High Level Landmark-Based Visual Navigation Using Unsupervised Geometric Constraints in Local Bundle Adjustment

Yan Lu, Dezhen Song, and Jingang Yi

***Abstract*— We present a high level landmark-based visual navigation approach for a monocular mobile robot. We utilize heterogeneous features, such as points, line segments, lines, planes, and vanishing points, and their inner geometric constraints as the integrated high level landmarks. This is managed through a multilayer feature graph (MFG). Our method extends local bundle adjustment (LBA)-based framework by explicitly exploiting different features and their geometric relationships in an unsupervised manner. The algorithm takes a video stream as input, initializes and incrementally updates MFG based on extracted key frames; it also refines localization and MFG landmarks through the LBA. Physical experiments show that our method can reduce the absolute trajectory error of a traditional point landmark-based LBA method by up to 63.9%.**

I. Introduction

Visual navigation using low cost cameras, such as cameras in mobile devices like cell phones and tablets, has gained more research attention due to the increasing needs for navigation assistance in indoor and/or GPS challenged environments. Visual navigation is often conducted under the simultaneous localization and mapping (SLAM) framework. Despite its great success, visual SLAM still suffers from technical issues such as scale drift and robustness to dynamic environment. Besides the limitations of camera itself (as a bearing-only sensor), another possible reason is that most systems use low level features (e.g. salient points) as sole landmarks. This may not be the best choice when higher level landmarks are available in man-made environments where abundant lines in parallel directions and the salient building facades exist. These higher level landmarks not only enable the possibility of higher level tasks such as object recognition and human-robot interaction, but also can potentially help improve navigation performance.

We utilize heterogeneous visual features, including points, line segments, lines, planes, and vanishing points, and their inner geometric constraints as the integrated high level landmarks to assist robot navigation (see Fig. 1). This is managed through a multilayer feature graph (MFG), an open data structure containing geometric relationships, such as parallelism and coplanarity. Our method extends local bundle adjustment (LBA)-based framework by explicitly exploiting different features and their geometric relationships in an unsupervised manner. The algorithm takes a video stream as input, initializes and incrementally updates and expands MFG based on extracted key frames, and refines localization and MFG landmarks through the LBA. Physical experiments show that our method can reduce the mean absolute trajectory error of a traditional point landmark-based LBA method by up to 63.9%.

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

J. Yi is with the Department of Mechanical and Aerospace Engineering, Rutgers University, Piscataway, NJ 08854, USA. E-mail:*jgyi@rutgers.edu.*

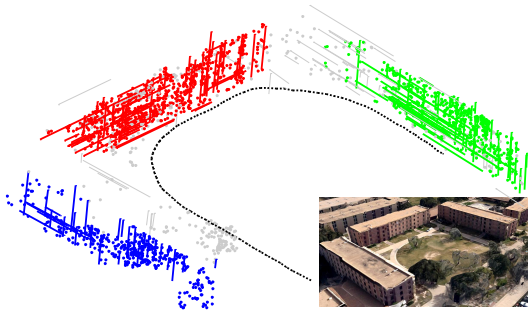


Fig. 1. Output of our algorithm, and a Google Earth™ view of the same site from a similar perspective. Coplanar landmarks (points and lines) are coded in the same color, while general landmarks are in gray color. The dotted line is the estimated camera trajectory.

II. Related Work

Our work is an extension of SLAM problems, and visual SLAM in particular [1].

There are two prevalent methodologies in visual SLAM: the bundle adjustment (BA) approaches (e.g., [2]) rooted in the structure from motion (SFM) area in computer vision, and the filtering methods (e.g. [3]) originated from the traditional SLAM field of robotics research. Strasdat et al. have analyzed the advantages of each method in [4]. For both methods, various camera types/modalities have been studied, such as a monocular camera [5], [6], a stereo camera [7], an omnidirectional camera [8], and an RGB-D camera [9].

Besides methodology and sensor configuration, another critical issue in visual SLAM is environment representation. For example, point cloud [10] and sparse feature points [11] are often employed as landmarks in a map. Recently, many researchers have realized that landmark selection is an important factor in visual odometry and SLAM performance. Lower level landmarks such as Harris corner and SIFT point, are relatively easy to use due to their geometric simplicity, which shares many properties with traditional point clouds generated from laser range finders. However, point features are merely mathematical singularities in color, texture, and geometric space. They can also be easily influenced by lighting and shadow conditions. To overcome these shortcomings, higher level landmarks have received more and more attention for visual SLAM, such as line segments [12] and planes [13].

