Planar Building Facade Segmentation and Mapping Using Appearance and Geometric Constraints

Joseph Lee, Yan Lu, and Dezhen Song

***Abstract*— Segmentation and mapping of planar building facades (PBFs) can increase a robot's ability of scene understanding and localization in urban environments which are often quasi-rectilinear and GPS-challenged. PBFs are basic components of the quasi-rectilinear environment. We propose a passive vision-based PBF segmentation and mapping algorithm by combining both appearance and geometric constraints. We propose a rectilinear index which allows us to segment out planar regions using appearance data. Then we combine geometric constraints such as reprojection errors, orientation constraints, and coplanarity constraints in an optimization process to improve the mapping of PBFs. We have implemented the algorithm and tested it in comparison with state-of-the-art. The results show that our method can reduce the angular error of scene structure by an average of 82.82%.**

I. Introduction

Vision is important for robots navigating in GPS-challenged environments. Since such environments are mostly man-made urban environments, they are often quasi-rectilinear. Planar building facades (PBFs) can be viewed as basic components of a quasi-rectilinear environment. For better scene understanding and the establishment of high-level landmarks for robust motion estimation, it is important to detect PBFs. At the first sight, PBFs may be detected using homographies between two images due to their geometric relationship. However, the plane homography can be difficult to be obtained when a PBF is far away relative to the baseline distance between the two camera views. This often happens when a monocular robot takes an image with a small step or when the baseline of a stereo camera is limited by the physical size of the robot. Also, the non-planar objects in the scene often bias the homography estimation.

We propose a method that uses both geometric and appearance information to segment PBFs and uses geometric constraints to refine the 3D PBF mapping. Fig. 1 partially illustrates our approach. The contribution of this work is twofold: 1) we propose a rectilinear index metric which allows us to identify building regions from its surroundings. Then we identify homographies using vanishing points as constraints for better planar surface detection, and 2) we combine three geometric constraints, i.e. the reprojection errors, orientation constraints, and coplanarity constraints, in an optimization process to improve the 3D mapping of the building structure. We have tested our method in physical experiments. We compare our algorithm with the state-of-the-art J-linkage-based PBF detection [1] and reprojection error-based 3D mapping. The results show that our method outperforms the J-linkage-based approach by an average of 45.27% precision increase and reduces the angular error of the reprojection error-based 3D mapping by an average of 82.82%.

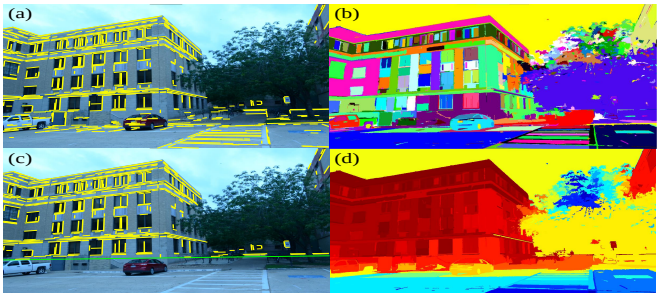


Fig. 1: Building facade segmentation using rectilinear properties (Best viewed in color). (a) Detected line segments on an image. (b) Segmented homogeneous regions (color-coded). (c) Ground line segments removed below the (green) horizontal vanishing line. (d) Heat map visualization of rectilinearity indices. Regions with warmer color indicate rectilinear regions.

II. Related Work

Reconstructing PBFs in a man-made environment has been widely studied for visualization and robot navigation. For visualization, computer graphics research has focused on reconstructing complex and accurate architectural models using polygonal meshes. This usually requires either repetitive scene scanning or range finders that can produce dense measurements with high precision. Here we do not focus on fully reconstructing the robot's environment since this is often not the main task in robot navigation.

In robot navigation, when a range finder is used, planes can be mapped by directly fitting a 3D plane model to the depth data. For example, in [2], a real-time plane segmentation algorithm is developed for noisy 3D point clouds acquired by less accurate and portable LIDAR. Recently, as RGB-D sensors became more available, a number of researchers have used these sensors to map planar surfaces in an indoor environment [3], [4], [5]. Delmerico et al. [6] present a stereo-based building facade mapping method which fits a plane model in the 3D disparity space. In their facade detection step, local surface normals are first computed for each point in 3D with a Markov random field model. Then, random sample consensus (RANSAC) [7] is used to progressively cluster these points with the same plane normal one plane at a time. Although they do not use a range finder,

This work was supported in part by the National Science Foundation under IIS-1318638.

J. Lee, Y. Lu, and D. Song are with the Department of Computer Science and Engineering, Texas A&M University, College Station, TX 77843, USA. E-mail: {jslee,ylu,dzsong}@cse.tamu.edu. D. Song is also a visiting scientist at Shenzhen Institute of Advanced Technology (SIAT), Chinese Academy of Sciences, Shenzhen, China.

