Manhattan-world Stereo

Yasutaka Furukawa   Brian Curless   Steven M. Seitz
Department of Computer Science & Engineering
University of Washington, USA
{furukawa,curless,seitz}@cs.washington.edu

Richard Szeliski
Microsoft Research
Redmond, USA
szeliski@microsoft.com

Abstract

Multi-view stereo (MVS) algorithms now produce reconstructions that rival laser range scanner accuracy. However, stereo algorithms require textured surfaces, and therefore work poorly for many architectural scenes (e.g., building interiors with textureless, painted walls). This paper presents a novel MVS approach to overcome these limitations for Manhattan World *scenes, i.e., scenes that consists of piece-wise planar surfaces with dominant directions. Given a set of calibrated photographs, we first reconstruct textured regions using an existing MVS algorithm, then extract dominant plane directions, generate plane hypotheses, and recover per-view depth maps using Markov random fields. We have tested our algorithm on several datasets ranging from office interiors to outdoor buildings, and demonstrate results that outperform the current state of the art for such texture-poor scenes.*

1. Introduction

The 3D reconstruction of architectural scenes is an important research problem, with large scale efforts underway to recover models of cities at a global scale (e.g., Google Earth, Virtual Earth). Architectural scenes often exhibit strong structural regularities, including flat, texture-poor walls, sharp corners, and axis-aligned geometry, as shown in Figure 1. The presence of such structures suggests opportunities for constraining and therefore simplifying the reconstruction task. Paradoxically, however, these properties are problematic for traditional computer vision methods and greatly *complicate* the reconstruction problem. The lack of texture leads to ambiguities in matching, whereas the sharp angles and non-fronto-parallel geometry defeat the smoothness assumptions used in dense reconstruction algorithms.

In this paper, we propose a multi-view stereo (MVS) approach specifically designed to exploit properties of architectural scenes. We focus on the problem of recovering *depth maps*, as opposed to full object models. The key idea is to replace the smoothness prior used in traditional methods with priors that are more appropriate. To this end we invoke the so-called *Manhattan-world* assumption [10], which states that all surfaces in the world are aligned with three dominant directions, typically corresponding to the X, Y, and Z axes; i.e., the world is piecewise-axis-aligned-planar. We call the resulting approach *Manhattan-world stereo*. While this assumption may seem to be overly restrictive, note that *any* scene can be arbitrarily-well approximated (to first order) by axis-aligned geometry, as in the case of a high resolution voxel grid [14, 17]. While the Manhattan-world model may be reminiscent of blocks-world models from the 70's and 80's, we demonstrate state-of-the-art results on very complex environments.

Figure 1. Increasingly ubiquitous on the Internet are images of architectural scenes with texture-poor but highly structured surfaces.

Our approach, within the constrained space of Manhattan-world scenes, offers the following advantages: 1) it is remarkably robust to lack of texture, and able to model flat painted walls, and 2) it produces remarkably clean, simple models as outputs. Our approach operates as follows. We identify dominant orientations in the scene, as well as a set of candidate planes on which most of the geometry lies. These steps are enabled by first running an existing MVS method to reconstruct the portion of the scene that contains texture, and analyzing the recovered geometry. We then recover a depth map for each image by assigning one of the candidate planes to each pixel in the image. This step is posed as a Markov random field (MRF) and solved with graph cuts [4, 5, 13] (Fig. 2).

1.1. Related work

Our work builds upon a long tradition of piecewise-planar stereo, beginning with the seminal work of Wang

978-1-4244-3991-1/09/$25.00 ©2009 IEEE

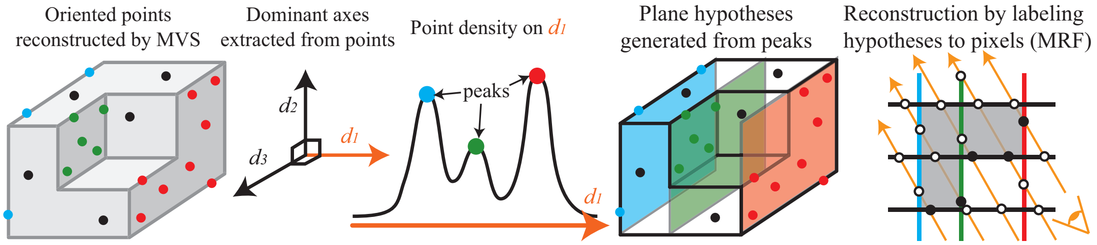
Oriented points reconstructed by MVS
Dominant axes extracted from points
Point density on d_1
Plane hypotheses generated from peaks
Reconstruction by labeling hypotheses to pixels (MRF)
d_2
d_1
d_3
peaks
d_1

Figure 2. Our reconstruction pipeline. From a set of input images, an MVS algorithm reconstructs oriented points. We estimate dominant axes d_1,d_2,d_3. Hypothesis planes are found by finding point density peaks along each axis d_i. These planes are then used as per-pixels labels in an MRF.

