these approaches' lack of robustness and geometric detail makes them unusable for the kind of complex scenes we handle. The work of joint image triangulation [17, 19] is interesting, but depends on fairly dense point clouds recovered by structure from motion in order to work.

U-shaped canyons [8] can recover good models for urban street scenes with tall buildings, but do not generalize easily to other scenes, which can lead to lower realism in view interpolation.

The state of the art multi-view stereo methods [13, 15, 22, 29] have recently shown extremely compelling results on a wide variety of scenes. However these work at the pixel level and do not have any mechanism to exploit higher-level scene priors such as dominant orientations. These methods will typically fail to reconstruct large planes with textureless or non-Lambertian surfaces seen from a few cameras.

Our work is similar to the recent approach of [12] but goes beyond Manhattan scenes, and is related to the method of [28], which recovers an elevation map from a set of aerial images using a combination of dense depth maps, lines matched in multiple images, and an MRF based on superpixels. In comparison, our method detects planes up front and evaluates the photo-consistency cue for them directly, instead of first estimating a depth map and then fitting planes to dense points.

3. Generating Candidate Planes

The first step in our approach involves detecting all the salient scene planes using global scene information – the 3D point cloud (including their covariances and estimated normals), 3D line segments, multi-view correspondence of points and lines and vanishing directions in the scene. Fitting multiple planes to sparse data has traditionally been done using robust regression on noisy 3D points [6, 11] although some techniques have used heuristics based on 3D lines [1]. RANSAC-based [10] plane-fitting can easily extract dominant scene planes but is unreliable for detecting

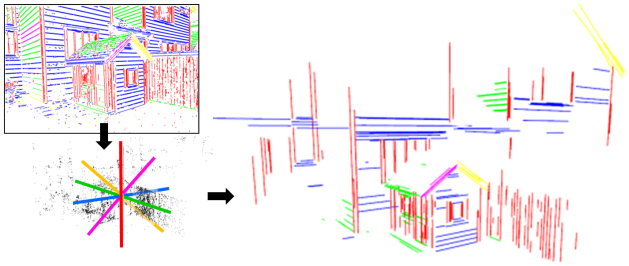
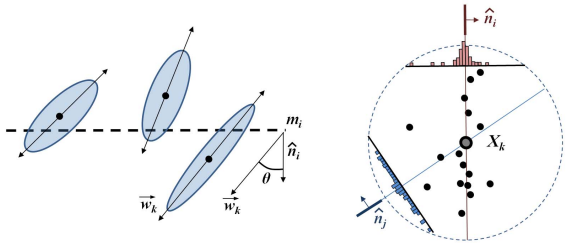


Figure 2. Single-view vanishing point detection followed by joint vanishing direction estimation in multiple views. Reconstructed 3D lines color-coded by the VDs (best seen in color).



(a) (b)

Figure 3. (a) Accounting for the depth uncertainty of 3D points is important for accurate and robust plane-detection in large scenes. (b) Estimating a likelihood distribution for a discrete set of surface normals for each 3D point using its local neighborhood.

planes with smaller support [2]. While vanishing points have been used for calibration [20], they have not been used for automatic plane extraction on large image collections.

Vanishing Directions To detect vanishing directions in the scene, 2D line segments are first detected in each image via edge detection, connected component analysis of the edgel map and a subsequent line segment growing approach. Single-view vanishing point estimation [20] is done to repeatedly search for subsets of concurrent line segments. The vanishing directions (VDs) corresponding to multiple vanishing point estimates are computed using the known camera calibration. A mean-shift clustering (on a sphere) is performed on these VD estimates and the pre-dominant clusters are retained (see Figure 2). A unique VD is computed for each cluster, while optimizing across multiple views using all the supporting 2D line segments [23].

3.1. Reconstructing 3D Lines

Our line reconstruction approach is similar to [21] and uses image appearance and epipolar constraints for matching line segments. As segment endpoints rarely correspond, care must be taken to match only the epipolar intervals that overlap. In addition, we have also found that initially constraining the search to pairs of 2D line segments supporting the same vanishing direction and with proximal matched 2D interest points improves the matching reliability considerably. Multiple match pairs are then linked into a connected component which is then verified for consensus in at least four views. The 3D line is computed by triangulating the 2D segment pairs and finding the one that has the overall minimum reprojection error within the connected component. The endpoints of the 3D line segment are recovered via interval analysis.