These works have demonstrated the advantages of higher level landmarks in robustness and accuracy, but they either treat these landmarks as isolated objects, or partially explore the inner relationship between them. This treatment simplifies the SLAM problem formulation but cannot fully utilize the power of high level landmarks. Very recently, Tretyak et al. present an optimization framework for geometric parsing of image by jointly using edges, line segments, lines, and vanishing points [14]. However, this method limits itself to a single image for now. At almost the same time, Li et al. propose the initial MFG concept based on two views [15], which is then applied to building exterior mapping under an EKF framework [16]. Inspired by [4], we present an LBA-based approach to constructing MFG from a video stream.

III. Background and Problem Formulation

A. Assumptions and Notations: Consider a monocular robot navigating in a previously unknown environment. Assume:

- **a.1** The robot operates in a largely static environment with rectilinear structures, a characteristic of typical man-made environments, and
- **a.2** The camera is calibrated with radial distortion removed.

Let us define the following notations,

$\mathcal{V}$	Input camera video,
$\{W\}$	3D Cartesian world coordinate system,
I_k	A key frame extracted from $\mathcal{V}$, $I_k \in \mathcal{V}, k \in \mathbb{N}$,
$\{C_k\}$	Camera coordinate system at I_k,
K	Camera calibration matrix,
R_k	Camera rotation matrix at I_k,
t_k	Camera translation vector at I_k,
P_k	Camera projection matrix, $\mathrm{P}_k = \mathrm{K}\,[\mathrm{R}_k \,|\, \mathrm{t}_k]$,
$X_{i:j}$	Collection defined as $X_{i:j} = \{X_k, i \le k \le j\}$,
$\mathcal{M}_k$	MFG constructed based on key frames $I_{0:k}$,
$\mathbb{E}^n$	n-dimensional Euclidean space,
$\mathbb{P}^n$	n-dimensional projective space, and
$\mathbf{X}$	A homogeneous vector, $\mathbf{X} = [\tilde{\mathbf{X}}^\mathsf{T}, 1]^\mathsf{T}$, where $\tilde{\mathbf{X}}$ denotes the inhomogeneous, counterpart of $\mathbf{X}$. $\mathbf{X} \in \mathbb{P}^n \Rightarrow \tilde{\mathbf{X}} \in \mathbb{E}^n$.

We abuse "$=$" to denote real and up-to-scale equalities for inhomogeneous and homogeneous vectors, respectively.

B. Review of Multilayer Feature Graph: Introduced in [15], MFG is the key data structure for organizing landmarks. Fig. 2 shows the structure of MFG, composed of five types of features in separate layers with four kinds of geometric constraints. Let us briefly review MFG for completeness.

1) A **key point** node represents a 3D point landmark. We denote its 3D position by $\mathbf{P}_i \in \mathbb{P}^3$ and its image observation in I_k by $\mathbf{p}_{i,k} \in \mathbb{P}^2$.

2) A **line segment** node represents a 3D segment object. We denote it in 3D by $\mathbf{S}_i = [\mathbf{D}_{i1}^\mathsf{T}, \mathbf{D}_{i2}^\mathsf{T}]^\mathsf{T}$, where $\mathbf{D}_{i1}$ and $\mathbf{D}_{i2}$ are the two endpoints. Its observation in I_k is then $\mathbf{s}_{i,k} = [\mathbf{d}_{i1,k}^\mathsf{T}, \mathbf{d}_{i2,k}^\mathsf{T}]^\mathsf{T}$.

3) An **ideal line** node represents an infinite 3D line. We denote an ideal line in 3D by $\mathbf{L}_i = [\mathbf{Q}_i^\mathsf{T}, \mathbf{J}_i^\mathsf{T}]^\mathsf{T}$, where $\mathbf{Q}_i$ is a finite 3D point locating on $\mathbf{L}_i$ and $\mathbf{J}_i$ is an infinite 3D point defining the direction of $\mathbf{L}_i$. The image of $\mathbf{L}_i$ in I_k is denoted by $\mathbf{l}_{i,k}$. Image ideal line $\mathbf{l}_{i,k}$ is detected by identifying collinear image line segment(s). Thus, an ideal line has a set of supporting line segments.