and Adelson on layered motion models [20]. Several authors, including Baker *et al.* [1], Birchfield and Tomasi [3], and Tao *et al.* [19], have specialized the 2D affine motion models first suggested by Wang and Adelson to the rigid multi-view stereo setting. What all these algorithms have in common is that they alternate between assigning pixels to 3D planes and refining the plane equations. In all of these approaches, the scene is treated as a collection of simple primitives. Missing in these models, however, is a model of structural *relations* between these primitives that govern how they meet and combine to form more complex scenes. A key innovation in our work is to incorporate consistency constraints on how planes meet to ensure valid surfaces, and to exploit image lines as cues for crease junctions. Another departure from [1, 3, 19, 20] is that we leverage a state-of-the-art multi-view stereo algorithm to derive plane equations and data terms, rather than directly optimize photoconsistency (appearance similarity); photoconsistency can perform poorly in wide baseline settings or in the presence of occlusions.

Another line of related research uses dominant plane orientations in outdoor architectural models to perform plane sweep stereo reconstruction. Notable examples are the work of Coorg and Teller [8], Werner and Zisserman [21], and Pollefeys *et al.* [15]. These approaches first estimate the gravity (up) vector and then find one or two dominant plane directions orthogonal to this vector using low-level cues such as reconstructed 3D points or lines. They then sweep families of planes through the scene [6, 16] and measure the photoconsistency or correlation at each pixel in order to estimate depth maps. There also exist approaches specifically designed for architectural scenes. Cornelis *et al.* [9] estimate ruled vertical facades in urban street scenes by correlating complete vertical scanlines in images. Barinova *et al.* [2] also use vertical facades to reconstruct city building models from a single image. However, these approaches that estimate vertical facades cannot handle more complex scenes consisting of mixed vertical and horizontal planes. In contrast, our approach uses robust multi-view stereo correlation scores to measure the likelihood of a given pixel to lie on a plane hypothesis, and uses a novel MRF to interpolate these sparse measurements to dense depth maps. We demonstrate good reconstruction results on challenging complex indoor scenes with many small axis-aligned surfaces such as tables and appliances. Zebedin *et al.* [22] also use an MRF to reconstruct building models, where they segment out buildings from images based on a height field, a rough building mask, and 3D lines, then recover roof shapes. Their system produces impressive building models, but one important difference from our approach is that height fields (or depth maps) are given as input in their system to reconstruct a roof model, while our algorithm produces depth maps as outputs that can be used for further modeling. (See our future work in Section 5.)

2. Hypothesis planes

Rather than solve for per-pixel disparity or depth values, as is common in stereo algorithms, we instead restrict the search space to a set of axis-aligned *hypothesis planes*, and seek to assign one of these plane labels to each pixel in the image (Fig. 2). This section describes our procedure for identifying these hypothesis planes.

Given a set of calibrated photographs, the first step of our algorithm is to use publicly available MVS software [11] to reconstruct a set of oriented 3D points (positions and normals). We retain only high-confidence points in textured areas. The normals are then used to extract three dominant axes for the scene, and the positions are used to generate axis-aligned candidate planes. The candidate planes are later used as hypotheses in MRF depth-map reconstruction.

2.1. MVS preprocessing

To recover oriented points, we employ freely available, patch-based MVS software (PMVS) [11]. PMVS takes calibrated photographs and produces a set of oriented points $\{P_i\}$. Associated with each point P_i are 3D location P_i, a surface normal N_i, a set of visible images V_i, and a photometric consistency score (normalized cross correlation) $C(P_i) \in [-1, 1]$. Note that with some abuse of notation, P_i is used to denote both the oriented point as well as its 3D position coordinates.

While PMVS works well for textured regions, the output tends to be unreliable where texture is weak or the surface is far from Lambertian. Since we do not require dense coverage for generating plane hypotheses, we reconstruct and retain points conservatively. In particular, we require PMVS to recover only points observed in at least three views, and we set its initial photometric consistency threshold to 0.95 (which PMVS iteratively relaxes to 0.65). Further, to remove points in nearly textureless regions, we project each point into its visible views and reject it if the local texture variance is low in those views. More precisely, we project each point P_i into its visible images V_i and, in each image, compute the standard deviation of image intensities inside a 7 × 7 window around the projected point. If the average standard deviation (averaged over all the images in V_i) is below a threshold τ, the point is rejected. We use $\tau = 3$ for intensities in the range $[0, 255]$.

In the remainder of the paper, some of the parameters depend on a measure of the 3D sampling rate R implied by the input images. For a given MVS point P_i and one of its visible views $I \in V_i$, we compute the diameter of a sphere centered at P_i whose projected diameter in I equals the pixel spacing in I, and then weight this diameter by the dot product between the normal N_i and viewing direction to arrive at a foreshortened diameter. We set R to the average foreshortened diameter of all points projected into all their visible views in this manner.