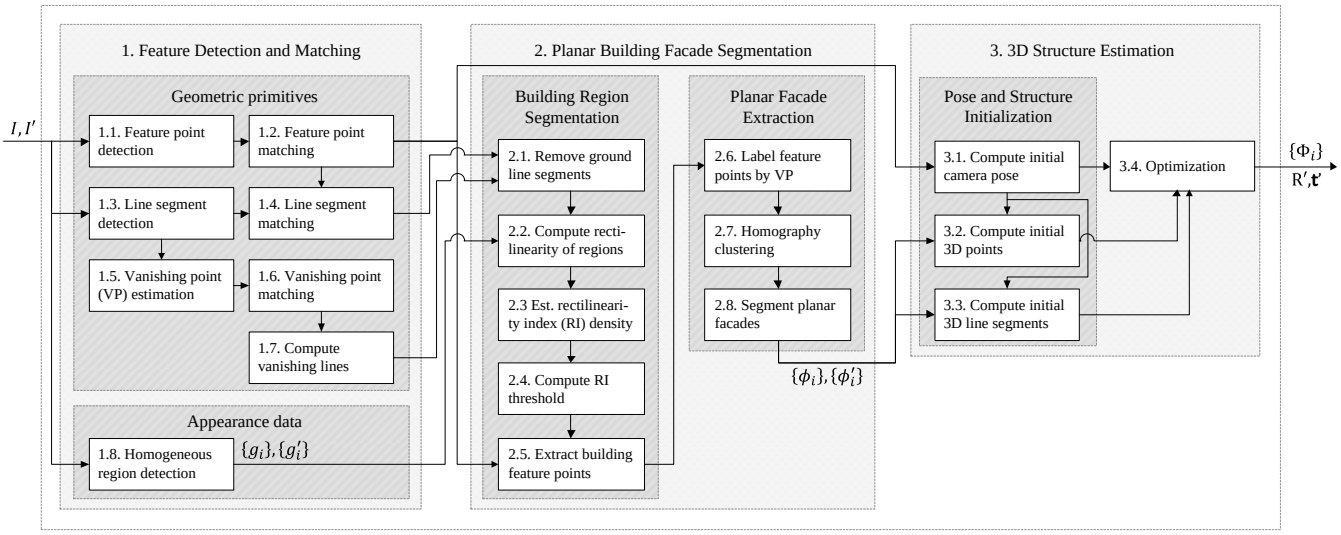


Fig. 2: System diagram

their plane mapping method is similar to the range finder-based methods in that the plane is directly fitted in the 3D space. Our work only uses a passive vision sensor which does not provide depth measures. However, this allows us to explore appearance data to segment planar surfaces in the 2D domain prior to estimation of building facades in 3D so that noises from non-planar objects can be reduced.

For vision-based navigation, detection of planes in the 2D domain precedes the 3D mapping. Plane detection in 2D is usually done by using the homography constraint. Since RANSAC is not suited to detect multiple models, Fouhey et al. [1] use an agglomerative clustering technique, called J-linkage [8], to detect multiple building facades from two images using the homography model. Baillard and Zisserman [9] propose a method that reconstructs piecewise planar building rooftops from multiple aerial images using half-planes and their associated homographies. Zhou and Li [10] develop a modified expectation-maximization (EM) algorithm that uses the homography constraint to classify ground-plane pixels and non-ground plane pixels for a mobile robot. Other existing methods use appearance information to help planar surface detection and mapping. Li and Birchfield [11] use an image-based segmentation method to detect ground planes in an indoor corridor. In their method, vertical and horizontal line segments are used to find the wall-floor boundary. Visual mapping has also been done using learned structural priors from appearance data [12]. In our work, we use both geometric and appearance information to help the homography detection process.

Previously, a building facade mapping method was proposed in [13] where they use a multi-layered feature graph [14], [15] for robust estimation of building facades. For better reconstruction results, the reconstruction process enforces geometric constraints such as parallelism and coplanarity. In this work, we focus on extracting coplanar features using appearance data and use different cost functions for the coplanarity constraint.

III. Problem Statement

A. Notations and Assumptions: We denote two input images by I and I', where the prime symbol ($'$) denotes the entities of the second image. Feature points in projective space $\mathbb{P}^2$ and $\mathbb{P}^3$ are denoted as $\mathbf{x}$ and $\mathbf{X}$, respectively. Similarly, $\mathbf{e} \in \mathbb{P}^2$ and $\mathbf{E} \in \mathbb{P}^3$ denote line-segment endpoints. We define line segments by their two endpoints, i.e. $\mathbf{s} := (\mathbf{e}_1, \mathbf{e}_2)$ and $\mathbf{S} := (\mathbf{E}_1, \mathbf{E}_2)$. A plane in $\mathbb{P}^3$ is defined by $\Pi := [\mathbf{n}^\mathsf{T}, d]^\mathsf{T}$, where $\mathbf{n}$ is the plane normal and $d/\|\mathbf{n}\|$ is the distance from the plane to the origin. Formally, we define:

Definition 1 (PBF): A *PBF* in I is defined as a 3-tuple $\phi := (X, S, G)$, where X and S denote a set of n_x feature points $\{\mathbf{x}_i\}_{i=1}^{n_x}$ and a set of n_s line segments $\{\mathbf{s}_i\}_{i=1}^{n_s}$, respectively, lying in pixel region G. In 3D world coordinates, a PBF is defined by $\Phi := (X, S, \Pi)$, where X and S denote a set of feature points $\{\mathbf{X}_i\}_{i=1}^{n_x}$ and a set of line segments $\{\mathbf{S}_i\}_{i=1}^{n_s}$, respectively, which are associated to plane Π.