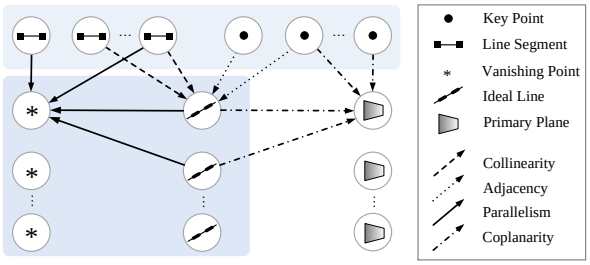


Fig. 2. MFG structure. All nodes exist in 3D space, and the shaded regions indicate nodes which also exist in 2D image. Geometric relationships between nodes are represented by edges of different line types.

4) A **primary plane** node represents a planar 3D object (e.g., a wall). We denote a primary plane by $\mathbf{\Pi}_i = [\mathrm{n}_i^\mathsf{T}, d_i]^\mathsf{T}$ in 3D, where $\mathrm{n}_i \in \mathbb{E}^3$ and $d_i \in \mathbb{R}$, such that $\mathbf{X}^\mathsf{T}\mathbf{\Pi}_i = 0$ for any point $\mathbf{X}$ on the plane.

5) A **vanishing point** node represents a particular 3D direction. We denote a vanishing point by $\mathbf{V}_i$ in 3D and its observation in I_k by $\mathbf{v}_{i,k}$.

Besides, the geometric relationships between these nodes are represented by MFG edges connecting them, including *parallelism*, *coplanarity*, *collinearity* and *adjacency* (see Fig. 2).

C. Problem Formulation: I_0 and I_1 are given along with an initial MFG $\mathcal{M}_1$ from [15]. For $k \ge 2$, the problem is:

Definition 1: Given video $\mathcal{V}$, MFG $\mathcal{M}_{k-1}$, and historical camera poses $\{\mathrm{R}_{0:k-1}, \mathrm{t}_{0:k-1}\}$, select key frame I_k, estimate camera pose $\{\mathrm{R}_k, \mathrm{t}_k\}$, and update the nodes and edges of $\mathcal{M}_{k-1}$ to attain $\mathcal{M}_k$.

IV. System Design

Our system architecture is illustrated in Fig. 3, where the main blocks are shaded and explained in this section.

A. Key Frame Selection: Given a video input, it is necessary to select a set of key frames for motion estimation. This is to guarantee sufficient baseline distance between two frames and avoid ill-posed epipolar geometry problems. The basic principle is to find a good balance between two needs: a) large camera movement to provide sufficient motion parallax and b) sufficient overlap of scene. Based on existing methods [2], [5], [8], we make the following criteria for key frame selection. Supposing I_{k-1} and $\mathcal{M}_{k-1}$ are given, a video frame is chosen as key frame I_k if it satisfies: 1) there are as many video frames between I_{k-1} and I_k as possible, 2) the number of SIFT point correspondences between I_{k-1} and I_k is no less than N_{2d} ($N_{2d} = 50$ here), and 3) the number of MFG key points that are observable in I_k is not less than N_{3d} ($N_{3d} = 5$ here). Since I_0 and I_1 are manually selected by user, the selection criteria only apply for $k \ge 2$.

B. Image Feature Processing: Once I_k is selected, we proceed to extract image features from it in the *image feature processing* step. 2D key points and line segments are extracted from I_k using SIFT and LSD [17], respectively, and 2D ideal lines and vanishing points are detected based on the resulting line segments. The correspondences of these