2.2. Extracting dominant axes

Under the Manhattan-world assumption, scene structure is piecewise-axis-aligned-planar. We could require that the axes be mutually orthogonal, however, to compensate for possible errors in camera intrinsics and to handle architecture that itself is not composed of exactly orthogonal planes, we allow for some deviation from orthogonality. To estimate the axes, we employ a simple, greedy algorithm using the normal estimates N_i recovered by PMVS (See [8, 15, 21] for similar approaches). We first compute a histogram of normal directions over a unit hemisphere, subdivided into 1000 bins.[1] We then set the first dominant axis $\overrightarrow{d_1}$ to the average of the normals within the largest bin. Next, we find the largest bin within the band of bins that are in the range 80 to 100 degrees away from $\overrightarrow{d_1}$ and set the second dominant axis $\overrightarrow{d_2}$ to the average normal within that bin. Finally, we find the largest bin in the region that is in the range 80 to 100 degrees away from both $\overrightarrow{d_1}$ and $\overrightarrow{d_2}$ and set the third dominant axis $\overrightarrow{d_3}$ to the average normal within that bin. In our experiments, we typically find that the axes are within 2 degrees of perpendicular to each other.

2.3. Generating hypothesis planes

Given the dominant axes, the next step of our algorithm is to generate axis-aligned candidate planes to be used as hypotheses in the MRF optimization. Our approach is to have the positions of the MVS points vote for a set of candidate planes. For a given point P_i, a plane with normal equal to axis direction $\overrightarrow{d_k}$ and passing through P_i has an offset $\overrightarrow{d_k} \cdot P_i$; i.e., the plane equation is $\overrightarrow{d_k} \cdot X = \overrightarrow{d_k} \cdot P_i$. For each axis direction $\overrightarrow{d_k}$ we compute the set of offsets $\{\overrightarrow{d_k} \cdot P_i\}$ and perform a 1D mean shift clustering [7] to extract clusters and peaks. The candidate planes are generated at the offsets of the peaks. Some clusters may contain a small number of samples, thus providing only weak support for the corresponding hypothesis; we exclude clusters with fewer than 50 samples. The bandwidth σ of the mean shift algorithm controls how many clusters (and thus how many candidate planes) are created. In our experiments, we set σ to be either R or $2R$. (See Sect. 4 for more details on the parameter selection.)

Note that we reconstruct surfaces using oriented planes; i.e., we distinguish front and back sides of candidate planes. Thus, for each plane, we include both the plane hypothesis with surface normal pointing along its corresponding dominant axis, and the same geometric plane with normal facing in the opposite direction.

3. Reconstruction

Given a set $H = \{H^1, H^2, \cdots\}$ of plane hypotheses, we seek to recover a depth map for image I_t (referred to as a *target image*) by assigning one of the plane hypotheses to each pixel. We formulate this problem as an MRF and solve it with graph cuts [4, 5, 13].

The energy E to be minimized is the sum of a per-pixel data term $E_d(h_p)$ and pairwise smoothness term $E_s(h_p, h_q)$:

$$E = \sum_p E_d(h_p) + \lambda \sum_{\{p,q\} \in \mathcal{N}(p)} E_s(h_p, h_q), \qquad (1)$$

where h_p is a hypothesis assigned to pixel p, and $\mathcal{N}(p)$ denotes pairs of neighboring pixels in a standard 4-connected neighborhood around p. λ is a scaling factor for the smoothness term. (See Table. 1 for the choice of this parameter for each dataset.) Note that we do not consider plane hypotheses which are back-facing to the target image's center of projection.

3.1. Data term

The data term $E_d(h_p)$ measures visibility conflicts between a plane hypothesis at a pixel and all of the points $\{P_i\}$ reconstructed by PMVS. We start with some notational preliminaries. Let X_p^l denote the 3D point reconstructed for

[1] A hemisphere is the voting space instead of a sphere, because dominant axes rather than (signed) directions are extracted in this step.

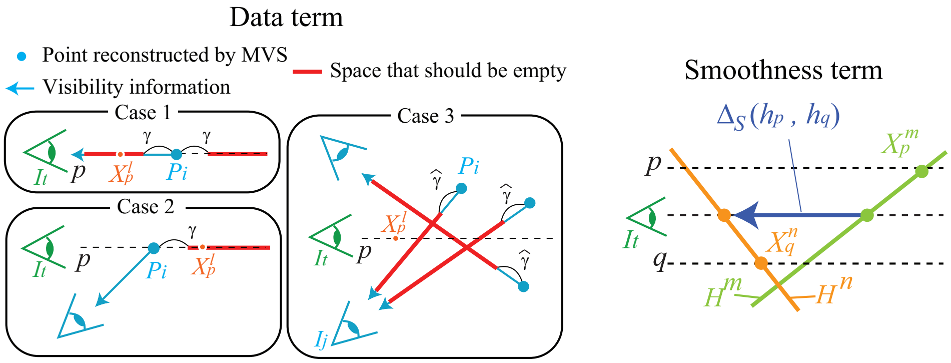

Figure 3. Data term measures visibility conflicts between a plane hypothesis at a pixel and all the reconstructed points $\{P_i\}$. There are three different cases in which the visibility conflict occurs. The smoothness term in this figure measures the penalty of assigning a hypothesis H^n to pixel q, and a hypothesis H^m to pixel p. See the text for details.

pixel p when H^l is assigned to p, i.e., the intersection between a viewing ray passing through p and the hypothesis plane H^l. We define $\hat{\pi}_j(P)$ as the projection of a point P into image I_j, rounded to the nearest pixel coordinate in I_j. Finally, we define the depth difference between two points P and Q observed in image I_j with optical center O_j as:

$$\Delta_d^j(P,Q) = (Q - P) \cdot \frac{O_j - P}{||O_j - P||}. \tag{2}$$

$\Delta_d^j(P,Q)$ can be interpreted as the signed distance of Q from the plane passing through P with normal pointing from P to O_j, where positive values indicate Q is closer than P is to O_j.

