Piecewise Planar and Compact Floorplan Reconstruction from Images

Ricardo Cabral
Carnegie Mellon University
rscabral@cmu.edu

Yasutaka Furukawa
Washington University in St. Louis
furukawa@wustl.edu

Abstract

This paper presents a system to reconstruct piecewise planar and compact floorplans from images, which are then converted to high quality texture-mapped models for free-viewpoint visualization. There are two main challenges in image-based floorplan reconstruction. The first is the lack of 3D information that can be extracted from images by Structure from Motion and Multi-View Stereo, as indoor scenes abound with non-diffuse and homogeneous surfaces plus clutter. The second challenge is the need of a sophisticated regularization technique that enforces piecewise planarity, to suppress clutter and yield high quality texture mapped models. Our technical contributions are twofold. First, we propose a novel structure classification technique to classify each pixel to three regions (floor, ceiling, and wall), which provide 3D cues even from a single image. Second, we cast floorplan reconstruction as a shortest path problem on a specially crafted graph, which enables us to enforce piecewise planarity. Besides producing compact piecewise planar models, this formulation allows us to directly control the number of vertices (i.e., density) of the output mesh. We evaluate our system on real indoor scenes, and show that our texture mapped mesh models provide compelling free-viewpoint visualization experiences, when compared against the state-of-the-art and ground truth.

1. Introduction

Automated reconstruction of accurate 3D models from images has been one of the most fruitful outcomes of Computer Vision. Several 3D reconstruction methods have surfaced [6, 19] whose accuracy compares to laser range sensor systems at a fraction of the cost [17]. The emergence of Kinect-style depth cameras has also been a revolution for 3D Computer Vision research in recent years. Although its use is limited to short-range indoor scanning [8, 13, 20], state-of-the-art systems using these cameras produce impressive results, ranging from detailed 3D models of an office [8] to a building-scale reconstruction [20].

The majority of existing 3D reconstruction methods focus on producing more "accurate" and "dense" 3D models.

Figure 1. Our system reconstructs high quality texture mapped mesh models of cluttered indoor scenes from panorama images.

Despite their immense improvement, *perfect* results are restricted to objects or small-scale scenes, where many photos can be acquired, and surfaces are well-textured and roughly Lambertian [6, 19]. For indoor scenes, reconstructions become incomparably challenging due to violations of these conditions plus abundant clutter that is difficult to model and render. In such scenarios, reconstructions seeking for accuracy and density often yield unsatisfactory visualization [11, 21], because models are never perfect, and complex geometries induce more stitching seams, which trigger noticeable high frequency rendering artifacts.

Our primary objective is visualization, so we propose instead to seek for a 3D model that may lose certain geometric details but can provide better visualization experiences. This idea resembles the *Uncanny Valley* hypothesis for human face reconstruction, and agrees with observations from existing image-based rendering work on challenging scenes [11, 21]. [1] While it is generally not clear what kind of 3D models yield better rendering while sacrificing geometric accuracy, for indoor scene visualization there is a simple answer: *piecewise planarity*. The justification for this assumption is twofold. First, the dominant structure is the floorplan, which is usually piecewise planar; thus, enforcing piecewise planarity can suppress reconstructions of

[1] Our visualization application is free-viewpoint rendering for mapping applications, which requires much higher quality and cleanness in the 3D model, as opposed to typical view-dependent texture mapping, whose viewpoints are restricted, but works well even with corrupted geometry.

clutters in a scene, which are typically the source of rendering artifacts. Second, while regularized piecewise planar geometry also suffers from rendering artifacts, these are texture distortions rather than stitching. Our visual system is known to be very good at correcting such low-frequency distortions, while high frequency stitching artifacts are noticeable and unpleasant. Fig. 1 shows an example reconstruction of our method on a location full of clutter with challenging shapes and reflectance that are nearly impossible to model perfectly. Yet, our piecewise planar model visualizes the indoor scene effectively.

There are two challenges and technical contributions. The first challenge is the lack of 3D information from Structure from Motion (SfM) and Multi-View Stereo (MVS) due to the presence of non-diffuse and homogeneous surfaces and poor image overlaps in a confined space. We propose a structure classification technique to classify each pixel into three structure regions, *floor*, *ceiling* or *wall*, which provides complementary 3D cues to stereo even from a single image. We employ image segmentation and rely more on geometric reasoning via calibrated panoramas to improve classification, rather than resorting to appearance priors as in prior art [9]. The second challenge is the need for a sophisticated regularization technique to enforce piecewise planarity. Unfortunately, this is not attainable with most existing techniques. For example, pairwise terms in Markov Random Field (MRF) have only local influence. We cast floorplan reconstruction as a shortest path problem on a specially crafted graph, whose construction allows us to globally enforce piecewise planarity. This formulation also enables us to directly control the number of vertices in the output mesh, while globally minimizing the same objective function. Contrary to typical regularization control such as a scalar weighing term in MRF, our work provides more intuitive and powerful regularization scheme. As far as we know, we are the first to propose such framework, since existing methods typically have no control over model complexity, and are followed by a separate decimation stage.