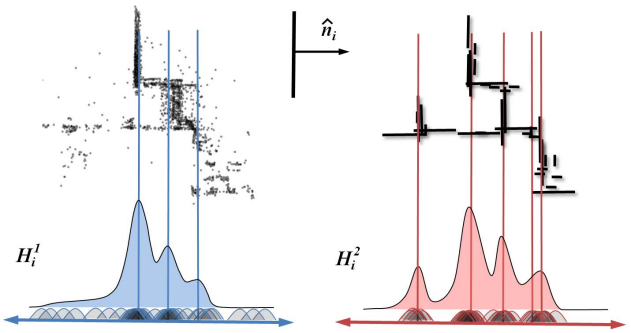
Once all VD-constrained 3D lines have been reconstructed, matching and reconstruction is done on the remaining segments. These additional lines are clustered based on their direction while accounting for their covariance (uncertainty). Our approach is able to reliably recover subtle vanishing directions, although it can produce some spurious lines. However our plane-fitting approach (described next) is robust to both point and line outliers. We inspect the set of VDs for pairs that are almost orthogonal to each other, in which case we add the third direction that completes the orthogonal triplet unless it is already present.

3.2. Plane-Fitting

Reliable vanishing directions and the presence of compatible 3D lines provide strong constraints on the orientation of planes in the scene, especially in man-made environments. We generate a set of likely plane orientations $\{\hat{n}_i\}$ by taking the cross-product of every pair of vanishing directions while treating two samples within 5^o of each other as duplicates. To ensure that we sweep dominant directions first, the set $\{n_i\}$ is sorted by saliency, where saliency is measured by the line cluster size of associated VDs.

An important aspect of fitting planes to 3D points obtained by SFM is the covariance (uncertainty in 3D point location) which tends to be an elongated ellipsoid with its major axis pointing away from the cameras it is triangulated from. The objective function that evaluates a plane hypothesis should use Mahalonobis distance in 3D (see Figure 3(a)) which is equivalent to measuring the image re-projection of the fitting error. In our approach, we approximate the uncertainty by computing for each point X_k, a vector $\vec{w}_k$ oriented along the mean viewing direction whose magnitude corresponds to a projected uncertainty of a pixel in the source images and use it while evaluating plane hypotheses.

We also compute a discrete likelihood distribution (over the set $\{n_i\}$) for the surface normal at each 3D point X_k. This is done by scoring oriented planes passing through X_k based on neighboring points around it and measuring robust variance (weighted via a Tukey function) of a 1D distribution as shown in Figure 3(b). For 3D points having dense neighborhoods that have a strong, unique peak in the likelihood distribution, we make an early commitment by assigning it a surface normal, thereby restricting which planes it can vote for during plane fitting while other points are allowed to vote more liberally. Incorporating the normals in



(a) (b)

Figure 4. (a) Histogram H_i^1 reflects the likelihood distribution for family of planes with normal $\hat{n}_i$ derived from 3D points and their normals. (b) Histogram H_i^2 reflects the same for 3D lines.

plane-fitting reduces the number of false positives generated. Photo-metric information could also be used [2] but typically requires more computation. For lines, the line direction itself is used to decide which planes it can vote for.

3.2.1 Oriented Planes

To find all scene planes that share orientation $\hat{n}_i$, we use a non-parametric approach to build two 1D histograms H_i^1 and H_i^2 that accumulates votes from 3D points and lines respectively (see Figure 4). Points and lines can vote only when their source cameras lie in front of the plane. Points whose normals were pre-estimated are pruned if their normals do not match n_i.

A 1D subspace orthogonal to n_i is chosen for projecting 3D points $\{X_k\}$ and a set of samples $\mathcal{S} = \{x_k : x_k = \hat{n}_i \cdot X_k\}$ is constructed. An adaptive Gaussian kernel size for each sample x_k is chosen as $W\vec{w}_k \cdot \hat{n}_i$ to account for the depth uncertainty bias where W (set to 5 in our experiments) controls the smoothness of histogram H_i^1. This is repeated for 3D lines in H_i^2 but the samples are further weighted by the line segment length.

Local peaks are then detected in the two histograms and plane candidates are instantiated after performing non-maximal suppression to avoid candidates too close to each other. An extra check is required for 3D lines: we only count peaks in H_i^2 that correspond to lines with multiple (≥ 2) unique directions all orthogonal to $\hat{n}_i$. Multiple parallel 3D lines are often accidentally co-planar. However non parallel lines are less likely to be coplanar unless a real plane exists in the scene.