A pixel hypothesis h_p is considered to be in visibility conflict with an MVS point P_i under any one of the three following cases (illustrated in Figure 3):

Case 1. If P_i is visible in image I_t, the hypothesized point X_p^l should not be in front of P_i (since it would occlude it) and should not be behind P_i (since it would be occluded). For each P_i with $I_t \in V_i$, we first determine if $\hat{\pi}_t(P_i) = p$. If so, we declare h_p to be in conflict with P_i if $|\Delta_d^t(P_i, X_p^l)| > \gamma$, where γ is a parameter that determines the width of the no-conflict region along the ray to P_i, and is set to be $10R$ in our experiments.[2]

Case 2. If P_i is *not* visible in image I_t, X_p^l should not be behind P_i, since it would be occluded. Thus, for each P_i with $I_t \notin V_i$ and $\hat{\pi}_t(P_i) = p$, we declare h_p to be in conflict with P_i if $\Delta_d^t(P_i, X_p^l) > \gamma$.

Case 3. For any view I_j that sees P_i, not including the target view, the space in front of P_i on the line of sight to I_j should be empty. Thus, for each P_i and for each view $I_j \in V_i, I_j \neq I_t$, we first check to see if P_i and X_p^l project to the same pixel in I_j, i.e., $\hat{\pi}_j(P_i) = \hat{\pi}_j(X_p^l)$. If so, we declare h_p to be in conflict with P_i if $\Delta_d^j(P_i, X_p^l) < -\hat{\gamma}_{i,j}$ with respect to any view I_j. In this case, we employ a modified distance threshold, $\hat{\gamma}_{i,j} = \gamma / |N_{h_p} \cdot r_j(P_i)|$, where N_{h_p} is the normal to the plane corresponding to h_p, and $r_j(P_i)$ is the normalized viewing ray direction from I_j to P_i. [3]

Now, given a P_i and hypothesis h_p, we set the contribution of P_i to the data term as follows:

$$E_d^i(h_p) = \begin{cases} \max(0, C(P_i) - 0.7) & \text{if } h_p \text{ conflicts with } P_i \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where $C(P_i)$ is the photometric consistency score of P_i reported by PMVS. Note that the penalty is automatically zero if $C(P_i)$ is less than 0.7. Finally, the data term for h_p is given as follows, where 0.5 is the upper-bound, imposed for robustness:

$$E_d(h_p) = \min(0.5, \sum_i E_d^i(h_p)). \tag{4}$$

3.2. Smoothness term

The smoothness term $E_s(h_p, h_q)$ enforces spatial consistency and is 0 if $h_p = h_q$. Otherwise, we seek a smoothness function that penalizes inconsistent plane neighbors, except when evidence suggests that such inconsistency is reasonable (e.g., at a depth discontinuity).

3.2.1 Plane consistency

We score plane consistency $\Delta_s(h_p, h_q)$ by extrapolating the hypothesis planes corresponding to h_p and h_q and measuring their disagreement along the line of sight between p and q. In particular, $\Delta_s(h_p, h_q)$ is the (unsigned) distance between candidate planes measured along the viewing ray that

[2] $10R$ approximately corresponds to ten times the pixel spacing on the input images. This large margin is used in our work in order to handle erroneous points in texture-poor regions and/or compensate for possible errors in camera calibration.

[3] The modification of the threshold is necessary, because the visibility information becomes unreliable when the corresponding visible ray is nearly parallel to both the image plane of I_t and the plane hypothesis.

passes through the midpoint between p and q. [4] Large values of $\Delta_s(h_p, h_q)$ indicate inconsistent neighboring planes.

3.2.2 Exploiting dominant lines

When two dominant planes meet in a Manhattan-world scene, the resulting junction generates a crease line in the image (referred to as a *dominant line*) that is aligned with one of the vanishing points (Figure 4). Such dominant lines are very strong image cues which we can exploit as structural constraints on the depth map.