Fig. 3. System Diagram

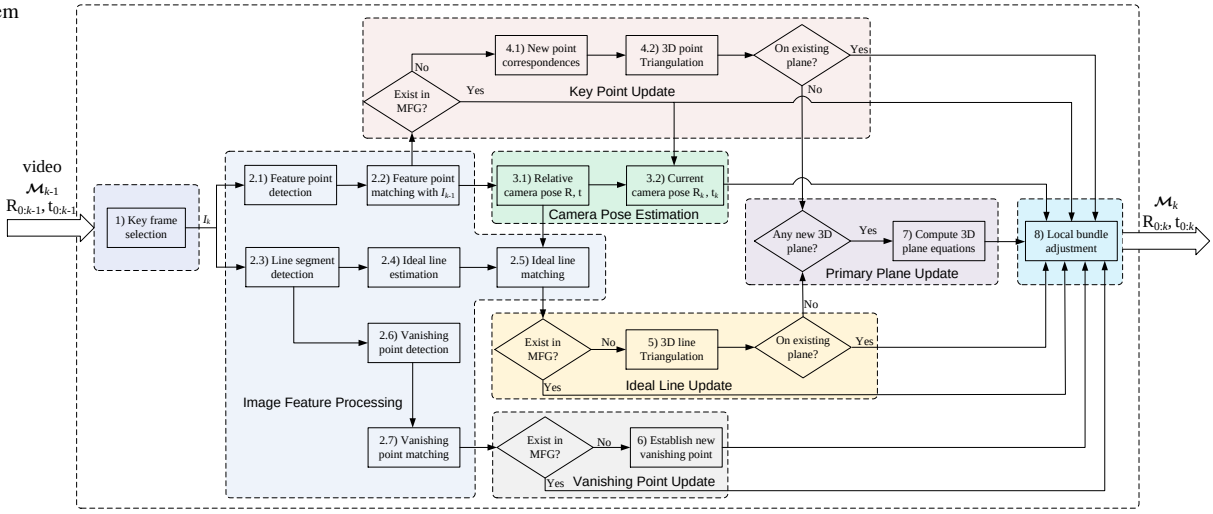


image features are then found between I_{k-1} and I_k. Detailed methods in this process can be found in [15].

Remark 1: All features detected from I_k only exist in the image space so far, and they will be associated with 3D landmarks, or used to establish new 3D landmarks in the MFG update step.

C. Camera Pose Estimation: With image feature correspondences obtained, estimating the 6 degrees of freedom (DoF) camera pose R_k and t_k for I_k is a key step for inferring 3D information and updating MFG. Existing methods (e.g. [18]) usually solve this problem using 3-point algorithms [19] based on the 3D-2D correspondences $\{\mathbf{P}_i \leftrightarrow \mathbf{p}_{i,k}\}$ between known 3D points and their observations in I_k. This method omits the 2D-2D correspondences between I_{k-1} and I_k whose 3D positions are unknown yet. To fully use both kinds of information, Tardif et al. propose [8] decoupling the estimation of R_k from t_k. We modify the method as below.

Step 1: Based on SIFT point correspondences between I_{k-1} and I_k, compute essential matrix E using the 5-point algorithm in RANSAC [20]. Decompose E to recover the relative rotation R and translation t, with $\|\mathrm{t}\|$ unknown.

Step 2: Compute $\|\mathrm{t}\|$ using 3D-2D correspondences through a RANSAC process using only one correspondence for a minimal solution. This completes the 6 DoF estimation.

In the Step 2 of [8], Tardif et al. estimate the full 3 DoFs of t using two 3D-2D correspondences for a minimal solution. This difference can be justified by the different cameras we use - an omnidirectional camera in [8] with $360°$ horizontal field of view (HFOV) vs. a regular camera we use with $40°\sim 80°$ HFOV. Narrower HFOV results in fewer observable 3D landmarks in view and thus fewer 3D-2D correspondences, especially in a turning situation. Therefore, we choose to reduce the problem dimension in Step 2 to fit our needs.

D. MFG Update and LBA: MFG update includes key point update, ideal line update, vanishing point update, and primary plane update, as shown in Fig. 3. Updating MFG basically means associating existing MFG nodes with their latest observations if available, and establishing new MFG nodes and edges based on image features. The update algorithms for each type of feature are presented in Sections V-A.1, V-A.2, V-A.3, and V-A.4, respectively. Following MFG update, an LBA is performed to jointly refine recent camera poses and MFG nodes using a window of key frames. Due to the uniqueness of MFG, we propose a new LBA formulation that integrates the heterogeneous features in MFG along with the embedded geometric constraints in Section V-B.

V. Algorithms

We begin with the MFG update algorithms for each type of feature, and then details the MFG-based LBA formulation.