2. Related Work

Indoor scanning has become increasingly popular in recent years. Newcombe *et al*. presented a depth sensor based 3D reconstruction system, called *KinectFusion*, for a small-scale object and a scene [13]. This work has been extended for building-scale reconstructions [20]. Albeit dense, these methods produce raw 3D measurements and are often not suitable for applications such as visualization and mapping.

To obtain compact 3D models, researchers have exploited structural regularities such as planarity or orthogonality as priors. Okorn *et al*. recovered floorplans by fitting line segments to dense point clouds projected onto a ground plane using Hough transforms [14]. Sanchez and Zakhor directly fit planes to 3D point clouds [15]. Despite their visual appeal, these models do not output a mesh model but rather a set of disconnected fragments obtained by greedy primitive fitting. Xiao and Furukawa presented a system that fits 3D geometric primitives to laser scanned points to produce a "water-tight" mesh [21]. However, it still relies on greedy primitive fitting, which requires dense point clouds and is often sensitive to termination conditions and early mistakes in the processing. Pure image-based indoor reconstruction systems also exist. Furukawa *et al*. use graph-cuts optimization in a volumetric MRF formulation [5]. However, regularization in MRFs is only based on pairwise interaction terms, and thus susceptible to noisy input data.

Interactive floorplan reconstruction has also been popular. Sankar *et al*. use smartphone sensors to reconstruct a floorplan on site [16]. Kim *et al*. [10] presented a depth camera based floorplan reconstruction system, but only handled simple uncluttered scenes. These approaches require manual input, while ours is fully automatic.

Our system makes use of structural cues directly obtained from a single image. Geometric context learning from appearance priors was proposed for outdoor scenes by Hoiem *et al*. [9] and extended to indoor scenes in [7, 12, 22**?**]. Xiong *et al*. [22] use patch similarities in images, but they restrict classification to planar patches extracted from dense laser scans. Hedau *et al*. [7] assumed a single box layout and explicitly modeled objects to reason free-space for an indoor scene. Lee *et al*. [12] proposed a *line-feature* that generates a per-pixel surface normal map from line segments under a Manhattan world assumption. Flint *et al*. [4] merged the above line-features with stereo cues and 3D points into MAP optimization. These methods are typically applied to uncluttered or single room box layouts, as opposed to the scenarios in this paper (see Fig. 1).

3. System Overview

The input to the system is a set of panorama images, where we use a standard SfM algorithm that operates on panorama images to estimate camera poses, and an MVS algorithm [6] on the original unstitched images to recover 3D points (see Fig. 2). MVS matching is a challenging problem in indoor scenes and tends to leave large reconstruction holes. We propose a single-view structure classification technique that labels pixels into three classes (floor, ceiling and wall). These can be converted into an additional point cloud, by assuming that an indoor scene is composed of vertical facades and horizontal floor and ceiling. Given a 3D point cloud, we reconstruct a 2D floorplan by solving a shortest path problem on a specially crafted graph. Finally, we extrude the estimated floorplan from the floor to the ceiling to obtain the final mesh model, and map textures.

Our core reconstruction algorithm is agnostic to the choice of 3D point acquisition technique, and depth cameras can be used as a replacement for SfM and MVS. How-

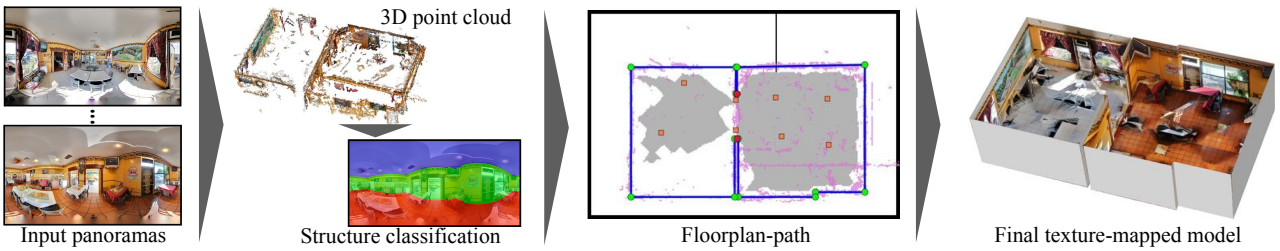


Figure 2. Overview of the proposed indoor scene reconstruction algorithm.

ever, we make the system fully image-based for several reasons. First, as a mapping application to visualize scenes, the first priority is to display high quality images to users. The minimal input for such an application is a sparse set of high quality panorama images, unattainable by a depth camera since its image quality is typically low. Second, a simple acquisition setup is easy to deploy and maintain in a production pipeline, particularly in emerging markets.

4. Preprocessing

This section provides preprocessing steps necessary for the floorplan reconstruction and structure classification.