Our procedure for identifying dominant lines is described as follows. Given an image I, we know that the projection of all dominant lines parallel to dominant direction $\overrightarrow{d_k}$ pass through vanishing point v_k. Thus, for a given pixel p, the projection of any such dominant line observed at p must pass through p and v_k and therefore has orientation $\overrightarrow{l_k} = v_k - p_k$ in the image plane. Thus, we seek an edge filter that strongly prefers an edge aligned with $\overrightarrow{l_k}$, i.e., with gradient along $\overrightarrow{l_k^\perp}$, the direction perpendicular to $\overrightarrow{l_k}$. We measure the strength of an edge along $\overrightarrow{l_k}$ as:

$$e_k(p) = \frac{\Sigma_{p' \in w(p)} \|\nabla_{\overrightarrow{l_k^\perp}} I(p')\|}{\Sigma_{p' \in w(p)} \|\nabla_{\overrightarrow{l_k}} I(p')\|} \tag{5}$$

where $\nabla_{\overrightarrow{l_k}} I(p')$ and $\nabla_{\overrightarrow{l_k^\perp}} I(p')$ are the directional derivatives along $\overrightarrow{l_k}$ and $\overrightarrow{l_k^\perp}$, respectively, and $w(p)$ is a rectangular window centered at p with axes along $\overrightarrow{l_k^\perp}$ and $\overrightarrow{l_k}$. [5] Intuitively, $e_k(p)$ measures the aggregate edge orientation (or the tangent of that orientation) in a neighborhood around p. Note that due to the absolute values of directional derivatives, an aligned edge that exhibits just a rise in intensity and an aligned edge that both rises and falls in intensity will both give a strong response. We have observed both in corners of rooms and corners of buildings in our experiments. In addition, the ratio computation means that weak but consistent, aligned edges will still give strong responses.

To allow for orientation discontinuities, we modulate smoothness as follows:

$$s(p) = \begin{cases} 0.01 & \text{if } \max(e_1(p), e_2(p), e_3(p)) > \beta \\ 1 & \text{otherwise} \end{cases} \tag{6}$$

Thus, if an edge response is sufficiently strong for any orientation, then the smoothness weight is low (allowing a plane discontinuity to occur). We choose $\beta = 2$ in our experiments, which roughly corresponds to an edge within a patch being within 25 degrees of any dominant line direction. Note that the smoothness weight is set to a value slightly larger than zero; this is necessary to constrain the optimization at pixels with zero data term contribution.

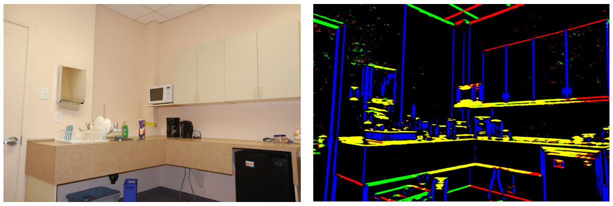

Figure 4. Input image and extracted dominant lines, used as cues for the meeting of two surfaces. The red, green and blue components in the right figure shows the results of edge detection along the three dominant directions, respectively. Note that yellow indicates ambiguity between the red and green directions.

Table 1. Characteristics of the datasets. See text for details.

	kitchen	*office*	*atrium*	*hall-1*	*hall-2*
N_c	22	54	34	11	61
N_r	0.1M	0.1M	0.1M	0.1M	0.1M
N_p	548162	449476	235705	154750	647091
N_h	227	370	316	168	350
λ	0.2	0.4	0.4	0.4	0.4
σ	$2R$	$2R$	R	R	$2R$
T_1	44	21	13	3	49
T_2	2	3	2	1	5
T_3	2.2	3.5	3.0	1.9	8.0

Putting together the smoothness components in this section, we now give the expression for the smoothness term between two pixels:

$$E_s(h_p, h_q) = \min(10, s(p)\frac{\Delta_s(h_p, h_q)}{R}) \tag{7}$$

Note that the plane depth inconsistency is scored relative to the scene sampling rate, and the function is again truncated at 10 for robustness.

To optimize the MRF, we employ the α-expansion algorithm to minimize the energy [4, 5, 13] (three iterations are sufficient). A depth map is computed for each image.

4. Experimental Results

We have tested our algorithm on five real datasets, where sample input images are shown on the left side of Fig. 6. All datasets contain one or more structures – e.g., poorly textured walls, non-lambertian surfaces, sharp corners – that are challenging for standard stereo and MVS approaches. The camera parameters for each dataset were recovered using publicly available structure-from-motion (SfM) software [18].

Table 1 summarizes some characteristics of the datasets, along with the choice of the parameters in our algorithm.

[4] The distance between X_p^m and X_q^n may be a more natural smoothness penalty, but this function is not sub-modular [13] and graph cuts cannot be used.

[5] $w(p)$ is not an axis aligned rectangle, and image derivatives are computed with a bilinear interpolation and finite differences. We use windows of size 7×21, elongated along the $\overrightarrow{l_k}$ direction.

