A Two-View based Multilayer Feature Graph for Robot Navigation

Haifeng Li, Dezhen Song, Yan Lu, and Jingtai Liu

Abstract— **To facilitate scene understanding and robot navigation in a modern urban area, we design a multilayer feature graph (MFG) based on two views from an on-board camera. The nodes of an MFG are features such as scale invariant feature transformation (SIFT) feature points, line segments, lines, and planes while edges of the MFG represent different geometric relationships such as adjacency, parallelism, collinearity, and coplanarity. MFG also connects the features in two views and the corresponding 3D coordinate system. Building on SIFT feature points and line segments, MFG is constructed using feature fusion which incrementally, iteratively, and extensively verifies the aforementioned geometric relationships using random sample consensus (RANSAC) framework. Physical experiments show that MFG can be successfully constructed in urban area and the construction method is demonstrated to be very robust in identifying feature correspondence.**

I. Introduction

When a mobile robot travels in a modern urban environment, the robot often needs visual signals from its on-board camera to assist navigation. A typical modern urban environment is usually rectilinear and consists of many structured objects and distinctive features such as vertical walls, parallel edges, orthogonal planes, etc. Extracting such features from video frames to form a quick scene understanding can directly benefit navigation tasks such as localization, mapping, obstacle avoidance, and motion planning.

Here we design a multilayer feature graph (MFG) to facilitate the scene understanding in urban area. Nodes of an MFG are features such as scale invariant feature transformation (SIFT) feature points, line segments, lines, and planes while edges of the MFG represent different geometric relationships such as adjacency, parallelism, collinearity, and coplanarity. MFG also connects the features in two views and the corresponding 3D world coordinate system. Fig. 1 illustrates the MFG in the 3D world coordinate system. We design an MFG construction method using a feature fusion process which incrementally, iteratively, and extensively verifies the aforementioned geometric relationships using random sample consensus (RANSAC) framework.

We have implemented MFG construction algorithm and tested it in physical experiments. Results show that MFG can be successfully constructed from raw image data. Since the process utilizes multiple types of geometric relationships, the feature correspondence outcome is significantly improved over existing feature matching methods. As an important part of MFG, the algorithm is able to detect all primary vertical planes with reasonable accuracy.

This work was supported in part by National Science Foundation under CAREER grant IIS-0643298 and Chinese Scholarship Council.

H. Li and J. Liu are with the Institute of Robotics and Automatic Information System, Nankai University, Tianjin 300071, P. R. China. Emails: {*lihf, liujt*}*@robot.nankai.edu.cn*.

D. Song and Y. Lu are with the Department of Computer Science and Engineering, Texas A&M University, College Station, TX 77843, United States. Emails: {*ylu, dzsong*}*@cse.tamu.edu*.

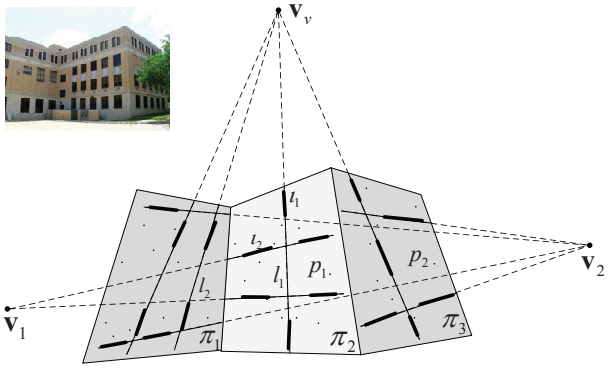


Fig. 1. An illustration of the multilayer feature graph. Based on vanishing points, building facades, parallel line/line segment groups, and feature points, an MFG is a feature-based scene reconstruction which focuses on robot navigation needs. The camera frame at the top left corner is one of the input two views and the output is the MFG.

II. Related Work

Our MFG is a novel scene understanding and knowledge representation method for vision-based robot navigation in urban area. A mobile robot usually has different ways of constructing and representing its surrounding environments, which usually depend on different sensors, 3D reconstruction schemes, and feature matching methods.

The most common sensors for robot navigation include sonar arrays [1], laser range finders [2], [3], depth cameras [4], regular cameras [5]–[9], or their combinations [10], [11]. Simultaneous localization and mapping (SLAM) is the typical framework employed by robot navigation [12] to make decisions based on the sensory input. In a SLAM framework, the physical world is represented as a collection of "landmarks." For example, landmarks are point clouds if a laser ranger finder or a depth camera is the primary sensor. In vision-based SLAM, SIFT feature points or its variants [5], [6], [13] and line features [7]–[9] are often employed as landmarks. Landmarks can be viewed as a rudimentary way of scene understanding which serves SLAM purposes well. Our MFG complements SLAM approaches because a better scene understanding method can also increase the robustness of localization results and make it easy to utilize prior knowledge such as existing maps.

In computer vision and graphics field, 3D reconstruction

has been a very popular topic for research as well as commercial applications. Sensors used there also include laser range finder [14] and more often, aerial cameras [15]. Google Earth and Microsoft Virtual Earth are successful showcases for 3D reconstruction of city models [16]. Following the taxonomy of Seitz et al. [17], 3D reconstruction algorithms can be categorized into four classes: voxel approaches [18], level-set techniques [19], [20], line segment matching [21], polygon mesh methods [22], and algorithms that compute and merge depth maps [23], [24]. Unlike those methods, our MFG does not pursuit a full scale reconstruction and hence does not require intensive computation or suffers from occlusion, correspondence, and lighting problems in the field.

Our MFG focuses on key features that represent building facades and orthogonal/parallel lines which are insensitive to illumination and shadow problems caused by natural lighting. In a closely related work, Cham et al. [25] identify vertical corner edges of buildings as well as the neighboring plane normals from a single ground-view omnidirectional image to estimate the camera pose. Recent work by Delmerico et al. [26] proposes a method to determine a set of candidate planes by sampling and clustering points from stereo images with RANSAC using estimated local normals. These methods provide the inspiration that planes are important and robust features to be extracted in reconstruction. However, our MFG does not rely on the depth information from existing stereo images or a specialized imaging device. MFG simply employs multiple types of features and utilizes multiple internal geometric relationships between features for better robustness and applicability.