Coordinate frame: We rotate the coordinate frame so that the XY axes are aligned with the horizontal Manhattan directions. These are determined by a multi-view vanishing point detection algorithm [18], operating on line segments extracted by Hough Transform for panorama images [1]. The Z axis is aligned with the gravity direction from SFM.

Domain and 3D evidence: The domain of the floorplan reconstruction is determined by the axis-aligned bounding box of the 3D points projected onto a ground plane, plus a constant margin of 2m. We discretize the domain by a grid of cells, where the cell size τ is set to 0.15 times the average distance of MVS points to their visible panorama centers. We collect two kinds of 3D evidence at each cell c_j. First, the evidence E_j^W that cell j belongs to a wall is calculated as the number of 3D points projected inside c_j (Fig. 3, first column). Second, the evidence E_j^F that cell j is in free-space (*i.e.*, space one can see through) is calculated as the number of times c_j is intersected by rays connecting MVS points to their visible panoramas (Fig. 3, second column).

5. Floorplan Reconstruction as Shortest Path

Our approach is similar in spirit to typical reconstruction techniques, which employ the weighted minimal surface formulation with a graph embedded in the domain [6, 19]. The key differences are the topology of the graph and the shortest path problem formulation to solve for a 2D *floorplan-path*. These enable very compact reconstructions through piecewise planarity enforcement and the ability to control the number of vertices (i.e., density) of the output. We also handle the shrinkage bias issue, common to most

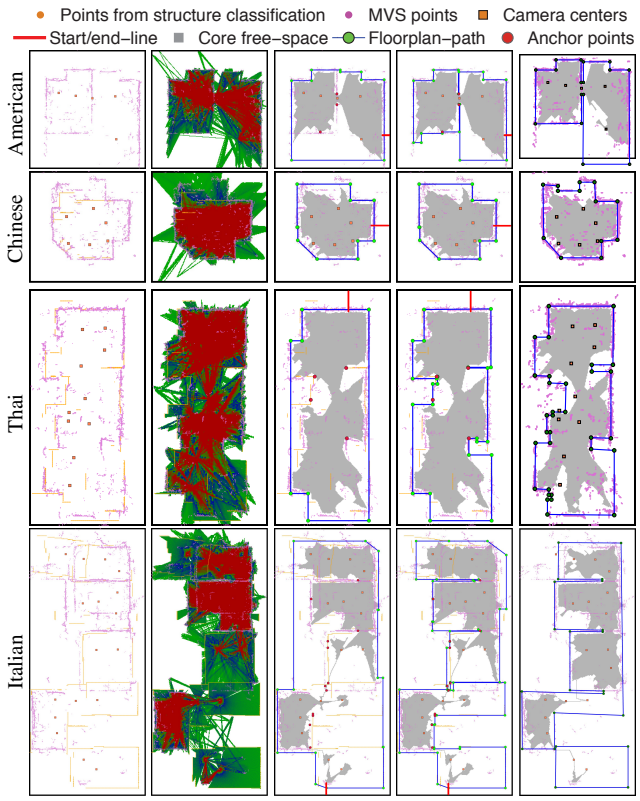


Figure 3. Our reconstruction is based on two kinds of 3D evidence: wall evidence from 3D points (first column) and free-space evidence from their visibility (second column). Red, green and blue illustrate high, medium and low confidence, respectively. Note that 3D points from structure classification form chains and look like orange lines in the figure. After identifying a high free-space evidence region as *core free-space* (grey), a shortest path problem is formulated to reconstruct a floorplan around it (third column). We overcome shrinkage bias by re-solving the problem with anchor points (fourth column). Ground truth is obtained manually by clicking room corners in images for comparison (fifth column).

reconstruction methods, by solving the shortest path problem twice, by imposing additional constraints the second time to recover thin structures. Each step is detailed below.

5.1. Graph construction

Given a domain covered by a grid of cells, we know that the floorplan-path should not go through regions with high free-space evidence. We define this *core free-space* region

as: $\{c_j | E_j^F \geq \delta_1\}$, and generate a node of the graph for each cell outside the core free-space in the domain.

Edge construction is the key of our algorithm, where an edge is added for every pair of nodes as long as it does not intersect with the core free-space. The graph has "long" edges to allow a rectangular room to be modeled by only four edges, for example. The edge weight is defined so that the path prefers long edges with high wall-evidence:

$$\sum_j \rho(E_j^W) + \alpha. \quad (1)$$

The first term penalizes going through low wall-evidence cells and is an accumulation of costs over cells along the edge, where $\rho(E_j^W)$ is an indicator function that is 1 when $E_j^W < \delta_2$ and 0 otherwise. The second term is a constant model-complexity penalty, which biases our solution towards paths with less edges. $\delta_2 = 1$ and $\alpha = 5$ are used. [2]

5.2. Initial floorplan-path computation

We seek for a floorplan-path that goes around core free-space with minimum cost. This resembles a shortest path problem, but with two problems. First, a path must be a closed loop for the floor to be well-defined. Second, an empty path with zero cost is a trivial solution. To avoid the trivial solution, we extract a *start/end-line* from the core free-space to the domain boundary (See red lines in Fig. 3, third column), and remove edges along this line. A path must start from one side and end in the other side of the line. Since we do not know where on the line, a floorplan passes through, we seek to identify such a point (dubbed *start/end-point*) together with the start/end-line as follows, from which a shortest path problem can be formulated.