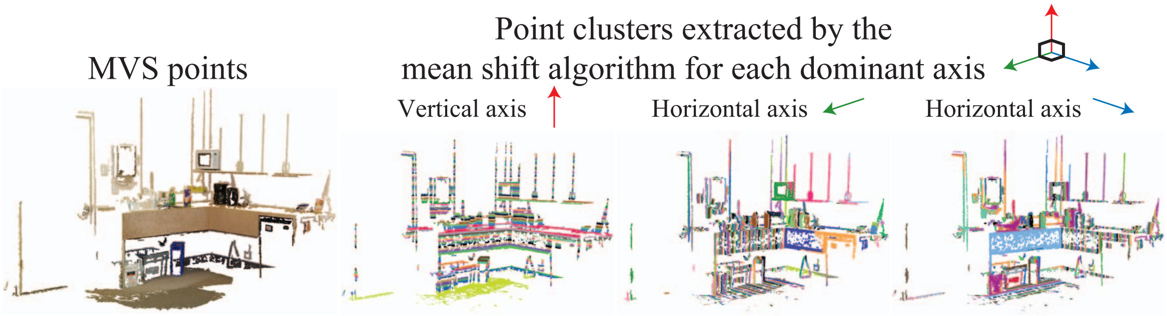
MVS points
Point clusters extracted by the
mean shift algorithm for each dominant axis
Vertical axis
Horizontal axis
Horizontal axis

Figure 5. Oriented points reconstructed by [11], and point clusters that are extracted by mean shift algorithm are shown for each of the three dominant directions. Points that belong to the same cluster, and hence, contribute to the same plane hypothesis are shown with the same color.

N_c is the number of input photographs, and N_r denotes the resolution of the input images in pixels. Since we want to reconstruct a simple, piecewise-planar structure of a scene rather than a dense 3D model, depth maps need not be high-resolution. N_p denotes the number of reconstructed oriented points, while N_h denotes the number of extracted plane hypotheses for all three directions.

There were two parameters that varied among the datasets. λ is a scalar weight associated with the smoothness term, and is set to be 0.4 except for the *kitchen* dataset, which has more complicated geometric structure with many occlusions, and hence requires a smaller smoothness penalty. σ is the mean shift bandwidth, set to either R or $2R$ based on the overall size of the structure. We have observed that for large scenes, a smaller bandwidth – and thus more plane hypotheses – is necessary. In particular, reconstructions of such scenes are more sensitive to even small errors in SfM-recovered camera parameters or in extracting the dominant axes; augmenting the MRF with more planes to choose from helps alleviate the problem.

Finally, T_1, T_2, and T_3 represent computation time in minutes, running on a dual quad-core 2.66GHz PC. T_1 is the time to run PMVS software (pre-processing). T_2 is the time for both the hypothesis generation step (Sections 2.2 and 2.3) and the edge map construction. T_3 is the running time of the depth map reconstruction process for a single target image. This process includes a step in which we pre-compute and cache the data costs for every possible hypothesis at every pixel. Thus, although the number of variables and possible labels in the MRFs are similar among all the datasets, reconstruction is relatively slow for the *hall-2* dataset, because it has many views and more visibility consistency checks in the data term.

Figure 5 shows the points reconstructed by PMVS for the *kitchen* dataset. Note that each point P_i is rendered with a color computed by taking the average of pixel colors at its image projections in all the visible images V_i. The right side of the figure shows the clusters of points extracted by the mean shift algorithm for each of the three dominant axes. Points that belong to the same cluster are rendered with the same color. As shown in the figure, almost no MVS points have been reconstructed at uniformly-colored surfaces; this dataset is challenging for standard stereo techniques. Furthermore, photographs were taken with the use of flash, which changes the shading and the shadow patterns in every image and makes the problem even harder. Our reconstruction results are given in Figure 6, with a target image shown at the left column. A reconstructed depth map is shown next, where the depth value is linearly converted to an intensity of a pixel so that the closest point has intensity 0 and the farthest point has intensity 255. A *depth normal map* is the same as a depth map except that the hue and the saturation of a color is determined by the normal direction of an assigned hypothesis plane: There are three dominant directions and each direction has the same hue and the saturation of either red, green, or blue. The right two columns show mesh models reconstructed from the depth maps, with and without texture mapping.

In Figure 7, we compare the proposed algorithm with a state of the art MVS approach, where PMVS software [11] is used to recover oriented points, which are converted into a mesh model using Poisson Surface Reconstruction software (PSR) [12]. The first row of the figure shows PSR reconstructions. PSR fills all holes with curved surfaces that do not respect the architectural structure of the scenes. PSR also generates closed surfaces, including floating disconnected component "blobs," that obscure or even encapsulate the desired structure; we show only front-facing surfaces here simply to make the renderings comprehensible. In the second row, we show PSR reconstructions with the hole-filled and spurious geometry removed using a threshold on large triangle edge lengths. The bottom two rows show that our algorithm successfully recovers plausible, flat surfaces even where there is little texture. We admit that our models are not perfect, but want to emphasize that these are very challenging datasets where standard stereo algorithms based on photometric consistency do not work well (e.g., changes of shading and shadow patterns, poorly textured