Properly matching multiple types of features across views is essential to MFG construction. Although point feature matching [13], [27] is relatively mature, line segment matching is very challenging due to inaccurate end points, occlusion, and complex correspondence. Schmid and Zisserman [28] utilize the epipolar constraints of line segment end points for matching, which requires the prior knowledge of epipolar geometry. The color-based methods [29]–[31] perform poorly when color features are not distinctive and hence are sensitive to illumination conditions. Other grouping matching methods [32], [33] are limited by either high computational complexity or sensitivity to inaccurate endpoints of line segments. Recently, Fan et al. [34] propose a line segment matching method by leveraging feature point correspondences in adjacent regions. This is the best available method for line segment matching and is named as "point-based line matching (PBLM)" in this paper. This method is promising but still misses many line segment matches due to lack of feature points. Inspired by this method, our MFG addresses these issues by introducing ideal lines and analyzing collinear and coplanarity relationships.

Our group has worked on robot navigation using passive vision system in past decade. We have developed appearance-based method [35], investigated how depth error affects navigation [36], and used vertical line segments for visual odometry tasks [37], [38]. In the process, we have realized it is necessary to combine benefits of different features to assist navigation which leads to this work.

III. Problem Definition

A. Assumptions

To formulate the problem and focus on the most relevant issues, we have the following assumptions.

- The robot is in a modern urban environment where rectilinear polygonal buildings dominate the scene.
- To assist scene understanding, the robot is equipped with a gravity sensor and knows its vertical direction.
- The intrinsic parameters of the finite perspective camera are known by pre-calibration. The lens distortion of the camera has been removed.
- The robot knows baseline distance for two views. This can be achieved with on-board inertial sensors or wheel encoders. These sensors are good at short distance measurement. It is worth noting that the measurement may contain error. We use the measurement as an initial input and the baseline distance will be refined during the image feature maturing process.

B. Notations, Coordinate Systems, and Problem Definition

In this paper, all the coordinate systems are right hand systems. The superscript $'$ denotes the corresponding notation in the second view. Notations in the format of (a,a') refer to a corresponding pair of variables or parameters in the first and second views, respectively. Let us define

- $\{W\}$ as a 3D Euclidean world coordinate system (WCS) with its $x-z$ plane being horizontal,
- $\{C\}$ and $\{C'\}$ as two 3D camera coordinate systems (CCS) for the first and the second views, respectively, (For each CCS, its origin is at the camera optical center, its $z-$axis coincides with the optical axis and points to the forward direction of the camera, its $x-$axis and $y-$axis are parallel to the horizontal and vertical directions of the CCD sensor plane, respectively.)
- $\{I\}$ and $\{I'\}$ as two 2D image coordinate systems (ICS) for the first and the second views, respectively, (For each ICS, its $u-$axis and $v-$axis are parallel to x and y axes of the corresponding CCS, respectively)
- F_r and F_r' as the raw images taken at the first and the second views, respectively, and
- $\mathbf{K}$ as the intrinsic parameter matrix of the camera.

With these notations defined, our problem is,

Definition 1: Given F_r and F_r', construct the MFG.

Now let us introduce MFG.

IV. Multilayer Feature Graph

Fig. 2 illustrates how MFG organizes different types of features according to their geometric relationships. The bottom layers of MFG are raw features such as SIFT points and line segments while the top layers of MFG contain planes describing the structure of the scene. To explain the structure of MFG, we begin with the raw feature extraction.

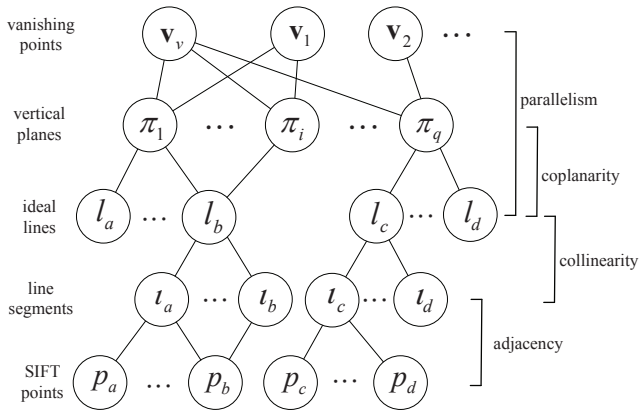


Fig. 2. The structure of the multilayer feature graph.

A. Layers 1&2: Raw Features

SIFT points and line segments are the raw features of MFG (see Fig. 2). Define $P := \{p_1, p_2, \ldots\}$ as the SIFT point feature set with p_i being the i-th SIFT point feature. Line segments are extracted using the line segment detector (LSD) method [39]. Define $\iota_i = [\mathbf{E}_{i,0}, \mathbf{E}_{i,1}]^T$ as the i-th line segment with its two end points: $\mathbf{E}_{i,0} = [u_{i,0}, v_{i,0}]^T$ and $\mathbf{E}_{i,1} = [u_{i,1}, v_{i,1}]^T$. Let $L = \{\iota_1, \iota_2, \ldots, \iota_n\}$ denote a line segment set with each element being a line segment in ICS.

Each SIFT point or line segment represents a node in MFG. If a SIFT point is in the neighborhood (will be defined later in the paper) of a line segment, then an edge between them is established with the line segment being the parent node. It is possible that a line segment node does not have any child SIFT nodes (ι_d in Fig. 2). A SIFT node may have multiple parent line segment nodes (p_b in Fig. 2).

Note that SIFT points are only used to assist line segment matching in the paper. That is why we exam their adjacency to line segments.

B. Layer 3: Ideal Lines

To aggregate line segments and extract potential lines from the raw features, we introduce ideal lines.

Definition 2: An ideal line is defined as a real or virtual line passing through a set of collinear line segments. An ideal line might/might not correspond to a real line in 3D space.

Define $\mathbf{L} = \{l_1, l_2, \ldots, l_m\}$ as an ideal line set with $l_i, i = 1, \ldots m$, being an ideal line in ICS. For a given set of collinear line segments $\{\iota_a, \iota_b, \ldots\}$, its ideal line l_i is obtained by fitting the line through the end points of all line segments in the set using the maximum likelihood estimation (MLE) method [40]. As shown in Fig. 2, each line segment must have only one corresponding ideal line as its parent node. An ideal line may have multiple child line segment nodes. An ideal line usually has three representations: 3D format in $\{W\}$ and two 2D formats in $\{I\}$ and $\{I'\}$ of the two views, respectively.