Suppose we have a start/end-line, denoted as an array of cells: $\{c_1, \cdots c_{j-1}, c_j, c_{j+1}, \cdots, c_n\}$, where c_1 touches the core free-space, c_n touches the domain boundary, and c_j is the cell containing the start/end-point. If this is the right choice of line and point, then wall-evidence (1) should be high only at c_j. Therefore, the quality of the start/end-line and point can be measured as the wall evidence at c_j minus the wall evidence in the other cells, as

$$E_j^W - \sum_{|k-j|>\delta_3} E_k^W. \quad (2)$$

We used $\delta_3 = 5$ in our experiments to exclude nearby cell contributions for robustness. We limit the direction of the line to be either horizontal or vertical (two Manhattan directions) and exhaustively check all possible configurations to find the optimal one according to (2). Given a start/end-line and starting point, Dijkstra's algorithm finds the optimal path going around the core-freespace.

[2] Free-space evidence and surface normals associated with 3D points can be also used to compute edge weights. We tried various combinations, but this simple formulation produces comparable results.

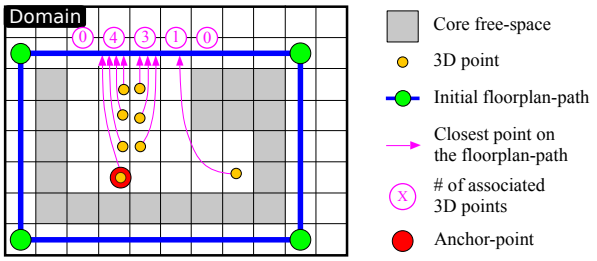


Figure 4. After the initial floorplan-path reconstruction, an anchor point is inserted at the presence of "unexplained" 3D points to avoid shrinkage bias. Start/end-line is not visualized for simplicity.

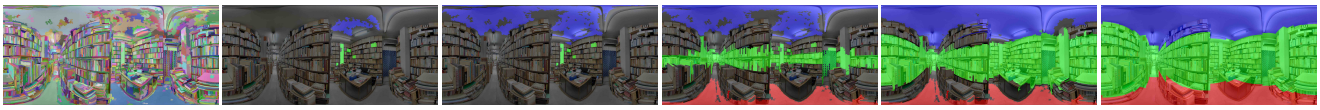
5.3. Handling shrinking bias with anchor points

The above problem formulation produces precise and compact 3D models (see Fig. 3), but suffers from a shrinkage bias as in many previous methods [6, 19]. Fig. 4 illustrates a typical scenario where a thin structure is missing. However, the existence of many "unexplained" 3D points far away from the path implies that the initial floorplan-path missed some structure. Our strategy is to identify regions with high unexplained wall-evidence and insert additional "anchor points" there. Then, we compute the shortest path between every pair of consecutive anchor points and concatenate these solutions to form a closed loop.

Let us denote the set of cells along the initial path as S. For each 3D point, we assign it to its closest cell in S in a geodesic sense through the solution space. We ignore points with distances less than 20τ (τ is the cell size) from S. The number of accumulated points along cells in S is evidence of missing structure. After smoothing out counts along S by a Gaussian with standard deviation of 6τ, we extract cells with count larger than 40. For each extracted minima, we identify its farthest associated 3D point as an anchor point.

5.4. Final floorplan-path computation

Given additional anchor points, we compute the shortest path between every pair of consecutive anchor points including the start/end-point, and concatenate them, where anchor points are ordered in the same order as the corresponding cells along the initial path. However, we would like to also compute the optimal paths with different numbers of vertices to provide floorplans of varying complexity, and take a different approach. More concretely, for every pair of consecutive anchor points, a simple dynamic programming, instead of Dijkstra's algorithm, is used to compute the optimal paths with $1, 2, \cdots \beta$ edges together with the costs. We then find the optimal combination/concatenation of these paths forming a loop with a specific total number of edges by another dynamic programming (See the supplementary material for the two dynamic programming constructions, which are straightforward and not new). β is not a sensitive parameter but should be large for the initial floorplan-path and is set to 30.



(a) Superpixels (b) MVS evidence (c) Free space evidence (d) Geometry reasoning (e) Label order (f) Dynamic program. [3]

Figure 5. Structure classification, where we classify each pixel of an image into ceiling (blue), wall (green) or floor (red).