3.2.2 Additional Planes

Explicit 3D line reconstruction sometimes fails to discover subtle vanishing directions in the scene, in which case certain key orientations will be not be swept. We address this by performing two additional stages of RANSAC-based plane fitting on the residual 3D points. First a 2-point RANSAC is done involving two points generated at random (the second one is sampled from the neighborhood of the first) and one random vanishing direction. This helps to detect scene planes with small support which are compatible with at least one of the vanishing directions detected *e.g.* for most roof planes in ordinary houses. Finally, a standard 3-point RANSAC is done while sampling the last two points randomly from the neighborhood of the first. A robust Tukey function is used to score the hypotheses.

We also compute a ground plane and back-planes for each camera [16]. An approximate *up-vector* U is estimated by solving for the direction that is orthogonal to the *side-vector* of most cameras in a robust least square sense. A plane orthongonal to U such that 95% of the SFM points lie above it becomes the ground plane which is then locally refined for a better plane-fit. The process is repeated for the backplanes using the respective camera optical axis.

4. Graph-cut based energy minimization

Given a set of scene planes $\mathcal{P}$, we independently estimate a depth map for each image by solving a pixel labeling problem using an energy minimization framework. The energy function E represents the log likelihood of the posterior probability distribution of a Markov Random Field (MRF) typically defined on the underlying pixel grid with a neighborhood system (for eg. the 4-connected neighborhood considers all vertical and horizontal neighbors). E is of the following form.

$$E(l) = \sum_p D_p(l_p) + \sum_{p,q} V_{p,q}(l_p, l_q). \qquad (1)$$

Here, l represents a labeling of the image that assigns each pixel p a label $l_p \in \mathcal{L}$ where $\mathcal{L}$ is a finite set of labels. The data term in the energy function D_p measures the cost (penalty) of p being assigned the label l_p based on measured data. The smoothness term, $V_{p,q}$ encourages a piecewise smooth labeling (*i.e.* regularizes the solution) by assigning a cost (penalty) whenever neighboring pixels p and q are assigned labels l_p and l_q respectively. When $V_{p,q}$ is a metric, the *expansion move* algorithm [5] can compute a solution that is at a provable distance from the global optimum.

In our case, the MRF is defined on the underlying pixel grid and the standard 4-connected neighborhood system is chosen. The set of labels L represents a subset of planes $\mathcal{M} \subseteq \mathcal{P}$ where $\mathcal{M} = \{m_i\},\ i = 1 \ldots N$. Each selected plane m_i must face the camera and part of it must lie within the camera's view frustum. For larger scenes with many planes, $\mathcal{M}$ can be sorted by interest point density of the 3D points. The ground and back plane are included depending on the position of the horizon in the image. The energy is minimized using graph-cuts using the max flow algorithm of [4]. The data and smoothness terms are now described.

4.1. Data Terms

Our data term D_p denotes the cost (penalty) of assigning label l_p (*i.e.* plane m_p) to pixel p. It is computed by combining the following cues – multi-view photo-consistency (denoted by D_p^1), the geometric proximity of sparse 3D points and lines obtained from plane-fitting (denoted by D_p^2), and free space violations derived from the ray visibility of the fitted 3D points and planes (denoted by D_p^3). While D_p^1 is evaluated densely at each pixel, D_p^2 and D_p^3 impose sparse constraints at specific pixels in the image. The final data term is simply $D_p(l_p) = D_p^1(l_p) + D_p^2(l_p) + D_p^3(l_p)$.