We make the following assumptions in our approach.

- Lens distortion is removed from I and I'.
- The intrinsic camera parameter matrices $\mathtt{K}$ and $\mathtt{K}'$ are known from pre-calibration.
- The baseline distance is known and nonzero.

B. Problem Definition:

Problem 1: Given I and I', extract a set of n_f corresponding PBFs $\{\phi_i\}_{i=1}^{n_f}$ and $\{\phi'_i\}_{i=1}^{n_f}$ and map their 3D positions $\{\Phi_i\}_{i=1}^{n_f}$ relative to the cameras. Also, estimate the extrinsic camera parameters of rotation $\mathtt{R}'$ and translation $\mathbf{t}'$.

IV. System Design

Fig. 2 shows an overview of our system which consists of three main blocks: 1) feature detection and matching, 2) PBF segmentation, and 3) 3D structure estimation. The second and third blocks are the main contribution of this paper.

Given images I and I', the first block extracts geometric primitives, such as feature points and line segments, along with regions with homogeneous appearance. Feature points

are detected and matched using scale-invariant feature transform (SIFT) [16], and line segments detected by the Line Segment Detector [17] are matched using the method proposed by Fan et al. [18] which uses keypoint matches. From the raw line segments, vanishing points are estimated using [19]. We use only vanishing points $\mathbf{v}$ where the vanishing direction $\mathtt{K}^{-1}\mathbf{v}$ and $\mathtt{R}'\mathtt{K}'^{-1}\mathbf{v}'$ match. Then, vanishing lines are computed from each pair of vanishing points. For appearance data, we use a graph-based segmentation algorithm in [20] to extract regions with homogeneous appearance (see Fig. 1b).

The second block uses the geometric and appearance data from above to detect corresponding 2D building facades $\{\phi_i\} \leftrightarrow \{\phi_i'\}$ using a two-step approach. In the first step, rectilinear building regions are segmented out using both line segments and homogeneous appearance regions (Sec. V-A). Then, the next step detects each PBF by homography clustering using only the feature points and line segments that lie on the segmented building region (Sec. V-B).

The final block estimates the 3D structure of the building by estimating the set of PBFs $\{\Phi_i\}$. The camera pose, $\mathtt{R}'$ and $\mathbf{t}'$, and the 3D positions of the feature points and line segments are first initialized. Then, the parameters of $\{\Phi_i\}$ are further optimized using geometric constraints (Sec. VI).

V. PBF Segmentation

Now we describe how a set of PBFs $\{\phi_i\}$ are segmented from a 2D image using both geometric and appearance data. When detecting multiple PBFs using a homography-based clustering approach such as in [1], two problems can occur: 1) features from non-building objects will clutter the homographies and 2) facade boundaries become more ambiguous when the building is relatively far away compared to the baseline distance. These problems cannot be solved by merely adjusting the parameters of the clustering technique. Here, we propose a two-step approach, i.e. the building region segmentation step and the planar facade extraction step (Box 2 in Fig. 2), where the above two issues are addressed step by step. The first step extracts the target building region to suppress the non-building objects by combining geometric and appearance data. In the second step, to avoid ambiguous boundaries, homography clustering is applied to each group of feature points with the same horizontal vanishing direction.

A. Building Region Segmentation

Let $\{g_i\}_{i=1}^{n_g}$ be the set of n_g homogeneous pixel regions segmented from I using the algorithm in [20]. Since buildings are usually characterized by their rectilinear structure, we use line segments to determine which region g_i belongs to a PBF. We use the following steps (Boxes 2.1-2.5 in Fig. 2) to segment the building region.

Remove ground line segments. When an image is taken from a ground robot, most of the line segments below the horizontal vanishing line belong to the ground. Although this is not valid for near objects, in many cases, this is true for line segments on a distant building. Suppose a horizontal vanishing point $\mathbf{v}_i = [x_i, y_i, 1]^\mathsf{T}$ is counterclockwise

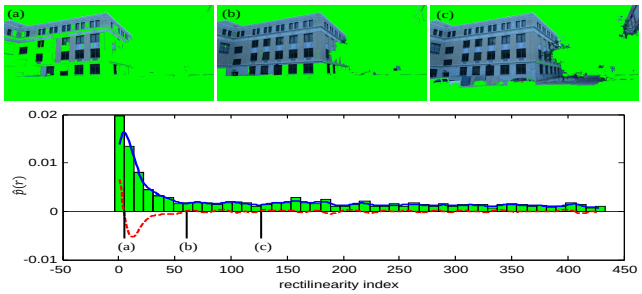


Fig. 3: RI density threshold. The solid (blue) line is the estimated RI density, and the dotted (red) line is its derivative. Segmentation results are shown for (a) below selected threshold, (b) selected threshold, and (c) above selected threshold.

from another horizontal vanishing point $\mathbf{v}_j = [x_j, y_j, 1]^\mathsf{T}$ with respect to the zenith vanishing point $\mathbf{v}_z = [x_z, y_z, 1]^\mathsf{T}$, i.e. $\begin{vmatrix} x_i - x_z & x_j - x_z \\ y_i - y_z & y_j - y_z \end{vmatrix} < 0$. Let $\mathbf{w}$ denote the vanishing line computed by $\mathbf{v}_i \times \mathbf{v}_j$. Then we remove line segments $(\mathbf{e}_1, \mathbf{e}_2)$ that satisfy the condition $\mathbf{e}_1^\mathsf{T}\mathbf{w} > 0 \vee \mathbf{e}_2^\mathsf{T}\mathbf{w} > 0$. Fig. 1c shows an example. After removing ground line segments, we use the remaining line segments to extract rectilinear regions as follows.