The floorplan is triangulated to generate a floor mesh, and extruded from the floor to the ceiling to generate a facade consisting of quads. The floor and ceiling heights are estimated by a plane-sweeping MVS (with a vertical sweeping direction) by identifying the height below or above the camera whose associated photo-consistency score (normalized cross correlation) summed over the plane is maximum. Texture image for each facade quad is simply projected from the closest panorama without blending or stitching. Since in the floor triangles may be badly shaped, we grab pixel color at point-basis: for each point on the floor mesh, identify the closest panorama and collect the pixel color.

5.5. Enhancement techniques

Graph optimization: The number of edges in our graph could be potentially large, as we essentially connect every pair of nodes. In practice, a scene has a few dominant directions, so we only connect edges along directions extracted by the multi-view vanishing point detection described in Sect. 4. The number of dominant horizontal directions are typically 2, but we extract more for robustness (see Table 1). Our experiments showed this approach achieves comparable solutions to allowing all possible directions in the graph, without the additional computation complexity.

Augmenting core free-space: Due to the scarcity of MVS points, the core free-space may be defragmented into multiple components (See Fig. 3). While the floorplan-path should circumnavigate all such components, some may drop due to the shrinkage bias. Since all panorama centers must be inside the floorplan-path, we construct a minimum cost spanning tree of panorama centers, then add cells on the tree as core free-space to guarantee the condition.

6. Structure Classification

Image-based indoor modeling is still a challenging problem for multi-view techniques, such as SfM and MVS. We employ a single-view structure classification method to infer 3D cues. As we only aim to classify underlying architecture, we only assign three structural labels (*floor*, *ceiling*, and *wall*) to pixels in images (See Fig. 5). Similar to existing works [3, 4], we assume that scenes consist of vertical walls with horizontal floor and ceiling. By estimating floor and ceiling heights, the floor is related to the ceiling through a homography, and the structure classification problem is reduced to the estimation of the y-coordinate of the ceiling-wall boundary in each image column [3, 4].

Our key technical contributions lie in the use of superpixels to exploit the texture homogeneity prevalent in indoor scenes, and geometric reasoning to enforce a correct label ordering in each image column: ceiling, wall and floor from top to bottom. While lines are often effective features for existing methods [4, 12], they are far less reliable in our case due to clutter (see Figs. 1 and 6). Our classification steps are illustrated in Fig. 5 and described below.

1. Images are segmented into superpixels [2] (Fig. 5(a)).

2. Wall and free-space evidence described in Sec. 4 are used to obtain an initial set of labeled segments (Fig. 5(b,c)).[3]

3. The lower- and upper-bounds of the distance from a camera to a wall at each image column are used to infer structure labels. The upper-bound is computed from the bounding box of the domain, which gives the interval of pixels that cannot be far away, and must be a wall. The lower-bound is simply set to 0.3m, since cameras are never that close. Similar reasoning is conducted to determine pixels that must be floor or ceiling. Pixel-wise label assignments are aggregated to superpixels by a majority vote (Fig. 5(d)).

4. Structure labels are propagated by enforcing the label order (*i.e.*, ceiling, wall and floor from top to bottom in each column): Every pixel above (resp. below) the top-most ceiling (resp. bottom-most floor) pixel is also labeled as ceiling (resp. floor). Every pixel between the top-most and bottom-most wall pixels is assigned a wall label. We also exploit the homography mapping: For each pixel with a floor label, we label the corresponding pixel through homography as ceiling, if it does not already have a label. We alternate the above procedure with superpixel-wise aggregation by majority vote to propagate structure labels until convergence (Fig. 5(e)). Superpixels are eroded by 5 pixels to make this propagation stage less susceptible to noise in their shapes.

5. We employ the dynamic programming technique of [3] to globally optimize and regularize the label assignments for an entire image, while using the current labels as data prior (Fig. 5(f)). Given structure classification, we can generate a 3D point from the floor-wall boundary at each column of an image by using the floor height. We deem this point visible in the panorama that generates the point. [4]

[3] We defer details to supplementary material, as this is similar to [4].

[4] Our datasets do not provide enough lines to distinguish two horizontal manhattan directions. Thus, we distinguish only wall and floor, as opposed to leftwall, rightwall, and floor [3]. The priors for the left/right walls are computed from the same label. While [3] yields Manhattan directions for wall pixels, we discard these since we only use the ceiling-wall boundary for 3D point generation.

Figure 6. Sample input panoramas for American and Book Store.