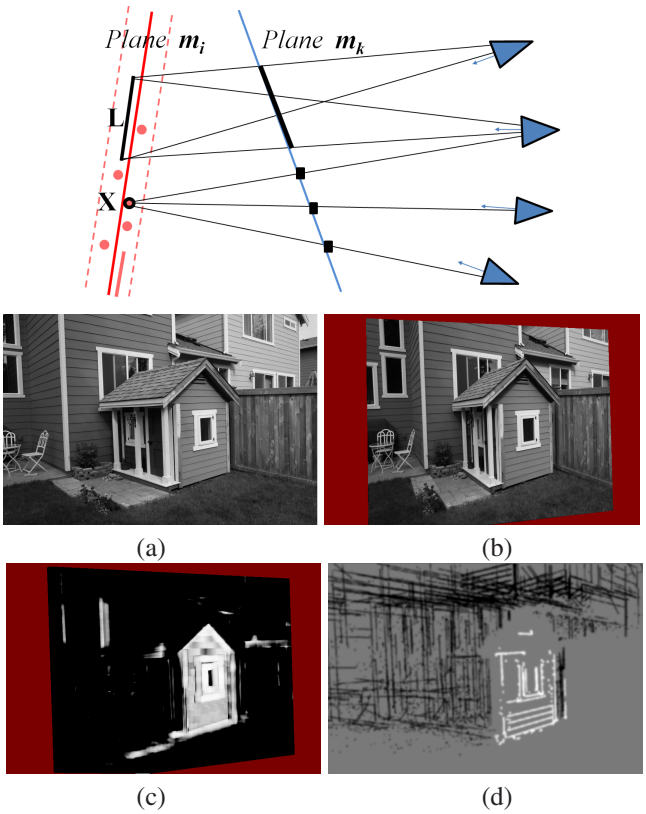


Figure 5. [Top] A 3D point X and line segment L are inliers to plane m_i while their ray-visibility imposes free-space violation penalties for plane m_k. (a,b) Image pair registered via plane induced homography. (c) Photo-consistency measure: White (black) denotes high (low) scores while unknown pixels are red. (d) Free space violation penalties (black) and inlier constraints (white).

Photo-consistency cue: We first select k (≤ 10) neighboring views $\{I_j'\}$ for the reference image I_r using a view selection strategy similar to [15]. Next, homographies induced by the scene plane m_p are used to warp each I_j' into the reference view after suitable prefiltering (see Figure 5(a,b)). The warped images are denoted by $\{I_j''\}$. Normalized cross correlation (NCC) is used to measure similarity of $\rho \times \rho$ patches (we set $\rho = 9$) at each pixel p in I_r and I_j'' which are denoted by $w_r(p)$ and $w_j(p)$ respectively. To account for error in the plane estimate or depth offsets from the plane, we search for the best match over a small disparity range d (which corresponds to a slab of thickness δ_p about m_p). For efficiency, each pair (I_r,I_j'') is rectified to allow a 1D sliding window search.

Each score $M_j(p)$ is set to $max(\{NCC(w_r(p),w_j(q)\})$ where $q = (p - d, p + d)$ is the integer disparity range. Finally we combine the similarity scores from all neighbors while treating occlusion as outliers. The final score $\overline{M}(p)$ is the average of the best 60% scores in $\{M_j(p)\}$. The data cost is then computed as follows.

$$D_p^1(l_p) = \frac{K}{2}\left(1 - exp\left(-\frac{(1-\overline{M}(p))^2}{\sigma^2}\right)\right) \qquad (2)$$

where $\sigma = 0.8$ and $K = 300$ in all our experiments.

Known sparse depth: For each 3D point X labeled as an inlier of plane m_p, we find the closest 3D point X' on the plane. These 3D points project into the reference view at pixels q and q' respectively. $D_p^2(l_p) = max(100, d(q,q')^2)$ for all pixels p within a 3×3 window around q. This is repeated for the 3D line segments, in which case the average reprojection error of its endpoints is used.

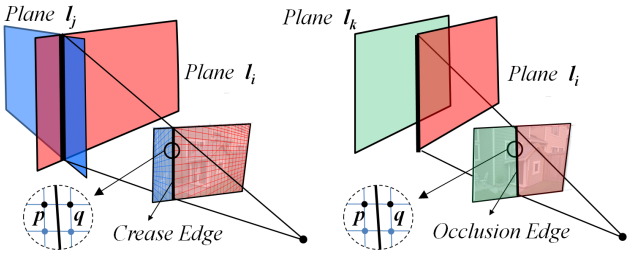
Ray Visibility: This term imposes a high penalty for assigning pixel p to m_p, whenever it violates a free space constraint. As shown in Figure 5, each 3D point X and line segment L must be visible in the views where they were matched, indicating that the corresponding ray segments must be in free space. Any plane that intersects these ray segments incurs a free space violation penalty. We compute a set of 3D points $\mathcal{F}$ on each plane m_k, where such ray-plane intersections occur. Each $X_j \in \mathcal{F}$ projects into the reference view at pixel q, and we set $D_p^3(l_p) = K$ for all pixels p in a 3×3 window around q.