C. Layer 4: Vertical Planes

Vertical planes in $\{W\}$ form layer 4. Let $\pi_i, i = 1, 2, \ldots, q$, denote vertical planes in $\{W\}$ and $\pi_i = [\mathbf{n}_i, d_i]^T$, where $\mathbf{n}_i$ is the normal vector of π_i, and d_i is the distance from the camera center to π_i. Note that we do not include horizontal planes in the design because horizontal planes represent either road planes or building top planes. The former does not exist when road is not flat and the latter is not visible from a ground robot.

If an ideal line is located in a vertical plane, an edge between the two nodes is established in MFG. A vertical plane must have at least two child nodes because two lines determine a plane when they are either intersecting or being parallel to each other. An ideal line may not necessarily have a parent node if it corresponds to an isolated linear object such as a light pole. An ideal line may have two parent vertical planes if it is a boundary line.

D. Layer 5: Vanishing Points

In a vertical plane, there usually are many parallel ideal lines. Those parallel ideal lines intersect each other at vanishing points. Each vertical plane usually has two groups of parallel ideal lines: one horizontal group and one vertical group from building and window boundaries. Therefore, each vertical plane has two dominating vanishing points. Fig. 2 illustrates the relationship by connecting each vertical plane node to two parent vanishing point nodes.

Since modern urban area is largely a rectilinear environment, there usually are three dominating vanishing points with one being vertical (denoted as $\mathbf{v}_v$) and two being horizontal (denoted as $\mathbf{v_1}$ and $\mathbf{v_2}$). For simplicity, we use three vanishing points in the rest of the paper although the approach can be adaptive to varying numbers. Fig. 2 also shows that a vanishing point may correspond to parallel ideal line groups in different planes. For example, the vertical vanishing point node links to every vertical plane node with vertical ideal lines.

V. MFG Construction via Feature Fusion

Constructing MFG is nontrivial. It is a scene understanding process. Raw features in layers 1&2 and vanishing points in layer 5 are straightforward to obtain using existing methods. However, ideal lines and vertical planes, which represent structure of the scene at different granularity, are difficult to compute because proper correspondence between two views needs to be established. Fig. 3 illustrates the process of MFG construction, which is named as feature fusion. The process can be divided into three main components: parallelism, collinearity, and coplanarity verifications.

A. Parallelism Verification

The purpose of parallelism verification is to divide line segments and ideal lines into different parallel groups and find group correspondence across the two views. Since parallel lines intersect each other at vanishing points, vanishing points are natural classifiers for the parallelism verification.

Steps 1-5 in Fig. 3 illustrate the procedure. In step 1, raw line segments are extracted using LSD. We then apply RANSAC framework to line segments to estimate vanishing points. Since we know the vertical direction from gravity

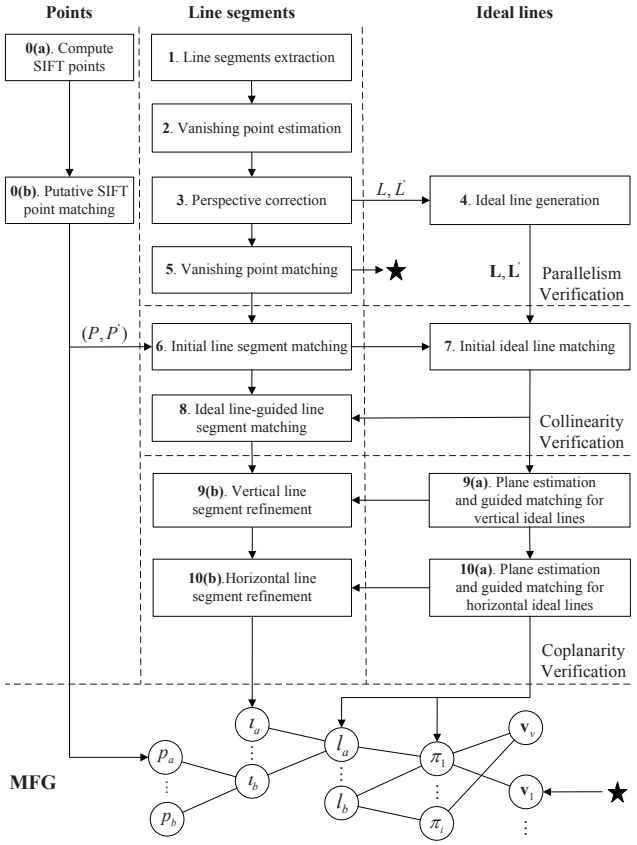


Fig. 3. A block diagram for multilayer feature fusion.

sensors, it is easy to identify vanishing point $\mathbf{v}_v$ which corresponds to vertical line segments. It is worth noting that $\mathbf{v}_v$ must exist because urban buildings have many vertical boundaries.

Horizontal vanishing points can be found by the following orthogonality check,

$$\|\mathbf{v}_v{}^T \omega \hat{\mathbf{v}}_i\| < \varepsilon, \tag{1}$$

where $\hat{\mathbf{v}}_i$ is a candidate horizontal vanishing point, $\omega = (\mathbf{K}\mathbf{K}^T)^{-1}$ is the image of the absolute conic, and ε is a small positive threshold.

After identifying vanishing points, we can perform the perspective correction [40] for both views to make v-axes of ICSs to be vertical and therefore all vertical lines appear vertical in the two views (step 3). We define F and F' as the first and second views after the perspective correction, respectively. Meanwhile, initial set of ideal lines in $\{I\}$ and $\{I'\}$ can be established using MLE and RANSAC as well (step 4). Ideal lines can be associated with line segments in each view based on the inlier set of the RANSAC output.

With the three vanishing points, the ideal lines and line segments in each view can be classified into three groups. This allows us to match different parallel groups across two views by matching vanishing points [41] (step 5). Fig. 4 illustrates the result of the vanishing point matching.

The correspondence between two views for parallel groups by vanishing point matching is still too gross to be used to establish scene understanding. In fact, we need to establish cross view correspondence between line segments and ideal lines. However, one-to-one correspondence for line segments is impossible to achieve due to severe occlusion and noises. Collinearity verification is proposed to handle the challenge.