Compute rectilinearity of regions. If a region g_i contains a group of line segments, it is a strong cue for g_i being a rectilinear region. To measure the rectilinearity of a region we define the rectilinearity index (RI) as follows:

Definition 2 (RI): Suppose b_i is the set of boundary pixel coordinates of region g_i. The *RI* of g_i is measured by finding the nearest line segments to each pixel coordinate $\mathbf{p} \in b_i$ and computing their average distance by

$$r_i = \frac{1}{|b_i|} \sum_{\mathbf{p} \in b_i} d(\mathbf{p}, \mathbf{s}_p), \tag{1}$$

where $\mathbf{s}_p$ is the nearest line segment to $\mathbf{p}$, and $d(\mathbf{p}, \mathbf{s}_p)$ is the shortest distance from $\mathbf{p}$ to line segment $\mathbf{s}_p$.
A smaller RI indicates a higher rectilinearity. Fig. 1d shows an example of computing the RIs. If none of the regions have an RI smaller than a certain threshold, we determine that the view does not contain a distinguishable rectilinear structure.

Estimate RI density. After computing the RI for each region, we use kernel density estimation to compute the density of the RIs. The RI density can be computed by

$$\hat{p}(r) = \frac{1}{n_g h_{\text{opt}}} \sum_{i=1}^{n_g} K\left(\frac{r - r_i}{h_{\text{opt}}}\right) \tag{2}$$

where K is a normalized Gaussian kernel $K(x) = e^{-x^2/2}/\sqrt{2\pi}$ and h_{opt} is the optimum kernel bandwidth estimated using leave-one-out cross-validation. The example in Fig. 3 shows a typical shape of the density estimate when the image contains a rectilinear building structure. A peak exists where the RI is small.

Compute RI threshold. To extract regions g that belong to a PBF, we segment the peak of the RI density using a threshold. Instead of using a fixed threshold, we want to find the rectilinear regions based on the estimated distribution. Hence, we set the second zero-crossing of the first derivative of the kernel density estimate

$$\hat{p}'(r) = \frac{1}{n_g h_{\text{opt}}} \sum_{i=1}^{n_g} K'\left(\frac{r - r_i}{h_{\text{opt}}}\right) \tag{3}$$

Fig. 4: Extracting PBFs (Best viewed in color). (a) Feature points associated to vanishing direction of neighboring (color-coded) horizontal line segments. (b) Homography clustering applied to feature points with same vanishing direction. (c) Convex hull for each homography cluster.

as our threshold. An example result is shown in Fig. 3b.

Extract building feature points. After the building region is segmented, we extract the feature points that lie on the building region.

B. Planar Facade Extraction

In the second step, our goal is to detect a set of corresponding PBFs $\{\phi_i\} \leftrightarrow \{\phi_i'\}$ by clustering feature points using homographies and vanishing directions. Since a PBF should contain only one horizontal vanishing direction, the horizontal vanishing direction is an important cue to separate PBFs. Instead of applying J-linkage to the entire feature points that would result in ambiguous boundary problems, we apply J-linkage to each group of features that have the same horizontal vanishing direction. We use the following steps (Box 2.6-2.8 in Fig. 2) to separate PBFs.

Label feature points. We label each feature point $\mathbf{x}$ with a horizontal vanishing direction. Let $\mathbf{s}^h = (\mathbf{e}_1^h, \mathbf{e}_2^h)$ be a line segment associated to a horizontal vanishing point and $\mathbf{l}^h = \mathbf{e}_1^h \times \mathbf{e}_2^h$, where $\mathbf{e}_1^h$ is counterclockwise from $\mathbf{e}_2^h$ with respect to the zenith vanishing point $\mathbf{v}_z$. For each $\mathbf{s}^h$, we compute two vertical lines $\mathbf{l}_1^v = \mathbf{v}_z \times \mathbf{e}_1^h$ and $\mathbf{l}_2^v = \mathbf{v}_z \times \mathbf{e}_2^h$. Feature point $\mathbf{x}$ is then assigned with the vanishing direction of the nearest $\mathbf{s}^h$ that satisfies $\mathbf{x}^\mathsf{T}\mathbf{l}^h < 0 \wedge (\mathbf{x}^\mathsf{T}\mathbf{l}_1^v)(\mathbf{x}^\mathsf{T}\mathbf{l}_2^v) < 0$. Fig. 4a shows an example after feature points are labeled with their horizontal plane direction.

Homography clustering. After feature points are associated with a horizontal vanishing direction, we apply homography clustering using J-linkage on the feature points with the same horizontal vanishing direction. This avoids ambiguities between facade boundaries and separates the facades that are in different horizontal directions. Fig. 4b shows an example of the clustering on one horiztonal vanishing direction.