4.2. Smoothness Terms

In energy minimization, the smoothness term $V_{p,q}$ is meant to encourage piecewise constant (smooth) labeling with suitable discontinuities. Unfortunately label boundaries can still suffer from discretization errors. We now describe how we choose $V_{p,q}$ so as to impose geometric constraints derived from plane arrangements and vanishing directions in order to recover more accurate label boundaries.

Piecewise planar depth maps can contain two types of discontinuities – *occlusion edges* and *crease edges* (see Figure 6(a)). Both plane labels and scene depths differ at pixels across an *occlusion edge* while only the plane label differs for pixels across a *crease edge*. A *crease edge* between a pair of plane labels coincides with the projection of the 3D intersection line of the two corresponding planes and is therefore always a straight line segment. *Occlusion edges* on the other hand can occur anywhere in the image, but for planar scenes they often coincide with visible 2D line segments. We therefore prefer label boundaries that pass

through 2D line segments detected in the image. We also use priors based on vanishing directions to impose a preference for straight line occlusion edges constrained to appropriate vanishing points in the image, whenever possible.



(a) (b)

Figure 6. (a) *Crease edges* arise from plane-plane intersections. (b) *Occlusion boundaries* can arise from salient 2D line segments and are often aligned with vanishing points in the image.

Given the set of planes $\mathcal{M}$, we compute crease lines $\{L_{ij}\}$ for all plane pairs (m_i,m_j) and select the ones that lie within the image bounds. For each of them, we find all pairs of neighboring pixels (p,q) in the image, where p and q lie on different sides of L_{ij} and accumulate all 4-tuples such as (p,q,l_i,l_j) in a set $\mathcal{S}_1$.

To impose vanishing direction (VDs) priors for plane m_i, we find VDs orthogonal to its normal. 2D line segments supporting the corresponding VPs are extended to full lines to deal with the discontinuities in 2D line detection. These lines will be preferred *occlusion edges* in the depth map.

For each such line L, we find all pairs of neighboring pixels (p,q) in the image, where p and q lie on different sides of L. This time, we back-project a ray from the midpoint of p and q and obtain a list of planes $\{m_k\}$ which lie beyond m_i when sorted w.r.t depth from the camera. For each such plane m_k, we insert a 4-tuples (p,q,l_i,l_k) in a set $\mathcal{S}_2$. The vanishing direction prior is thus imposed only for depth-discontinuities that reside on compatible scene planes.

Finally, we find pairs of neighboring pixels (p,q) that straddle the remaining 2D line segments. Such pairs are accumulated in a set $\mathcal{S}_3$. Once the sets $\mathcal{S}_1$, $\mathcal{S}_2$ and $\mathcal{S}_3$ have been constructed, we define the smoothness term as follows. When $l_p = l_q$, then we set $V_{p,q}(l_p, l_q) = 0$. When the pairwise labels l_p and l_q differ, we set it as follows.

$$V_{p,q}(l_p, l_q) = \begin{cases} \lambda_1 & \text{if } (p, q, l_p, l_q) \in \mathcal{S}_1 \\ \lambda_2 & \text{if } (p, q, l_p, l_q) \in \mathcal{S}_2 \text{ or if } (p, q) \in \mathcal{S}_3 \\ \lambda_3 & \text{otherwise} \end{cases} \tag{3}$$

Suitable values for the λ's were chosen empirically – in all our experiments with multiple datasets, λ_1, λ_2 and λ_3 are set to 1000, 1200 and 2000 respectively. However, it is possible to learn the optimal values from training data.

Name	imgs	pts	VDs	lines	planes	tris	time
PLAYHOUSE	14	4.5K	5	324	33	310	28
SHED	42	18K	6	807	45	387	87
ARCHES	25	40K	4	620	48	458	41
CASTLE	30	30K	6	1008	61	360	68
BROWNHOUSE	55	18K	12	650	127	510	145
ROOM	47	10K	6	480	72	450	130
LOBBY	18	4.8K	6	562	56	598	42

Table 1. Listed are the image count, and the number of 3D points, lines, vanishing directions and planes extracted in each dataset. Also listed are the average triangle count of the mesh proxy per image and the running time in minutes.