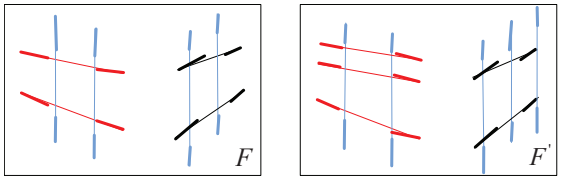


Fig. 4. An example of vanishing point matching across F and F'. The line segments and ideal lines associated with the same vanishing points are drawn in the same color (black, blue and red). In this example, vertical lines appear vertical because F and F' are results of the perspective correction.

B. Collinearity Verification

Collinearity verification is to find the correspondences of ideal lines and hence line segments across two views. This process also establishes ideal lines and line segments in $\{W\}$.

A corresponding pair of line segments or ideal lines must be in the same parallel groups defined by the same vanishing points, which can help us reduce the matching problem size. Since a corresponding pair of line segments in the two views do not necessarily have corresponding end points due to occlusion and camera perspective issues, checking end point correspondence is not viable. Also an one-to-one correspondence does not necessarily exist. In fact, many-to-one and one-to-many mappings are not unusual. Collinearity verification is designed to handle these challenging issues.

Steps 6-8 in Fig. 3 illustrate the process. We first find the initial candidate correspondences for line segments using the PBLM method [34] (step 6). For completeness, we brief the PBLM method using Fig. 5.

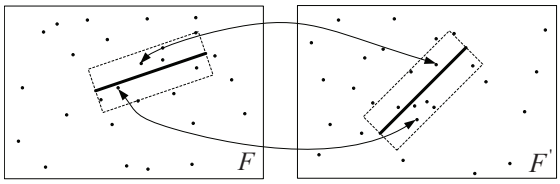


Fig. 5. An example of initial line segment matching using PBLM in [34].

For line segment ι_j, we define its neighbor region as $Ne(\iota_j)$. The doted rectangles in Fig. 5 are the neighbor regions. ι_j bisects $Ne(\iota_j)$ which has a half side length of d_t along the direction perpendicular to ι_j. On the other hand, we apply SIFT to F and F'. If a SIFT point p_i lies in the neighbor region, then we denote it as $p_i \in Ne(\iota_j)$. The initial line segment matching uses putative SIFT point correspondences: for each putative point pair (p_i, p_i'), if $p_i \in Ne(\iota_a)$ and $p_i' \in Ne(\iota_b')$, then the similarity between ι_a and ι_b' increases by 1. The initial line segment matches are determined by thresholding the overall similarity.

The pro of this method is that it does not depend on end point matching while the con is that it may miss many

potential matches due to lack of features. We need to refine the results using ideal lines. Steps 7 and 8 in Fig. 3 show the process. Using initial line segment correspondences, we can tentatively match the ideal lines: given two ideal lines l_m and l'_n in two views, we can determine l_m and l'_n as a corresponding pair if the following is true,

$$\exists \iota_i \in l_m, \iota'_j \in l'_n \quad \text{s.t.} \quad (\iota_i, \iota'_j) \in S_{\iota,\iota'}, \tag{2}$$

where $S_{\iota,\iota'}$ is the set of all initial line segment matches.

In step 8, we use the matched ideal line pairs to search for more line segment matches missed by the initial matching in step 7. Fig. 6 shows an example, where (l_1, l'_1) is a pair of corresponding ideal lines obtained from the matched line segment pair (ι_1, ι'_1). (ι_2, ι'_2) was not matched in the initial matching step due to lack of SIFT features in the neighbor region. There is no correspondence for ι_3 in the second view due to occlusions. Applying MSLD [31] metric, the line segment correspondence (ι_2, ι'_2) is found. Furthermore, (l_1, l'_1) helps us find the many-to-many correspondence $[(\iota_1, \iota_2, \iota_3), (\iota'_1, \iota'_2)]$ across two views.

Note that not all line segment or ideal line correspondences are correct at this step. However, most wrong matches can be removed by checking coplanar relationship next.

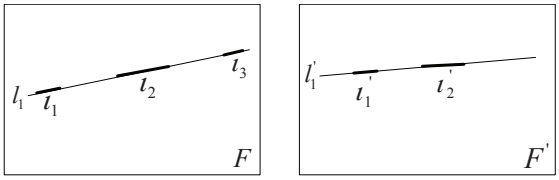


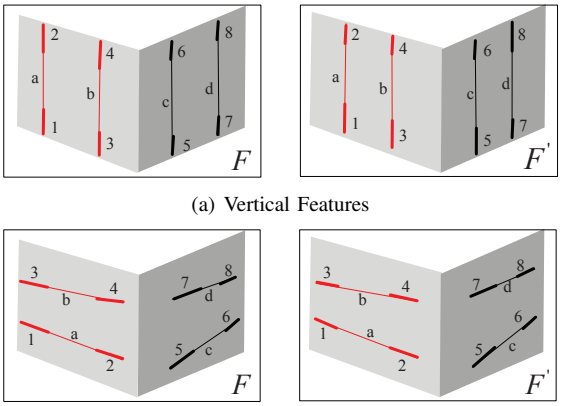
Fig. 6. An example of ideal line-guided line segment matching.

C. Coplanarity Verification

So far, we have completed the construction of layers 1, 2, 3, and 5 of MFG. The remaining layer 4 consists of vertical planes, which represent building facades and are of great importance for robot navigation. Layer 4 only exists in $\{W\}$. Coplanarity verification is designed to classify the coplanar features and reconstruct these planes. The coplanar verification process associates ideal lines to their residing planes (i.e. forming edges between ideal lines and vertical planes in MFG in Fig. 2). The process also removes false correspondences and finds more correspondences missed by previous steps.

1) Coplanarity Verification for Vertical Features: This process is done in two stages. Step 9(a) in Fig. 3 refers to the first stage where we estimate vertical planes from vertical ideal lines and use the plane to verify vertical line matches.