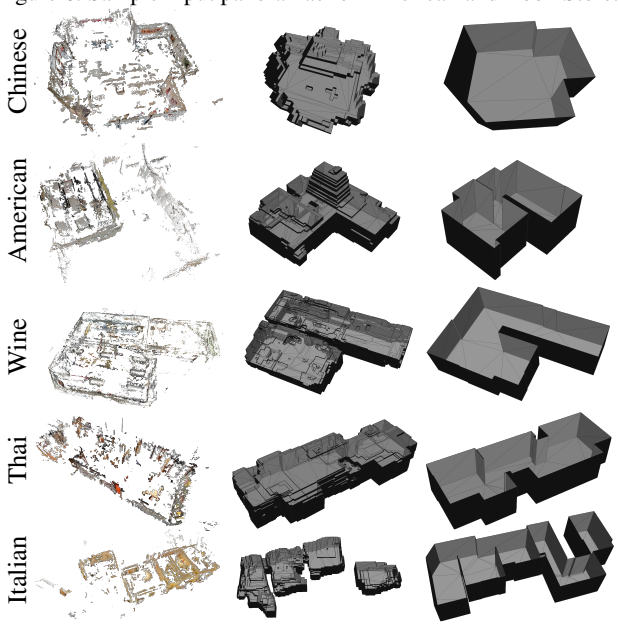


Figure 7. Input MVS point cloud (left), and comparison of 3D volumetric graph-cuts [5] (middle) against our method (right).

7. Experimental Results and Discussions

The proposed system was evaluated on a dataset of seven real locations, comprised of different kinds of restaurants and stores. Whilst small, this dataset illustrates the complexity and challenges in indoor reconstruction, as can be seen in the sample input panoramas in Figs. 1 and 6. The resolution of each panorama is 4096×2048. Statistics on the dataset, as well as parameters and running times of our algorithm are given in Table 1. This section is structured into three experiments followed by discussion to conclude the paper. First, we compare our reconstructions to state-of-the-art [5], its variants, and ground truth. Second, we illustrate the capability of our system to control the number of vertices in the output. Finally, we compare our structure classifcation technique to the line features used in [4, 12].

Comparison to ground truth and state-of-the-art: Fig. 7 shows the input MVS points at the left and our reconstructed floorplan models at the right for five of our locations. The middle column shows the floorplan models by the volumetric graph-cuts technique in [5], which extracts a surface from an axis-aligned voxel grid with MVS points. As shown, the input 3D points contain many holes. The graph-cut regularization produces noisy 3D mesh models and loses several rooms due to shrinkage bias. On the other hand, our algorithm is able to produce extremely compact and clean 3D models for the scenes. Note that in *Chinese*, a non-Manhattan diagonal wall is cleanly reconstructed.

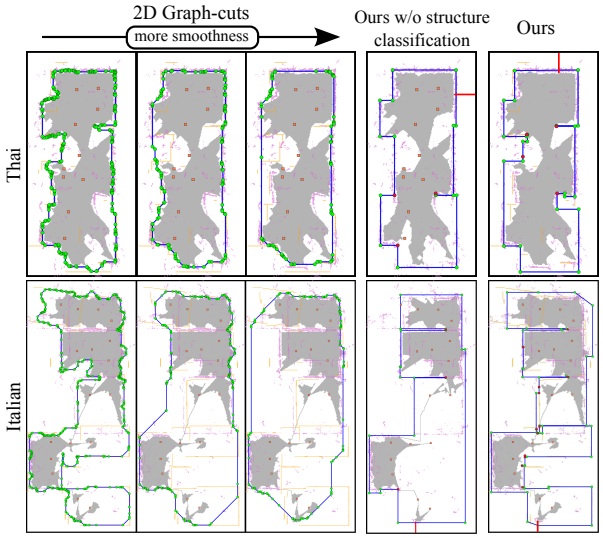


Figure 8. Left: Comparison against the 2D version of the volumetric graph-cuts [5] with several smoothness penalties. Middle: Our results without the structure classification step. Right: Our results.

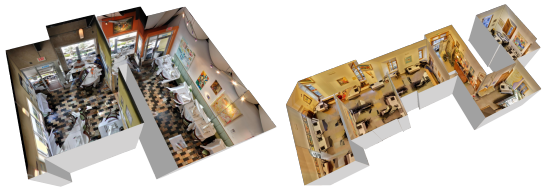


Figure 9. Final texture mapped models for American and Italian.

For fairness, we have also compared our floorplan shapes against ground truth (Fig. 3, right) and the 2D version of volumetric graph-cuts [5], using the same wall and free-space evidence in the same domain and cells (left of Fig. 8). Ground truth models are obtained by manually clicking room corners in the images. For graph-cuts, simple mesh simplification is applied to remove nodes on colinear segments to illustrate its effective resolution. The objective in [5] is the sum of data and smoothness penalties, and we varied the weight for the smoothness penalty and generated multiple results for each location. Compared to graph-cuts, our results are more compact yet capture floorplan structure accurately, in particular, thin walls and room dividers. Compared to ground truth, our results miss certain details, but mostly due to incomplete and noisy input 3D data.