5. Image-based Rendering

The pixel-wise plane labeling generated by the graph-cut is converted to a proxy mesh per image. The label image is first converted to a 2D triangulation of the image using polyline simplification on the label boundaries. Using the planes for each label, the triangulation maps directly to a set of 3D triangles. This representation is ideal for view-dependent projective texture mapping [9] as the visibility computation it requires comes for free with depth maps.

During view interpolation, the two images are projectively texture-mapped onto their corresponding proxy meshes in separate rendering passes. These (off-screen) renderings (with per-pixel colors C_1 and C_2) are blended into the displayed image as in [30].

$$C = \frac{(1-\mu)C_1 + \mu C_2}{(1-\mu)\alpha_1 + \mu\alpha_2}$$

where $0 \leq \mu \leq 1$ is the blend factor and α_1 and α_2 are binary opacities (set to 1 for pixels with valid depth after warping and 0 otherwise). This enables pixels in the interpolated image, covered only by one source image to be rendered at full opacity throughout the transition. Pixels with contributions from both images are linearly cross-faded during the interpolation. Cross-fading in this manner is crucial to prevent the eye from being drawn to disoccluded regions of an image that are filled in by the other. With simple linear cross-fades, the alpha values in the rendered image would have disturbing step discontinuities at occlusion boundaries.

6. Results

We have tested our approach on various challenging interior and outdoor scenes, with specular and transparent scene structures, strong occlusions, and with cameras having large baselines. Our viewpoint transitions show significant improvement over [24] who used simple planar proxies. Table 5 summarizes some relevant details for the datasets used in our experiments. The typical image resolution for all