In a top-down view, a vertical plane is reduced to a single line $\pi_i = [\mathbf{n}^T_{2\times1}, d]^T$ on the $x-z$ plane of $\{W\}$. All the vertical ideal lines are reduced to points on the plane. Also, the 2D camera degenerates into a 1D camera. For the two 1D views, we define the camera matrices as $\mathbf{P}_{2\times3} = \mathbf{K}_{2\times2}[\mathbf{I}_{2\times2}|\mathbf{0}_{2\times1}]$ and $\mathbf{P}'_{2\times3} = \mathbf{K}_{2\times2}[\mathbf{R}_{2\times2}|\mathbf{t}_{2\times1}]$, respectively, where the intrinsic parameters of the 1D camera, $\mathbf{K}_{2\times2}$, can be easily determined from the 2D camera parameter matrix



(a) Vertical Features

(b) Horizontal Features

Fig. 7. Coplanarity verification. These ideal line and line segments are classified into two groups representing in different colors (black and red) according to their residing vertical planes.

$\mathbf{K}$, $\mathbf{R}_{2\times2}$ is the 2D rotation matrix, and $\mathbf{t}_{2\times1} = [t_x, t_z]^T$ denotes the translation between the two camera centers.

According to [40], line $\pi_i = [\mathbf{n}^T_{2\times1}, d]^T$ introduces a 1D homography for corresponding points in the two views,

$$\mathbf{H}_{2\times2} = \mathbf{K}_{2\times2}(\mathbf{R}_{2\times2} - \mathbf{t}_{2\times1}\mathbf{n}^T_{2\times1}/d)\mathbf{K}^{-1}_{2\times2}, \tag{3}$$

where $\mathbf{R}_{2\times2}$ can be computed from one pair of horizontal vanishing point correspondence, $\mathbf{t}_{2\times1}$ is measured by on-board sensors, and $\mathbf{n}_{2\times1}$ is a unitary vector. Only $\mathbf{n}_{2\times1}$ and d are the unknown variables to be estimated.

$\mathbf{H}_{2\times2}$ has 2 degrees of freedom. Let u_i and u'_i be the $u-$coordinates of vertical ideal lines l_i and l'_i, respectively. Denote $\mathbf{u}_i = [u_i, 1]^T$ and $\mathbf{u}'_i = [u'_i, 1]^T$ as the homogeneous coordinates. From the definition of homography, we know

$$\mathbf{u}'_i = \mathbf{H}_{2\times2}\mathbf{u}_i. \tag{4}$$

Define $H_{a,b}$ as the (a,b)-th entry of $\mathbf{H}_{2\times2}$. Opening (4) and combining the resulting two equations, we have

$$u_i H_{1,1} + H_{1,2} - u_i u'_i H_{2,1} - u'_i H_{2,2} = 0. \tag{5}$$

Eq. (5) has two unknown variables. Given 2 pairs of coplanar vertical ideal lines, we can compute the minimal solution of $\mathbf{H}_{2\times2}$ by solving the two equations. All coplanar vertical ideal lines can be found using RANSAC iteratively, which selects a subset of inliers to minimize the geometric error,

$$\sum_i d_g(\mathbf{u}_i, \hat{\mathbf{H}}^{-1}_{2\times2}\mathbf{u}'_i) + d_g(\mathbf{u}'_i, \hat{\mathbf{H}}_{2\times2}\mathbf{u}_i), \tag{6}$$

where $d_g(\cdot)$ denotes the geometric distance and $\hat{\mathbf{H}}_{2\times2}$ is the estimation of $\mathbf{H}_{2\times2}$. $\hat{\mathbf{H}}_{2\times2}$ is used to verify the cross view vertical ideal line correspondences. It is clear that a correctly corresponded ideal line pair must satisfy the homography. Hence we can refine the cross view vertical ideal line correspondences by removing false correspondences and searching for new correspondences to increase vertical ideal line inlier set for each plane. The estimation of $\hat{\mathbf{H}}_{2\times2}$ and the refinement process can be iterated until the cross view vertical ideal line correspondences are stable.

Correspondingly, the change of cross view vertical ideal line correspondence in the refinement process changes line segment correspondences in the lower layer. We refine vertical line segment correspondences, which is the second stage (step 9(b) in Fig. 3). Fig. 7(a) illustrates the sample output of this process.

$\hat{\mathbf{H}}_{2\times2}$ also allows us to compute vertical plane in the top-down view. Define $\mathbf{B} = \mathbf{K}_{2\times2}^{-1}\mathbf{H}_{2\times2}\mathbf{K}_{2\times2}$. According to (3), we can obtain d and $\mathbf{n}_{2\times1}$ as follows,

$$d = \frac{1}{\sqrt{(\frac{B_{1,1}R_{2,1}-B_{2,1}R_{1,1}}{B_{1,1}t_z-B_{2,1}t_x})^2+(\frac{B_{1,2}R_{2,2}-B_{2,2}R_{1,2}}{B_{1,2}t_z-B_{2,2}t_x})^2}}, \quad \mathbf{n}_{2\times1} = \begin{bmatrix} \frac{d(B_{1,1}R_{2,1}-B_{2,1}R_{1,1})}{B_{1,1}t_z-B_{2,1}t_x} \\ \frac{d(B_{1,2}R_{2,2}-B_{2,2}R_{1,2})}{B_{1,2}t_z-B_{2,2}t_x} \end{bmatrix}, \tag{7}$$

where $B_{a,b}$ and $R_{a,b}$ are the (a,b)-th entry of $\mathbf{B}$ and $\mathbf{R}_{2\times2}$, respectively.

2) Coplanarity Verification for Horizontal Features: Candidate vertical planes found in the previous step can be further refined using horizontal line features (see Fig. 7(b)). Horizontal line features can help remove false positive planes and improve accuracy of vertical planes. Furthermore, we can remove falsely matched horizontal features and group the inliers based on the vertical planes (step 10 in Fig. 3).

A pair of corresponding lines (l_i, l_i') in two views can be related by a 2D homography $l_i = \mathbf{H}_{3\times3}^T l_i'$. According to [40], the 2D homograph can be obtained as follows,

$$\mathbf{H}_{3\times3} = \mathbf{K}(\mathbf{R}_{3\times3} - \mathbf{t}_{3\times1}\mathbf{n}_{3\times1}^T/d)\mathbf{K}^{-1}, \tag{8}$$

where $\mathbf{R}_{3\times3}$ denotes the 3D camera rotation, $\mathbf{t}_{3\times1} = [t_x, t_y, t_z]^T$ is the translation between camera center positions of the two views, and $\mathbf{n}_{3\times1} = [n_x, n_y, n_z]$ is the normal vector of the plane. $n_y = 0$ because the plane is vertical.