Segment planar facades. Finally, a convex hull is computed for the set of feature points that belong to the same cluster obtained in the previous step (see Fig. 4c). The pixel region in this convex hull is set to G. The feature points and line segments that lie in G are denoted as X and S, respectively.

Thus, we have the three components X, S, and G for a PBF ϕ in an image I. The algorithm for the PBF segmentation and its expected running time are summarized below. Hence, the overall computational complexity of Algorithm 1 is $O(n_x^3 + n_s(n_x + n_g) + n_g^2)$.

Algorithm 1 PBF Segmentation

Input: $\{\mathbf{x}_i\}_{i=1}^{n_x}$, $\{\mathbf{s}_i\}_{i=1}^{n_s}$, $\{g_i\}_{i=1}^{n_g}$, $\mathbf{w}$
Output: $\{\phi_i\}_{i=1}^{n_f}$

1:	remove ground line segments under $\mathbf{w}$	$\triangleright\ O(n_s)$
2:	compute RI for all $\{g_i\}$	$\triangleright\ O(n_s n_g)$
3:	estimate RI density using cross-validation	$\triangleright\ O(n_g^2)$
4:	extract building feature points	$\triangleright\ O(n_x)$
5:	label $\{\mathbf{x}_i\}$ by horizontal direction	$\triangleright\ O(n_s n_x)$
6:	$\{X_i\} \leftarrow$ cluster feature points by homography	$\triangleright\ O(n_x^3)$
7:	$\{\phi_i\} \leftarrow$ segment planar facades	$\triangleright\ O(n_x \log n_x)$

VI. 3D Structure Estimation

After a set of corresponding PBFs $\{\phi_i\} \leftrightarrow \{\phi_i'\}$ is extracted from two views, we simultaneously estimate the camera pose and 3D PBF positions. The camera pose, $\mathtt{R}'$ and $\mathbf{t}'$, is initialized using the standard steps in [21]; the fundamental matrix $\mathtt{F}$ is fit to keypoint matches using RANSAC; the camera pose $\mathtt{R}'$ and $\mathbf{t}'$ are obtained by decomposing the essential matrix $\mathtt{E} = \mathtt{K}'^\mathsf{T}\mathtt{F}\mathtt{K}$. The 3D PBF positions $\{\Phi_i\}$ are initialized by triangulating the set of feature points and the set of line segment endpoints. The plane is initialized by fitting Π to the triangulated feature points and line segment endpoints.

To refine the initial estimates using geometric constraints, our goal is to solve

$$\underset{\mathtt{R}',\mathbf{t}',\{\Phi_i\}}{\arg\min}\, J_{\text{rep}} + J_{\text{ori}} + J_{\text{cop}}, \tag{4}$$

where J_{rep}, J_{ori}, and J_{cop} are the cost terms for the reprojection error, orientation constraint, and coplanarity constraint, respectively. Each cost term J_{rep}, J_{ori}, and J_{cop} is defined in the following subsections.

A. Reprojection Error: The reprojection error J_{rep} in (4) is minimized when the projections of the estimated 3D points and line segments are close to their image measurements. We compute the reprojection error by

$$J_{\text{rep}} = \frac{J_{\text{rep}}^x}{\lambda_{\text{rep}}^x} + \frac{J_{\text{rep}}^l}{\lambda_{\text{rep}}^l} + \frac{J_{\text{rep}}^e}{\lambda_{\text{rep}}^e}, \tag{5}$$

where J_{rep}^x, J_{rep}^l, and J_{rep}^e are the reprojection errors for feature points, lines, and line-segment endpoints, respectively. The λ values for each cost term will be described later.

To compute J_{rep}^x, we use the standard reprojection error in [21]. Let X be the set of feature points. The reprojection error over X is defined by $J_{\text{rep}}^x = \sum_{\mathbf{x}\in X} d(\mathbf{x},\hat{\mathbf{x}})^2 + d(\mathbf{x}',\hat{\mathbf{x}}')^2$ where $d(\mathbf{x},\hat{\mathbf{x}})$ is the distance between the observation $\mathbf{x}$ and estimation $\hat{\mathbf{x}}$ in image coordinates.

We use the line reprojection error in [22] to compute J_{rep}^l. The cost function for the set of lines S is

$$\begin{aligned} J_{\text{rep}}^l = \sum_{\mathbf{s}\in S} & d(\hat{\mathbf{e}}_1,\mathbf{p}_1)d(\mathbf{p}_1,\mathbf{p}_3) + d(\hat{\mathbf{e}}_2,\mathbf{p}_2)d(\mathbf{p}_2,\mathbf{p}_3) \\ & + d(\hat{\mathbf{e}}_1',\mathbf{p}_1')d(\mathbf{p}_1',\mathbf{p}_3') + d(\hat{\mathbf{e}}_2',\mathbf{p}_2')d(\mathbf{p}_2',\mathbf{p}_3'), \end{aligned} \tag{6}$$

where $\mathbf{p}_1$ and $\mathbf{p}_2$ are the projections of the estimated endpoints $\hat{\mathbf{e}}_1$ and $\hat{\mathbf{e}}_2$ on the line $\mathbf{e}_1 \times \mathbf{e}_2$. The point $\mathbf{p}_3$ is the intersection of the estimated line $\hat{\mathbf{e}}_1 \times \hat{\mathbf{e}}_2$ and the observed line $\mathbf{e}_1 \times \mathbf{e}_2$. If the reprojection error for lines is used alone, the estimated line segments would "grow" or "shrink" during the estimation process. To avoid this, we also apply the reprojection error to line-segment endpoints by

$$J_{\text{rep}}^e = \sum_{\mathbf{s}\in S} d(\mathbf{e}_1,\hat{\mathbf{e}}_1)^2 + d(\mathbf{e}_1',\hat{\mathbf{e}}_1')^2 + d(\mathbf{e}_2,\hat{\mathbf{e}}_2)^2 + d(\mathbf{e}_2',\hat{\mathbf{e}}_2')^2. \tag{7}$$