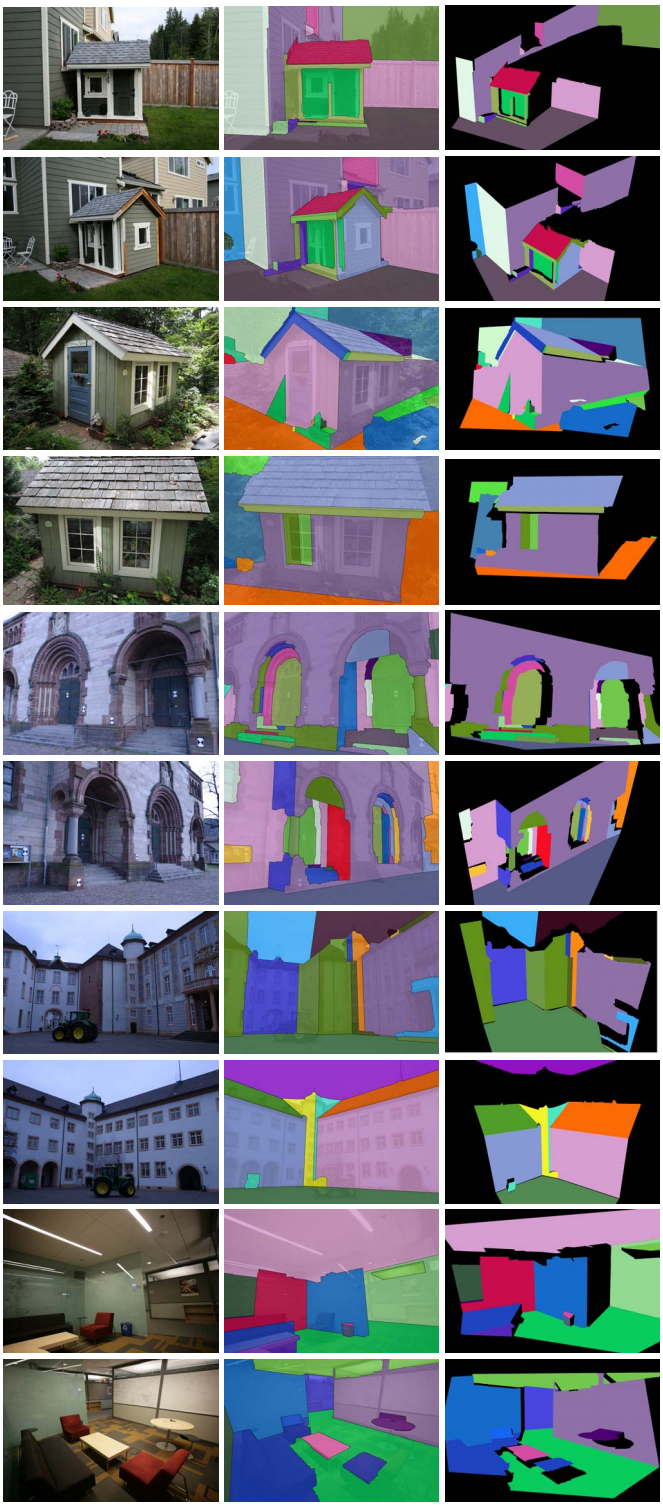


Figure 7. Results from the PLAYHOUSE, SHED, ARCHES, CASTLE and ROOM datasets. For each one of these datasets, we show two sets of the original image, the recovered depth map (colors indicate different planes) and the corresponding mesh rendered from a new viewpoint. Note the accurate polygonal boundaries especially the straight creases between adjoining polygons that meet.

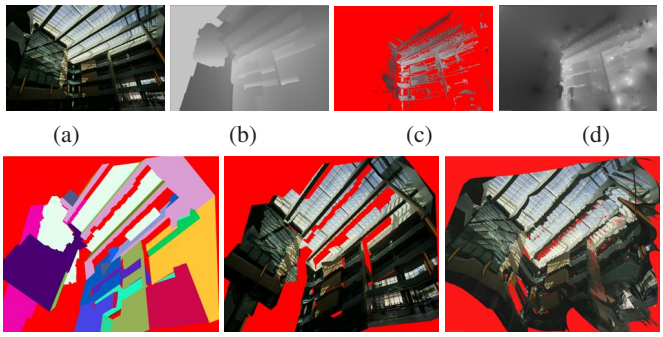


Figure 8. (a) LOBBY dataset (this image is quite under-exposed). (b) Piecewise planar depth map. (c) Semi-dense stereo [15] depth map. (d) interpolated depth map. (e) Our depth map in 3D (colors indicate planes) and (f) with texture. (g) Interpolated semi-dense stereo result with texture. The pixels in red have unknown depth.

these datasets was 2-3 Mpixels and the average running time per image varied between 2 to 3 minutes.

The ROOM dataset is challenging for multi-view stereo due to the presence of glass that exhibits both transparency and specularity in addition to having large textureless walls. Our depth maps are extremely plausible and produce pleasing viewpoint transitions even over large baselines (see the supplementary video). The LOBBY dataset contains many reflections and pronounced exposure variations. The use of global scene information enables us to recover various planes which would have been extremely challenging to reconstruct from a few neighboring images. Vanishing direction priors, lines, and crease edge constraints used in the MRF formulation produce depth maps that are higher quality than what multi-view stereo can normally compute (see Figure 8 for a comparison). Note how straight lines are preserved in our case.

For the BROWN HOUSE, we recovered twelve vanishing directions – one vertical, four for the footprint, and seven for the rooflines on the three wings of the house and the garage. This underscores that, even for relatively simple architectural scenes, we should not limit planes to the three canonical orthogonal orientations [14]. The roof planes are correctly recovered even under extreme viewing angles (see Figure 1). These would most likely be missed by simply fitting planes to the reference depth map [3, 28].

The PLAYHOUSE scene contains textureless surfaces resulting in large holes and some outliers in depth maps computed using multi-view stereo [15]. Note that horizontal lines common in man-made scenes can confuse most stereo algorithms as camera baselines are also typically horizontal. With our system, we are able to reliably extract the dominant scene planes with watertight crease edges wherever possible. On datasets such as CASTLE and ARCHES where traditional multi-view stereo would be expected to

work well, our depth maps are also plausible with good approximations for large planar surfaces. During view interpolation (see supplementary video), these approximations are almost imperceptible.

Our approach can miss a few planes if the 3D point cloud and lines are too sparse. However increasing the plane budget will make this less likely at the extra computational cost of evaluating more false positives. Currently, we do not handle occlusions in the scene and do not deal with large foreground objects, that clearly need their own proxies for compelling view interpolation. This will be a major focus of future work. We also hope to fuse piecewise planar depth maps in 3D to generate globally-consistent 3D models.

7. Conclusions

In conclusion, we have developed an automatic method for computing piecewise planar, dense depth maps to be used for image-based rendering of large unordered photo collections. Using these depth maps, we have demonstrated compelling view interpolation on a variety of man-made scenes where traditional stereo can be unreliable. Our method is based upon the idea of exploiting global scene information within a MRF stereo formulation – salient planes are first detected in a global fashion using vanishing directions, 3D points, lines and their visibility. Image pixels are then assigned to these planes using energy minimization. The ability to use vanishing direction priors and constraints from plane intersections and free space violation based on visibility of 3D points and lines makes our approach more powerful than traditional pixel-based MRF stereo.