Comparing $\mathbf{H}_{2\times2}$ in (3) and $\mathbf{H}_{3\times3}$ in (8), we find that $\mathbf{R}_{3\times3}$ and $\mathbf{n}_{3\times1}$ can be directly obtained from $\mathbf{R}_{2\times2}$ and $\mathbf{n}_{2\times1}$, respectively. t_x/d and t_z/d are both known from $\mathbf{H}_{2\times2}$. Given $\mathbf{H}_{2\times2}$, there is only one degree-of-freedom left in $\mathbf{H}_{3\times3}$. Therefore, the minimal solution of $\mathbf{H}_{3\times3}$ can be computed using one horizontal line correspondence and the resulting $\mathbf{H}_{2\times2}$ from the previous step. Defining $\tilde{\mathbf{H}}_{2\times2} = \mathbf{K}_{2\times2}^{-1}\mathbf{H}_{2\times2}\mathbf{K}_{2\times2}$, $\tilde{\mathbf{H}}_{3\times3} = \mathbf{K}^{-1}\mathbf{H}_{3\times3}\mathbf{K}$, we have

$$\tilde{\mathbf{H}}_{3\times3} = \begin{bmatrix} \tilde{H}_{1,1} & 0 & \tilde{H}_{1,2} \\ -n_x t_y/d & 1 & -n_z t_y/d \\ \tilde{H}_{2,1} & 0 & \tilde{H}_{2,2} \end{bmatrix}, \tag{9}$$

where $\tilde{H}_{a,b}$ is the (a,b)-th entry of $\tilde{\mathbf{H}}_{2\times2}$. In $\tilde{\mathbf{H}}_{3\times3}$, the only unknown is t_y. Note that we use $\tilde{a}$ indicate variable a is in normalized coordinates in this paper.

Define $\tilde{l}_i = \mathbf{K}^T l_i$ and $\tilde{l}_i' = \mathbf{K}^T l_i'$ where (l_i, l_i') is a corresponding pair of horizontal ideal lines. Denote $\tilde{l}_i = [a_1, a_2, a_3]^T$ and $\tilde{l}_i' = [a_1', a_2', a_3']^T$ as the homogeneous coordinates of $\tilde{l}$ and $\tilde{l}'$, respectively. Pair $(\tilde{l}_i, \tilde{l}_i')$ satisfies 2D homography by $\tilde{l}_i = \tilde{\mathbf{H}}_{3\times3}^T \tilde{l}_i'$. Thus

$$a_2 a_1' \tilde{H}_{1,2} - a_2 a_2' n_z t_y/d + a_2 a_3' \tilde{H}_{2,2} = a_3 a_2'. \tag{10}$$

Combining (9) and (10), we can obtain the minimal solution of $\mathbf{H}_{3\times3}$. The coplanar horizontal lines can be found using RANSAC iteratively. The homography-guided refinement of correspondences for horizontal ideal lines and line segments are similar to the process for vertical features.

Following the principle of the Gold Standard Algorithm approach in computer vision [40], the final inlier set of coplanar vertical and horizontal lines are used to re-estimate π_i's using MLE to improve accuracy. With π_i's obtained, this completes the construction of MFG.

VI. Experiments

We have implemented our MFG construction algorithm using Matlab 2008b on a laptop PC with an Intel 2.26Ghz Core 2 Duo CPU, 3GB RAM, a 160GB hard disk and a Windows XP OS. In the physical experiments, we use a BenQ DCE1035 camera with a resolution of 640×480 pixels. For the testing dataset, we have taken 8 different pairs of image on Texas A&M campus. For each pair, the baseline distance between two views is 5.5 m. The orientation settings of the camera are set to ensure a good overlap between the two views. Fig. 8 illustrates the first view of the 8 pairs. The algorithm has not been optimized yet. The average time for constructing an MFG from one image pair is 42 sec.

Fig. 8. Sample images of the dataset.

A. A Sample Output

Our algorithm has successfully constructed MFG. As a sample output, Fig. 9 illustrates the intermediate output of correctly matched line segments for the 8-th pair in Fig. 8. Four vertical planes have been identified in the figure.

Fig. 9. Sample output for the MFG construction. We use the same color (blue yellow, red, and green) for correctly matched line segments in the same residing vertical plane.

Testing the functionality is just the first step of experiments. Two more experiments have been conducted to verify MFG. The first experiment is to verify its robustness by checking if MFG can find better correspondence between

line segment features. This is important because the rest of MFG depends on the correspondence accuracy.

B. Robustness of Line Segment Matching

One key advantage of MFG is that it verifies line segments using multiple types of geometric relationships. If successful, the line segment correspondence is supposed to be more robust than the existing line segment matching method.

Since the best available method for line segment matching is the PBLM in [34] and the MFG construction algorithm has adopted the approach to find initial correspondence in Section V-B, the comparison essentially tries to find if our parallelism, collinearity, and coplanarity verifications can improve the matching results. The threshold of SIFT point matching in PBLM is set to 0.85. We compared the number of total matches (TM) and correct ratios (CR) between the two methods. Table I shows the matching performance using the aforementioned dataset. The last two columns of the table refer to how much TM and CR of MFG are more than those of PBLM, respectively. It can be seen that MFG can identify significantly more correctly matched line segments than those of PBLM. Also, CRs of MFG are larger than those of PBLM for all cases except the second pair.

TABLE I

LINE SEGMENT MATCHING RESULTS: PBLM VS. MFG.