B. Orientation Constraint The 3D orientation of each line segment and plane can be computed since their associated vanishing direction is known. To constrain the overall orientation of the building structure, we minimize the following cost function

$$J_{\text{ori}} = \frac{J^v_{\text{ori}}}{\lambda^v_{\text{ori}}} + \frac{J^n_{\text{ori}}}{\lambda^n_{\text{ori}}} + \frac{J^p_{\text{ori}}}{\lambda^p_{\text{ori}}}, \tag{8}$$

where J^v_{ori} and J^n_{ori} constrain the line segment orientation and J^p_{ori} constrains the plane orientation.

The first term in (8) makes each line segment $\mathbf{S}$ parallel to its associated vanishing direction by $J^v_{\text{ori}} = \sum_i \sum_{\mathbf{S}\in S} \|\hat{\mathbf{d}}_s \times \mathbf{d}_i\|^2$, where $\hat{\mathbf{d}}_s$ is the direction of the line segment $\mathbf{S}$ and $\mathbf{d}_i$ is the vanishing direction of the vanishing point $\mathbf{v}_i$. As in the line reprojection error, the above cost function represents the area between the line segment direction vector and the vanishing direction vector. Each line segment direction $\hat{\mathbf{d}}_s$ is also enforced to be perpendicular to its associated plane normal $\hat{\mathbf{n}}$ by $J^n_{\text{ori}} = \sum_i \sum_{\mathbf{S}\in S} (\hat{\mathbf{d}}_s \cdot \hat{\mathbf{n}}_i)^2$. The above cost function prevents line segments becoming perpendicular to the plane due to the coplanarity constraint in (10), which is described in the next section. Each plane normal $\hat{\mathbf{n}}_i$ is constrained so that it is perpendicular to the associated horizontal vanishing direction $\mathbf{d}_h$ and vertical vanishing direction $\mathbf{d}_z$ using $J^p_{\text{ori}} = \sum_i (\mathbf{d}_h \cdot \hat{\mathbf{n}}_i)^2 + (\mathbf{d}_z \cdot \hat{\mathbf{n}}_i)^2$.

C. Coplanarity Constraint: A strong constraint to reduce the depth ambiguity of 3D points and line segments is the coplanarity constraint. The coplanarity constraint for feature points and line segments is defined by

$$J_{\text{cop}} = \frac{J^x_{\text{cop}}}{\lambda^x_{\text{cop}}} + \frac{J^s_{\text{cop}}}{\lambda^s_{\text{cop}}}, \tag{9}$$

where J^x_{cop}, and J^s_{cop} are the cost terms for feature points and line-segment endpoints, respectively.

We compute J^x_{cop} by $J^x_{\text{cop}} = \sum_i \sum_{\mathbf{X}\in X_i} d_\perp(\hat{\mathbf{X}}, \hat{\Pi}_i)^2$ where $d_\perp(\hat{\mathbf{X}}, \hat{\Pi}_i)$ is the perpendicular distance from point $\hat{\mathbf{X}}$ to plane $\hat{\Pi}_i$. For line segments, we enforce coplanarity by minimizing the area between the line segment and the plane by

$$J^s_{\text{cop}} = \sum_i \sum_{\mathbf{S}\in S} (d_\perp(\hat{\mathbf{E}}_1, \hat{\Pi}_i) + d_\perp(\hat{\mathbf{E}}_2, \hat{\Pi}_i)) d(\mathbf{P}_1, \mathbf{P}_2), \tag{10}$$

where $\mathbf{P}_1$ and $\mathbf{P}_2$ are projected endpoints of the line segment $\mathbf{S}$ on plane Π_i. The area between a line segment and plane Π_i becomes zero when they are coplanar.

We use the Levenberg-Marquardt algorithm to solve our non-linear optimization problem in (4). To balance the three cost terms in (4), as in [23], we set each weight λ as the initial value of its corresponding J so that the initial cost of $J_{\text{rep}} + J_{\text{ori}} + J_{\text{cop}} = 8$.