One might argue we should compare against minimal-path based methods such as [**?**] instead of [5]. However, note that minimal-path (minimal-surface in 3D) methods are equivalent to graph-cuts formulations, as proven by Boykov and Kolmogorov in [**?**]. We have also conducted a quantitative evaluation on the reconstructed floorplan shapes against ground truth by computing the ratio of area incorrectly reconstructed (sum of both overestimated

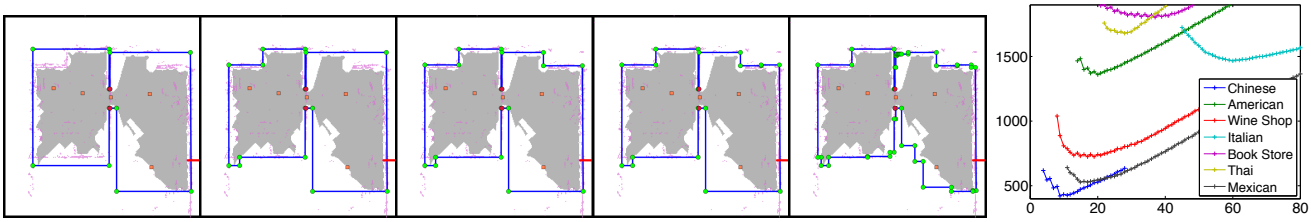


Figure 10. Left: Optimal floorplan-paths with different numbers of nodes, which are 14, 18, 20, 22 and 52 from left to right for American. Right: The cost of the optimal floorplan-path as a function of the number of nodes. The cost for Italian is divided by 3 to fit in the scale.

Figure 11. The left two columns show the extracted line segments (color represents the corresponding vanishing direction) and the structure classification result based on the line feature [12]. The right three columns show our structure classification results after initialization by MVS points, after initialization by free-space information, and the final result.

and underestimated areas) to the total ground truth area, a metric used in [16]. For each dataset, we computed a score for our algorithm, and scores for the 2D version of the graph-cuts [5], with small and large smoothness weights.

Results in Table 1 seem to suggest that 2D graph-cuts perform fairly well with a proper weight choice. However, as can be qualitatively verified from Fig. 8, several points are worth noting. First, when the smoothness weight is small, floorplans obtained with graph-cuts contain details but are extremely noisy. Furthermore, they fail to capture thin structures due to shrinkage bias. Note that areas of thin structures are very small, so these mistakes are not reflected well in the area metric. Neither is the noisiness in the floorplan shapes. Second, increasing the smoothness weight removes some noise but yields gross errors and exacerbates shrinking bias in most cases: this illustrates the limit of pairwise regularization terms in a typical MRF formulation. Furthermore, it is far from intuitive to tune such scaling parameter to control model shapes. This is opposed to our approach, which regularizes the number of edges of the model while enforcing piecewise linearity, and is able to generate proper floorplan shapes and high quality texture mapped models (Figs 1, 2 and 9) despite very noise input data. Despite all these advantages and benefits of our approach not being reflected in the area metric, it is worth noting that our scores are the best in nearly half the datasets with much fewer number of vertices, and never the worst.

Regularization control: Fig. 10 demonstrates the ability of our algorithm to control the number of vertices in the output. The optimal floorplan shape produced by our algorithm has 22 nodes in this case, but it can also generate the optimal shape with a specific number of nodes. Our algorithm succeeds in keeping proper floorplan structure even in the case of extremely low polygon counts (e.g., only 14) by enforcing piecewise planarity, which is difficult with existing methods. The cost of the floorplan-path (1) as a function of the number nodes is given at the right. Observe that, at a macro scale, each plot has one minimum. The zig-zag pattern, at a micro scale, is explained by the necessity of two nodes to create one "corner": Adding a single node simply ends up paying a model complexity penalty (α in (1)).

Structure classification: Fig. 11 provides a comparison of our superpixel classification results against the line feature of [12]. Our algorithm succeeds in generating complete and accurate classification results, starting from the incomplete label assignments from MVS points. The figure also shows the extracted line segments (straight in 3D but curved in panorama images) and the structure classification results based on the line feature of [12], a single-view structure inference technique. This line feature failed in extracting useful structure information, since it cannot deal with cluttered scenes. As a control experiment, we run our floorplan reconstruction algorithm without using the structure classification step to illustrate its effectiveness: The middle column of Fig. 8 shows that structure classification allows for recovery of thin walls and missing rooms, which would not

Table 1. Statistics of our datasets. N_p, N_l, N_d, N_i and N_f are the number of input panoramas, the average number of extracted line segments per panorama, the number of extracted horizontal dominant directions, the number of nodes in the initial floorplan-path, and the number of nodes in the final floorplan-path, respectively. R_{cell} is the resolution of the cell grid covering the domain. δ_1 is the threshold to determine the core free-space in Sect. 5, where σ is the average free-space evidence over cells with non-zero values. T_{pr}, T_{sc}, and T_{fp} are the running time for the preprocessing, structure classification and floorplan reconstruction steps, in minutes. e_{gcut-s}, e_{gcut-l} and e_{ours} are the quantitative error measures of the reconstructed floorplan shapes [16], for the graph-cuts technique with the small and large smoothness weights and our algorithm, respectively. The blue (resp. red) number is the minimum (resp. maximum) error for each dataset.