No.	PBLM		MFG		TM difference	CR difference
	TM	CR	TM	CR		
1	224	93.3%	297	95.6%	73	2.3%
2	157	94.9%	289	92.0%	132	-2.9%
3	124	92.7%	178	96.2%	54	3.5%
4	186	93.5%	282	96.1%	96	2.6%
5	157	93.0%	274	95.3%	117	2.3%
6	219	93.6%	302	94.0%	83	0.4%
7	126	94.4%	189	94.7%	73	0.3%
8	194	92.3%	314	95.5%	120	3.2%

C. Vertical Plane Detection and Reconstruction

Our second experiment is to verify if the MFG construction algorithm can properly identify vertical planes and to exam the accuracy of vertical plane reconstruction. Denote $\hat{\pi}_i$ and $\bar{\pi}_i$ as the estimation from the MFG construction and the ground truth of vertical plane π_i, respectively. Ground truth $\bar{\pi}_i$ is obtained by using three non-collinear 3D points lying in π_i. The coordinates of the 3D points are obtained using a BOSCH GLR225 laser distance measurer with a range up to 70 m and measurement accuracy of $\pm 1.5mm$. Baseline distances are measured with a tape measure.

Directly comparing $\hat{\pi}_i$ to $\bar{\pi}_i$ is not meaningful because the result depends on coordinate system and unit selections. To avoid the problem, we utilize the 3D point reconstruction error in comparison. Define x_j as a 2D image point lying in π_i. With the aid of camera intrinsic parameters and plane equations, we can reconstruct this point from $\bar{\pi}_i$ and $\hat{\pi}_i$, respectively. Let $\bar{X}_j$ and $\hat{X}_j$ be the corresponding results. We define a relative error metric as $\frac{\|\bar{X}_j - \hat{X}_j\|}{\|\bar{X}_j\|}$ where $\|\cdot\|$ represents the Euclidean distance. For each vertical plane, we manually select 20 image feature points as even as possible to cover the whole plane region in the image. The mean value and standard deviation of the relative errors are shown in Table II for the dataset.

TABLE II

PERCENTILE RELATIVE ERRORS OF THE RECONSTRUCTED 3D POINTS.

No.	π_1		π_2		π_3		π_4	
	mean	std. dev.	mean	std. dev.	mean	std. dev.	mean	std. dev.
1	2.58	0.82	4.11	1.18				
2	3.33	0.48	3.16	0.88				
3	4.02	1.28	4.49	0.92				
4	4.10	1.03	4.67	0.41				
5	3.43	0.14	4.43	0.28	4.37	0.28		
6	5.18	0.74	4.02	0.64	2.64	0.44		
7	4.08	0.16	4.18	0.43	5.20	0.47		
8	4.88	0.29	3.00	0.48	4.41	0.15	6.01	0.26

Tab. II shows that the algorithm has the ability to identify vertical planes in the images, which results in the different numbers of vertical planes for the image pairs in the dataset. The relative errors of points on planes are reasonably small which indicates that the estimate planes are reasonably accurate. The MFG construction algorithm design is successful.

VII. CONCLUSION AND FUTURE WORK

We reported a multilayer feature graph (MFG) to facilitate the robot scene understanding in urban areas. Nodes of an MFG were features such as SIFT feature points, line segments, lines, and planes while edges of the MFG represented different geometric relationships such as adjacency, parallelism, collinearity, and coplanarity. MFG also connected the features in two views and the corresponding 3D coordinate system. Our MFG construction method was a feature fusion process which incrementally, iteratively, and extensively verified the aforementioned geometric relationships using the RANSAC framework. We implemented the MFG construction method and tested it in physical experiments. Results showed that MFG can be successfully established and the feature correspondence outcomes were significantly improved over existing feature matching methods.

The current result is just an initial step for MFG. In the future, we plan to address the N-View ($N > 2$) construction of MFG by combining bundle adjustment idea with error analysis into MFG. We will analyze computation complexity and develop efficient data structures for MFG. We will also investigate how to evolve MFG into an incremental update process so that it can be efficiently constructed for continuous image streams from mobile robots. Distributed and parallel implementation of MFG will be another direction to be explored. Applying MFG to GPS-less robot localization will be an immediate application if Google street view database is used. We will present these developments in future publications.

ACKNOWLEDGMENT

Thanks for C. Kim, W. Li, and H. Ge for their inputs and contributions to the Networked Robots Laboratory in Texas A&M University.

REFERENCES

[1] A. Elfes, "Sonar-based real-world mapping and navigation," *Robotics and Automation, IEEE Journal of*, vol. 3, no. 3, pp. 249 –265, june 1987.

[2] A. Diosi and L. Kleeman, "Laser scan matching in polar coordinates with application to slam," in *Intelligent Robots and Systems, 2005.(IROS 2005). IEEE/RSJ International Conference on*. Alberta, Canada: IEEE, Aug. 2005, pp. 3317–3322.

[3] V. Nguyen, A. Harati, A. Martinelli, R. Siegwart, and N. Tomatis, "Orthogonal slam: a step toward lightweight indoor autonomous navigation," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*. Beijing, China: IEEE, Oct. 2006, pp. 5007–5012.

[4] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments," in *the 12th International Symposium on Experimental Robotics, New Delhi & Agra, India*, Dec. 2010.

[5] E. Eade and T. Drummond, "Scalable monocular slam," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2006*, vol. 1. New York, NY, USA: IEEE Computer Society, June 2006, pp. 469–476.

[6] K. Konolige and M. Agrawal, "Frameslam: From bundle adjustment to real-time visual mapping," *Robotics, IEEE Transactions on*, vol. 24, no. 5, pp. 1066–1077, 2008.

[7] P. Smith, I. Reid, and A. Davison, "Real-time monocular slam with straight lines," in *British Machine Vision Conference*, vol. 1, Edinburgh, UK, Sep. 2006, pp. 17–26.

[8] T. Lemaire and S. Lacroix, "Monocular-vision based SLAM using line segments," in *IEEE International Conference on Robotics and Automation, ICRA 2007*. Roma, Italy: IEEE, April 2007, pp. 2791–2796.

[9] Y. Choi, T. Lee, and S. Oh, "A line feature based SLAM with low grade range sensors using geometric constraints and active exploration for mobile robot," *Autonomous Robots*, vol. 24, no. 1, pp. 13–27, 2008.

[10] L. Chen, T. Teo, Y. Shao, Y. Lai, and J. Rau, "Fusion of lidar data and optical imagery for building modeling," *International Archives of Photogrammetry and Remote Sensing*, vol. 35, no. B4, pp. 732–737, 2004.