VII. Experiments

To measure the performance of our proposed method, we have tested the method on real data. We have taken a pair of images of 20 different buildings on Texas A&M University campus. Two tests have been conducted:

A. PBF Segmentation Test: We have measured the performance of each step described in Sec. V. For the building region segmentation step (Sec. V-A), we have counted the number of true-positive (TP), true-negative (TN), false-positive (FP), and false-negative (FN) feature points after the building region segmentation has been performed. The ground-truth was obtained by manually examining whether each feature point was on a building facade. Fig. 5 shows the sensitivity ($\text{TP}/(\text{TP}+\text{FN})$) and specificity ($\text{TN}/(\text{FP}+\text{TN})$) of our method. Next, for the planar facade detection step (Sec. V-B), we have computed the precision ($\text{TP}/(\text{TP}+\text{FP})$). We have compared the precision of our proposed method with a base-line method, i.e. the J-linkage in [8] which uses J-linkage directly on the image without use of the horizontal vanishing direction information. It is worth noting that our method also uses J-linkage as a sub-step. The parameters for the J-linkage step are the same in both methods. Fig. 6 shows that our proposed method is always better. The average precision increase over the J-linkage method is 45.27%.

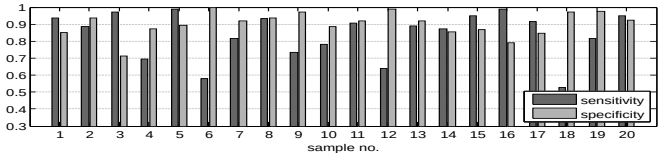


Fig. 5: Sensitivity and specificity of PBF region detection

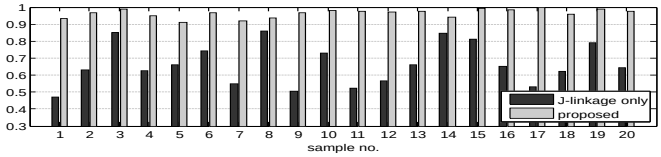
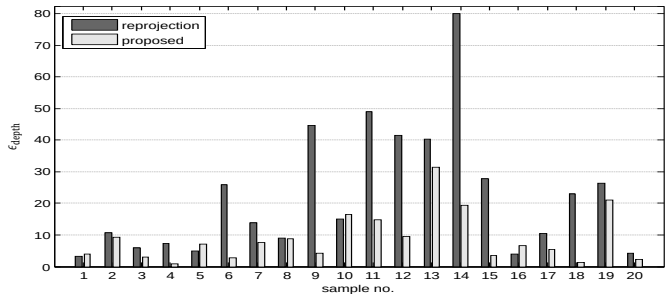


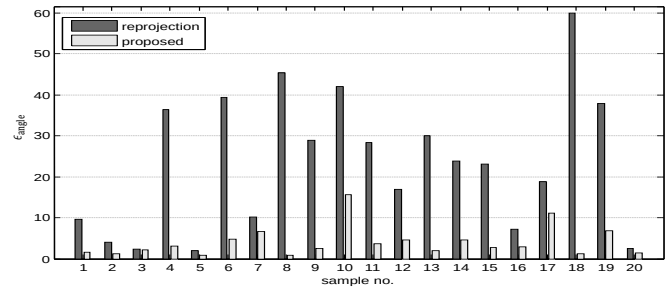
Fig. 6: Precision comparison of PBF detection

B. PBF Mapping Test: We have compared our proposed method with a base-line method which minimizes the standard reprojection error in structure estimation. Since the camera baseline distance is relatively small compared to the building distance, the camera positional error is not significant in either method. Hence, we present two measures, (1) the PBF depth error and (2) the PBF angular error, to illustrate the performance of our proposed method.

First, we measure the depth error of the mapped building. Each image has been taken so that the principal axis passes through the intersection of the two primary PBFs, i.e. the building corner. We set the baseline distance to be 1/60 of the distance to the building corner. Suppose $\mathbf{l}_k$ is the intersection of the estimated 3D planes $\hat{\Pi}_i$ and $\hat{\Pi}_j$ of PBFs, and let Π_z be the plane passing through the camera center $\mathbf{C}$ with the normal vector $\mathbf{d}_x \times \mathbf{d}_z$, where $\mathbf{d}_x = [1,0,0]^\mathsf{T}$ and $\mathbf{d}_z = \mathtt{K}^{-1}\mathbf{v}_z$. We measure the depth error of a building corner $\mathbf{l}_k$ by $\varepsilon_{\text{depth}} = \left|\bar{d} - \frac{1}{2}(d_\perp(\hat{\mathbf{Q}}_1, \Pi_z) + d_\perp(\hat{\mathbf{Q}}_2, \Pi_z))\right|$ where the ground-truth depth $\bar{d}$ is measured from a top-down view aerial map and a precision laser ranger (BOSCH GLR225) with 1 mm accuracy. If P is the set of feature points $X_i \cup X_j \cup E_i \cup E_j$ of Φ_i and Φ_j, then the points $\hat{\mathbf{Q}}_1$ and $\hat{\mathbf{Q}}_2$ are the intersections of $\mathbf{l}_k$ with the plane parallel to Π_z which passes the points $\max_{\mathbf{P}\in P}(\mathbf{P}^\mathsf{T}\mathbf{d}_z)\cdot\mathbf{d}_z$ and $\min_{\mathbf{P}\in P}(\mathbf{P}^\mathsf{T}\mathbf{d}_z)\cdot\mathbf{d}_z$, respectively. Fig. 7a shows that the overall depth error decreases after the refinement. In fact, the average depth error is reduced from 22.39 to 8.95, a 60% reduction.



(a) PBF depth error



(b) PBF angular error

Fig. 7: Comparison of the reprojection error-based 3D estimation and that of the proposed method.