	N_p	N_l	N_d	N_i	N_f	R_{cell}	δ_1	T_{pr}	T_{sc}	T_{fp}	e_{gcut-s}	e_{gcut-l}	e_{ours}
American	5	59.7	2	10	22	720×732	0.25σ	9	7	2	0.058	0.086	0.027
Chinese	6	92.6	3	9	9	561×537	0.5σ	8	9	2	0.273	0.019	0.048
Book Store	7	145.0	2	22	36	543×1203	0.5σ	16	8	4	0.217	0.120	0.156
Mexican	8	108.4	2	8	14	516×777	0.5σ	12	12	3	0.003	0.011	0.006
Thai	10	82.2	2	8	27	525×1071	0.5σ	13	15	5	0.129	0.133	0.111
Wine Shop	16	140.6	2	14	18	753×1080	0.5σ	25	23	3	0.006	0.102	0.036
Italian	17	64.4	2	16	60	801×1845	0.5σ	18	23	81	0.241	0.487	0.125

Figure 12. A failure example due to the lack of enough 3D points.

be recovered had we used only evidence from MVS points.

Discussion: Indoor digital mapping is still in an early stage. While computer vision techniques have been extensively used for digital outdoor mapping in a global scale, most indoor locations do not have photorealistic 3D models, let alone floorplan data. Our system is a significant improvement over the state-of-the-art, but it is by no means perfect. Fig. 12 shows a typical failure example (not included in Table 1) due to the lack of enough 3D points. This example shows that compact but inaccurate reconstruction produces unpleasing texture-mapped models, and that it is essential to capture compact but also accurate 3D geometry. Our system has several limitations. First, the floorplan reconstruction algorithm assumes a fixed ring-topology and cannot handle more complicated floorplan shapes, *e.g.*, patios surrounded by indoor areas. Second, we do not model objects present in a scene, which could be important to describe and visualize the space. Despite these shortcomings, we believe this work is a foray into bringing computer vision technologies to the ultimate goal of worldwide indoor digital mapping.

Acknowledgement: The first author is supported by the Carnegie Mellon Portugal program under grant FCT/CMU/P11. This research was done when the authors were an intern and a software engineer at Google, resp..

References

[1] J. Bermudez-Cameo, L. Puig, and J. J. Guerrero. Hypercatadioptric line images for 3d orientation and image rectification. *RAS*, 60(6):755–768, 2012.

[2] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2):167–181, 2004.

[3] A. Flint, C. Mei, D. Murray, and I. Reid. A dynamic programming approach to reconstructing building interiors. In *ECCV*, 2010.

[4] A. Flint, D. Murray, and I. Reid. Manhattan scene understanding using monocular, stereo, and 3D features. In *ICCV*, 2011.

[5] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Reconstructing building interiors from images. In *ICCV*, 2009.

[6] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *PAMI*, 32(8):1362–1376, 2010.

[7] V. Hedau, D. Hoiem, and D. Forsyth. Recovering the spatial layout of cluttered rooms. In *ICCV*, 2009.

[8] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. Rgb-d mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments. *IJRR*, 2012.

[9] D. Hoiem, A. A. Efros, and M. Hebert. Geometric context from a single image. In *ICCV*, 2005.

[10] Y. M. Kim, J. Dolson, M. Sokolsky, V. Koltun, and S. Thrun. Interactive acquisition of residential floor plans. In *ICRA*, 2012.

[11] A. Kushal, B. Self, Y. Furukawa, D. Gallup, C. Hernandez, B. Curless, and S. M. Seitz. Photo tours. In *3DIMPVT*, 2012.

[12] D. C. Lee, M. Hebert, and T. Kanade. Geometric reasoning for single image structure recovery. In *CVPR*, 2009.

[13] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *ISMAR*, 2011.

[14] B. E. Okorn, X. Xiong, B. Akinci, and D. Huber. Toward automated modeling of floor plans. In *3D DPVT*, 2010.

[15] V. Sanchez and A. Zakhor. Planar 3d modeling of building interiors from point cloud data. In *ICIP*, 2012.

[16] A. Sankar and S. M. Seitz. Capturing indoor scenes with smartphones. In *UIST*, 2012.

[17] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR*, 2006.

[18] S. N. Sinha, D. Steedly, R. Szeliski, M. Agarwala, and M. Pollefeys. Interactive 3d architectural modeling from unordered photo collections. In *SIGGRAPH ASIA*, 2008.

[19] H.-H. Vu, P. Labatut, J.-P. Pons, and R. Keriven. High accuracy and visibility-consistent dense multiview stereo. *PAMI*, 34(5):889 –901, 2012.

[20] T. Whelan, M. Kaess, M. Fallon, H. Johannsson, J. Leonard, and J. McDonald. Kintinuous: Spatially extended KinectFusion. In *RSS Workshop on RGB-D*, 2012.

[21] J. Xiao and Y. Furukawa. Reconstructing the world's museums. In *ECCV*, 2012.

[22] X. Xiong and D. Huber. Using context to create semantic 3d models of indoor environments. In *BMVC*, 2010.