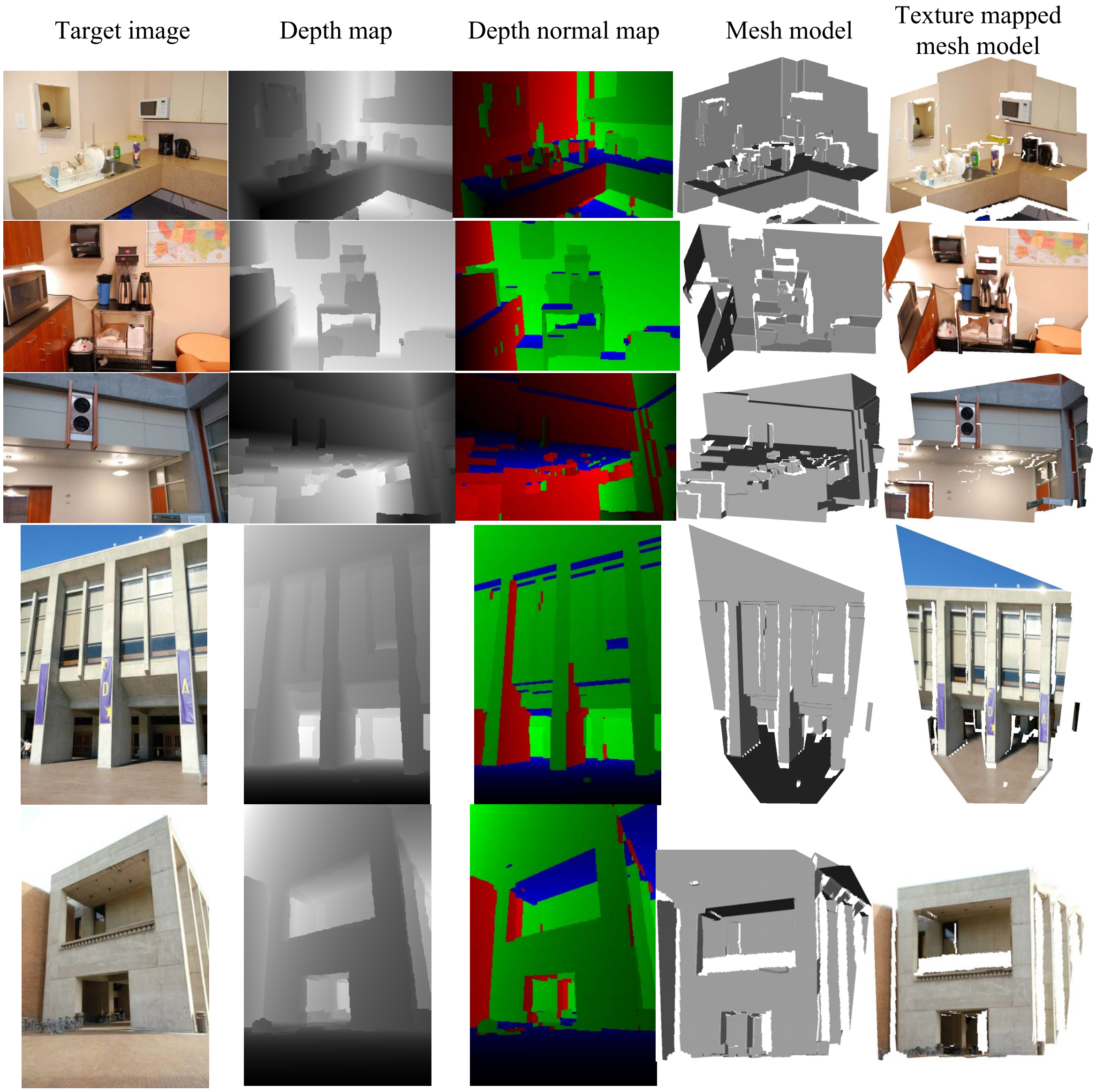

Figure 6. From left to right, a target image, a depth map, a depth normal map, and reconstructed models with and without texture mapping.

interior walls, a refrigerator and an elevator door with shiny reflections, the ground planes of outdoor scenes with bad viewing angles, etc.).

5. Conclusion

We have presented a stereo algorithm tailored to reconstruct an important class of architectural scenes, which are prevalent yet problematic for existing MVS algorithms. The key idea is to invoke a *Manhattan World* assumption, which replaces the conventional smoothness prior with a structured model of axis-aligned planes that meet in restricted ways to form complex surfaces. This approach produces remarkably clean and simple models, and performs well even in texture-poor areas of the scene.

While the focus of this paper was computing depth maps, future work should consider practical methods for merging these models into larger scenes, as existing merging methods (e.g., [12]) do not leverage the constrained structure of these scenes. It would also be interesting to explore priors for modeling a broader range of architectural scenes.

Acknowledgments: This work was supported in part by National Science Foundation grant IIS-0811878, the Office of Naval Research, the University of Washington Animation Research Labs, and Microsoft.