Next, to measure how well the building structure is recovered, we measure the angular error of PBFs by $\varepsilon_{\text{angle}} = \left|\bar{\theta}_a - \hat{\theta}_a\right|$, where $\bar{\theta}_a$ is the ground-truth angle between the two primary PBFs measured from a top-down view aerial map and the same high precision laser ranger, and $\hat{\theta}_a$ is the estimated angle which is computed by $\hat{\theta}_a = \cos^{-1}\left(\hat{\mathbf{n}}_i^{\mathsf{T}}\hat{\mathbf{n}}_j/(\|\hat{\mathbf{n}}_i\|\|\hat{\mathbf{n}}_j\|)\right)$. where $\mathbf{n}_i$ and $\mathbf{n}_j$ are plane normal vectors for $\hat{\Pi}_i$ and $\hat{\Pi}_j$, respectively. Fig. 7b shows that the pairwise angular errors are reduced using our proposed method. The average angular error reduction is 82.82%.

VIII. Conclusion and Future Work

We reported a novel algorithm for PBF segmentation and 3D estimation. We proposed a new RI to distinguish building regions based on homogeneous region detection results. We combined the reprojection errors, orientation constraints, and coplanarity constraints as cost functions in an optimization process to improve the 3D estimation of the building structure. In physical experiments, we compared our algorithm with state-of-the-art J-linkage-based facade detection. The results showed that our method increases precision in segmentation and reduces depth and angular errors in 3D estimation. In the future, we will extend this work to a high-level landmark-based SLAM approach where PBF will be used as landmarks. We will also consider other sensors such as inertial sensors and/or depth sensors to address scale drift issues.

ACKNOWLEDGMENT

We would like to thank W. Li, M. Hielsberg, Z. Gui, X. Wang, S. Jacob, and P. Peelen for their input and contributions to the NetBot Lab at TAMU.

References

[1] D. F. Fouhey, D. Scharstein, and A. J. Briggs, "Multiple plane detection in image pairs using J-linkage," in *Proceedings of the International Conference on Pattern Recognition*, 2010, pp. 336–339.

[2] R. Kaushik and J. Xiao, "Accelerated patch-based planar clustering of noisy range images in indoor environments for robot mapping," *Robotics and Autonomous Systems*, vol. 60, no. 4, pp. 584–598, Apr. 2012.

[3] C. Taylor and A. Cowley, "Parsing indoor scenes using RGB-D imagery," in *Proceedings of Robotics: Science and Systems*, July 2012.

[4] T. Lee, S. Lim, S. Lee, S. An, and S. Oh, "Indoor mapping using planes extracted from noisy RGB-D sensors," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 1727–1733.

[5] Y. Taguchi, Y.-D. Jian, S. Ramalingam, and C. Feng, "Point-plane slam for hand-held 3D sensors," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2013.

[6] J. A. Delmerico, P. David, and J. J. Corso, "Building facade detection, segmentation, and parameter estimation for mobile robot localization and guidance," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 1632–1639.

[7] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, June 1981.

[8] R. Toldo and A. Fusiello, "Robust multiple structures estimation with J-linkage," in *Proceedings of the European Conference of Computer Vision*, 2008, pp. 537–547.

[9] C. Baillard and A. Zisserman, "A plane-sweep strategy for the 3D reconstruction of buildings from multiple images," in *ISPRS Journal of Photogrammetry and Remote Sensing*, 2000, pp. 56–62.

[10] J. Zhou and B. Li, "Homography-based ground detection for a mobile robot platform using a single camera," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2006, pp. 4100–4105.

[11] Y. Li and S. Birchfield, "Image-based segmentation of indoor corridor floors for a mobile robot," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 837–843.

[12] O. Haines, J. Martnez-Carranza, and A. Calway, "Visual mapping using learned structural priors," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2013, pp. 2227–2232.

[13] Y. Lu, D. Song, Y. Xu, A. Perera, and S. Oh, "Automatic building exterior mapping using multilayer feature graphs," in *IEEE International Conference on Automation Science and Engineering*, 2013, pp. 162–167.

[14] H. Li, D. Song, Y. Lu, and J. Liu, "A two-view based multilayer feature graph for robot navigation," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2012, pp. 3580–3587.

[15] Y. Lu, D. Song, and J. Yi, "High level landmark-based visual navigation using unsupervised geometric constraints in local bundle adjustment," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2014.

[16] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[17] R. G. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: a line segment detector," *Image Processing On Line*, 2012.

[18] B. Fan, F. Wu, and Z. Hu, "Robust line matching through line-point invariants," *Pattern Recognition*, vol. 45, no. 2, pp. 794–805, Feb. 2012.

[19] E. Tretyak, O. Barinova, P. Kohli, and V. Lempitsky, "Geometric image parsing in man-made environments," *International Journal of Computer Vision*, vol. 97, no. 3, pp. 305–321, May 2012.

[20] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, Sept. 2004.

[21] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[22] C. J. Taylor and D. J. Kriegman, "Structure and motion from line segments in multiple images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 11, pp. 1021–1032, Nov. 1995.

[23] K. Kanatani, "Calibration of ultrawide fisheye lens cameras by eigenvalue minimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 4, pp. 813–822, 2013.