[11] C. Rasmussen, "A hybrid vision + ladar rural road follower," in *IEEE International Conference on Robotics and Automation, Orlando, Florida*, May 2006, pp. 156–161.

[12] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.

[13] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," in *9th European Conference on Computer Vision (ECCV)*, Graz, Austria, May 2006, pp. 404–417.

[14] C. Frueh, S. Jain, and A. Zakhor, "Data processing algorithms for generating textured 3d building facade meshes from laser scans and camera images," *International Journal of Computer Vision*, vol. 61, no. 2, pp. 159–184, 2005.

[15] L. Zebedin, J. Bauer, K. Karner, and H. Bischof, "Fusion of feature- and area-based information for urban buildings modeling from aerial imagery," in *Proceedings of the 10th European Conference on Computer Vision: Part IV*. Springer-Verlag, 2008, pp. 873–886.

[16] N. Cornelis, B. Leibe, K. Cornelis, and L. Van Gool, "3d urban scene modeling integrating recognition and reconstruction," *International Journal of Computer Vision*, vol. 78, no. 2, pp. 121–141, 2008.

[17] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2006*. New York, NY, USA: IEEE Computer Society, June 2006.

[18] G. Vogiatzis, P. Torr, and R. Cipolla, "Multi-view stereo via volumetric graph-cuts," in *Computer Vision and Pattern Recognition, CVPR 2005. IEEE Computer Society Conference on*, vol. 2. San Diego, CA, USA: IEEE, June 2005, pp. 291–298.

[19] H. Jin, S. Soatto, and A. Yezzi, "Multi-view stereo reconstruction of dense shape and complex appearance," *International Journal of Computer Vision*, vol. 63, no. 3, pp. 175–189, 2005.

[20] J. Pons, R. Keriven, and O. Faugeras, "Modelling dynamic scenes by registering multi-view image sequences," in *Computer Vision and Pattern Recognition, CVPR 2005. IEEE Computer Society Conference on*, vol. 2. San Diego, CA, USA: IEEE, June 2005, pp. 822–827.

[21] H. Bay, V. Ferrari, and L. V. Gool, "Wide-baseline stereo matching with line segments," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, June 2005, pp. 329 – 336.

[22] T. Yu, N. Xu, and N. Ahuja, "Shape and view independent reflectance map from multiple views," *International journal of computer vision*, vol. 73, no. 2, pp. 123–138, 2007.

[23] D. Bradley, T. Boubekeur, and W. Heidrich, "Accurate multi-view reconstruction using robust binocular stereo and surface meshing," in *Computer Vision and Pattern Recognition (CVPR), 2008 IEEE Conference on*. Anchorage, AK: IEEE, June 2008.

[24] D. Gallup, J. Frahm, and M. Pollefeys, "Piecewise planar and non-planar stereo for urban scene reconstruction," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. San Francisco, USA: IEEE, July 2010, pp. 1418–1425.

[25] T. Cham, A. Ciptadi, W. Tan, M. Pham, and L. Chia, "Estimating camera pose from a single urban ground-view omnidirectional image and a 2D building outline map," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. San Francisco, USA: IEEE, July 2010, pp. 366–373.

[26] J. Delmerico, P. David, M. Adelphi, and J. Corso, "Building facade detection, segmentation, and parameter estimation for mobile robot localization and guidance," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*. San Francisco, California, USA: IEEE, Sep. 2011.

[27] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[28] C. Schmid and A. Zisserman, "The geometry and matching of lines and curves over multiple views," *International Journal of Computer Vision*, vol. 40, no. 3, pp. 199–233, 2000.

[29] J. Guerrero and C. Sagues, "Robust line matching and estimate of homographies simultaneously," *Pattern Recognition and Image Analysis*, pp. 297–307, 2003.

[30] H. Bay, V. Ferrari, and L. Van Gool, "Wide-baseline stereo matching with line segments," in *Proc. IEEE Int'l Conf. on Computer Vision and Pattern Recognition*. San Diego, CA, USA: IEEE Computer Society, June 2005.

[31] Z. Wang, F. Wu, and Z. Hu, "Msld: A robust descriptor for line matching," *Pattern Recognition*, vol. 42, no. 5, pp. 941–953, May 2009.

[32] M. Lourakis, S. Halkidis, S. Orphanoudakis *et al.*, "Matching disparate views of planar surfaces using projective invariants," *Image and Vision Computing*, vol. 18, no. 9, pp. 673–683, 2000.

[33] Y. Deng and X. Lin, "A fast line segment based dense stereo algorithm using tree dynamic programming," in *Proc. Eur. Conf. on Computer Vision*. Graz, Austria: Springer, May 2006, pp. 201–212.

[34] B. Fan, F. Wu, and Z. Hu, "Line matching leveraged by point correspondences," in *Proc. IEEE Int'l Conf. on Computer Vision and Pattern Recognition*. San Francisco, USA: IEEE, June 2010, pp. 390–397.

[35] D. Song, H. Lee, J. Yi, and A. Levandowski, "Vision-based motion planning for an autonomous motorcycle on ill-structured roads," *Autonomous Robots*, vol. 23, no. 3, pp. 197–212, Oct. 2007.

[36] D. Song, H. Lee, and J. Yi, "On the analysis of the depth error on the road plane for monocular vision-based robot navigation," in *The Eighth International Workshop on the Algorithmic Foundations of Robotics, Guanajuato, Mexico, Dec. 7-9*, 2008.

[37] J. Zhang and D. Song, "On the error analysis of vertical line pair-based monocular visual odometry in urban area," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*. St. Louis, USA: IEEE, Oct. 2009, pp. 3486–3491.

[38] J. Zhang and D.Song, "Error aware monocular visual odometry using vertical line pairs for small robots in urban areas," in *Special Track on Physically Grounded AI, the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10)*, Atlanta, Georgia, USA, July 2010.

[39] R. von Gioi, J. Jakubowicz, J. Morel, and G. Randall, "LSD: A Fast Line Segment Detector with a False Detection Control," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 4, pp. 722–732, 2010.

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

[41] J. Leung and G. NcLean, "Vanishing point matching," in *Image Processing, 1996. Proceedings., International Conference on*, vol. 1. Lausanne, Switzerland: IEEE, Sep. 1996, pp. 305–308.