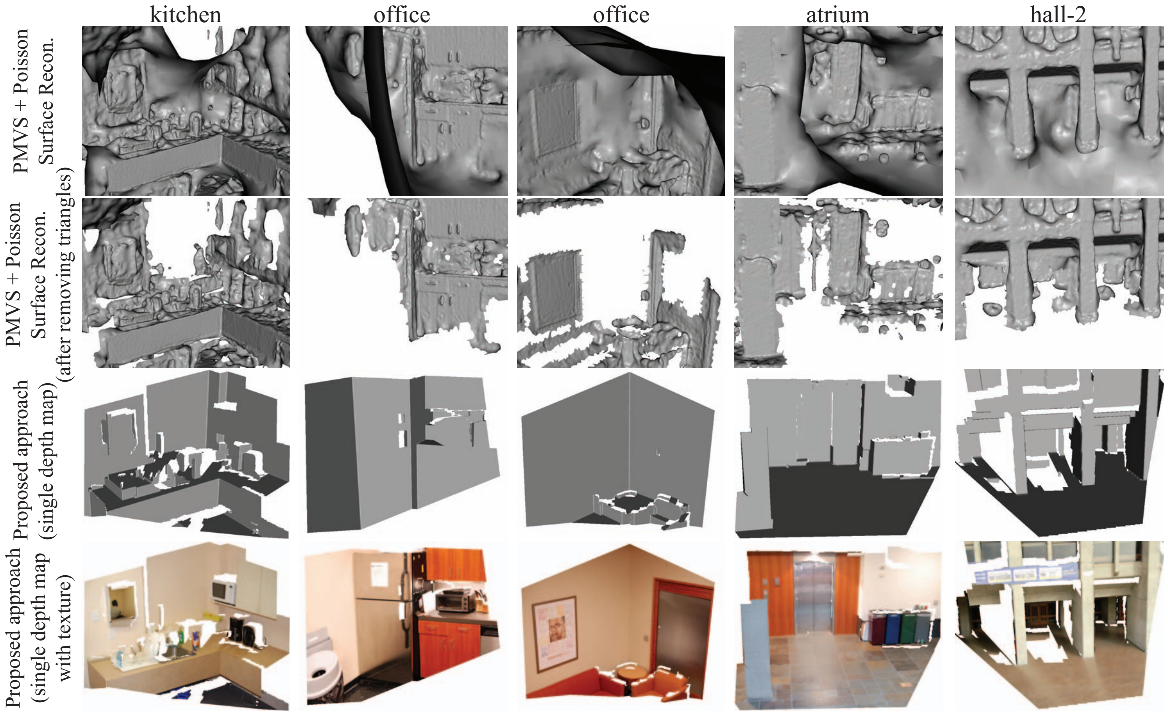
kitchen
office
office
atrium
hall-2
PMVS + Poisson Surface Recon.
PMVS + Poisson Surface Recon. (after removing triangles)
Proposed approach (single depth map)
Proposed approach (single depth map with texture)

Figure 7. Comparison with a state of the art MVS algorithm [11].

References

[1] S. Baker, R. Szeliski, and P. Anandan. A layered approach to stereo reconstruction. In *CVPR*, pages 434–441, 1998.

[2] O. Barinova, A. Yakubenko, V. Konushin, K. Lee, H. Lim, and A. Konushin. Fast automatic single-view 3-d reconstruction of urban scenes. In *ECCV*, pages 100–113, 2008.

[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*, 26:1124–1137, 2004.

[5] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23(11):1222–1239, 2001.

[6] R. T. Collins. A space-sweep approach to true multi-image matching. In *CVPR*, pages 358–363, 1996.

[7] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *PAMI*, 24(5):603–619, 2002.

[8] S. Coorg and S. Teller. Extracting textured vertical facades from controlled close-range imagery. In *CVPR*, pages 625–632, 1999.

[9] N. Cornelis, B. Leibe, K. Cornelis, and L. V. Gool. 3d urban scene modeling integrating recognition and reconstruction. *IJCV*, 78(2-3):121–141, 2008.

[10] J. M. Coughlan and A. L. Yuille. Manhattan world: Compass direction from a single image by bayesian inference. In *ICCV*, pages 941–947, 1999.

[11] Y. Furukawa and J. Ponce. PMVS. `http://www-cvr.ai.uiuc.edu/˜yfurukaw/research/pmvs`.

[12] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Symp. Geom. Proc.*, 2006.

[13] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *PAMI*, 26(2):147–159, 2004.

[14] K. Kutulakos and S. Seitz. A theory of shape by space carving. *IJCV*, 38(3):199–218, 2000.

[15] M. Pollefeys et al. Detailed real-time urban 3d reconstruction from video. *IJCV*, 78(2-3):143–167, 2008.

[16] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1-3):7–42, 2002.

[17] S. Seitz and C. Dyer. Photorealistic scene reconstruction by voxel coloring. In *CVPR*, pages 1067–1073, 1997.

[18] N. Snavely. Bundler: Structure from motion for unordered image collections. `http://phototour.cs.washington.edu/bundler`.

[19] H. Tao, H. Sawhney, and R. Kumar. A global matching framework for stereo computation. In *ICCV*, pages 532–539, 2001.

[20] J. Y. A. Wang and E. H. Adelson. Representing moving images with layers. *IEEE Transactions on Image Processing*, 3(5):625–638, 1994.

[21] T. Werner and A. Zisserman. New techniques for automated architectural reconstruction from photographs. In *ECCV*, pages 541–555, 2002.

[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 *ECCV*, pages IV: 873–886, 2008.