Acknowledgements: We would like to thank C. Strecha for the ARCHES and CASTLE datasets, and Y. Boykov and V. Kolmogorov for the max-flow implementation.

References

[1] C. Baillard and A. Zisserman. A plane-sweep strategy for the 3d reconstruction of buildings from multiple images. In *ISPRS Journal*, pages 56–62, 2000.

[2] A. Bartoli. A random sampling strategy for piecewise planar scene segmentation. *In CVIU*, 105(1):42–59, 2007.

[3] S. Birchfield and C. Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *ICCV*, pages 489–495, 1999.

[4] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. *PAMI*, 26(9):1124–1137, 2004.

[5] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *ICCV*, 1:377–384, 1999.

[6] H. Chen, P. Meer, and D. E. Tyler. Robust regression for data with multiple structures. *CVPR*, 1:1069–1075, 2001.

[7] R. T. Collins. A space-sweep approach to true multi-image matching. *CVPR*, 0:358–363, 1996.

[8] N. Cornelis, K. Cornelis, and L. V. Gool. Fast compact city modeling for navigation pre-visualization. *CVPR*, 2:1339–1344, 2006.

[9] P. Debevec, Y. Yu, and G. Borshukov. Efficient view-dependent image-based rendering with projective texture-mapping. *Eurographics*, pages 105–116, 1998.

[10] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[11] F. Fraundorfer, K. Schindler, and H. Bischof. Piecewise planar scene reconstruction from sparse correspondences. *Image Vision Comput.*, 24(4):395–406, 2006.

[12] Y. Furukawa, B. Curless, S. Seitz, and R. Szeliski. Manhattan-world stereo. *In CVPR*, 2009.

[13] Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. *CVPR*, pages 1–8, 2007.

[14] D. Gallup, J.-M. Frahm, P. Mordohai, Q. Yang, and M. Pollefeys. Real-time plane-sweeping stereo with multiple sweeping directions. *CVPR*, 0:1–8, 2007.

[15] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz. Multi-view stereo for community photo collections. *ICCV*, 2:265–270, 2007.

[16] D. Hoiem, A. A. Efros, and M. Hebert. Automatic photo pop-up. *(SIGGRAPH)*, 24(3):577–584, 2005.

[17] M. Lhuillier and L. Quan. Edge-constrained joint view triangulation for image interpolation. *CVPR*, 1:218–224, 2000.

[18] B. Micusik and J. Kosecka. Piecewise planar city 3d modeling from street view. *In CVPR*, 2009.

[19] L. Quan, J. Wang, P. Tan, and L. Yuan. Image-based modeling by joint segmentation. *IJCV*, 2007.

[20] G. Schindler, P. Krishnamurthy, and F. Dellaert. Line-based structure from motion for urban environments. *3DPVT*, pages 846–853, 2006.

[21] C. Schmid and A. Zisserman. Automatic line matching across views. In *CVPR*, volume 0, pages 666–671, 1997.

[22] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. *CVPR*, 1:519–526, 2006.

[23] S. N. Sinha, D. Steedly, R. Szeliski, M. Agrawala, and M. Pollefeys. Interactive 3d architectural modeling from unordered photo collections. *Siggraph Asia*, 27(5):1–10, 2008.

[24] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. *ACM Trans. on Graphics (SIGGRAPH)*, 25(3):835–846, 2006.

[25] R. Szeliski and P. Golland. Stereo matching with transparency and matting. *IJCV*, 32(1):45–61, August 1999.

[26] C. J. Taylor. Surface reconstruction from feature based stereo. *CVPR*, 1:184, 2003.

[27] T. Werner and A. Zisserman. New techniques for automated architecture reconstruction from photographs. *ECCV*, 2:541–555, 2002.

[28] L. Zebedin, J. Bauer, K. Karner, and H. Bischof. Fusion of feature- and area-based information for urban buildings modeling from aerial imagery. *ECCV*, pages 873–886, 2008.

[29] C. L. Zitnick and S. B. Kang. Stereo for image-based rendering using image over-segmentation. *IJCV*, 75(1):49–65, 2007.

[30] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. High-quality video view interpolation using a layered representation. *SIGGRAPH*, 23(3):600–608, 2004.