A. MFG Update

Updating MFG involves associating image features with existing 3D landmarks and augmenting MFG by setting up new nodes (landmarks) and edges (geometric relationships), which is detailed below.

*1) **Key Point Update**:* Key point update is similar to traditional point-based SFM methods. We briefly describe our algorithm for completeness. For a 2D point correspondence $\mathbf{x}_{i,k-1} \leftrightarrow \mathbf{x}_{i,k}$ between I_{k-1} and I_k,

- if it is a re-observation of key point $\mathbf{P}_j$, let $\mathbf{p}_{j,k} = \mathbf{x}_{i,k}$.
- if it is a newly discovered point, compute its motion parallax $\rho(\mathbf{x}_{i,k-1}, \mathbf{x}_{i,k})$ using (1). If $\rho(\mathbf{x}_{i,k-1}, \mathbf{x}_{i,k})$ is greater than a threshold τ_p, we compute its 3D position and add it to $\mathcal{M}_k$ as a new key point node. Otherwise, we start a new image point track $\mathcal{Q}_j = \{\mathbf{x}_{i,k-1}, \mathbf{x}_{i,k}\}$ to keep track of it in future frames.
- if it is an observation of an image point track $\mathcal{Q}_j$, append it to the track $\mathcal{Q}_j = \mathcal{Q}_j \cup \{\mathbf{x}_{i,k}\}$, and check whether $\mathcal{Q}_j$ can be converted to a key point node. To do this, we compute the motion parallax between each pair of points in $\mathcal{Q}_j$, and if anyone is larger than τ_p, a new key point node is established and added to $\mathcal{M}_k$.

*2) **Ideal Line Update**:* Before presenting the ideal line update algorithm, we need to define the motion parallax for ideal lines first. It is well known that the motion parallax of a point correspondence $\mathbf{x}_{i,k-1} \leftrightarrow \mathbf{x}_{i,k}$ can be defined as

$$\rho(\mathbf{x}_{i,k-1}, \mathbf{x}_{i,k}) := \langle \mathrm{K}^{-1}\mathrm{H}_\mathrm{r}\mathbf{x}_{i,k-1}, \mathrm{K}^{-1}\mathbf{x}_{i,k}\rangle, \ \mathrm{H}_\mathrm{r} = \mathrm{K}\mathrm{R}\mathrm{K}^{-1} \tag{1}$$

where H_r represents a rotational homography [21], $\langle\cdot,\cdot\rangle$ indicates the angle between two vectors, and R is the relative

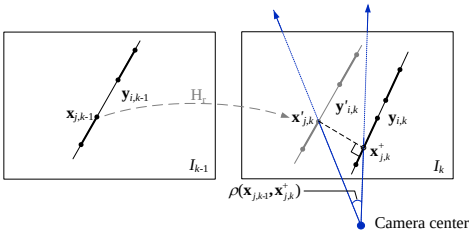


Fig. 4. Illustration of parallax computation for image ideal lines. $\mathrm{H_r}$ is a rotational homography mapping defined in (1). Bold lines are supporting line segments of the underlying (thin) ideal line. $\rho(\mathbf{x}_{j,k-1},\mathbf{x}^+_{j,k})$ is defined as the motion parallax for point pair $\mathbf{x}_{j,k-1}\leftrightarrow\mathbf{x}^+_{j,k}$.

rotation between I_{k-1} and I_k. Generally speaking, motion parallax has not been clearly defined for lines. Here we propose a heuristic motion parallax measurement for ideal lines by leveraging the line segment endpoints on them. For an image ideal line correspondence $\mathbf{y}_{i,k-1}\leftrightarrow\mathbf{y}_{i,k}$, define

$$\varrho(\mathbf{y}_{i,k-1},\mathbf{y}_{i,k}) := \frac{1}{n}\sum_{j=1}^{n}\rho(\mathbf{x}_{j,k-1},\mathbf{x}^+_{j,k}), \tag{2}$$

where $\{\mathbf{x}_{j,k-1}, j=1,\cdots,n\}$ are the endpoints of image line segments that support $\mathbf{y}_{i,k-1}$, and $\mathbf{x}^+_{j,k}$ is the perpendicular foot of $\mathbf{x}'_{j,k}:=\mathrm{H_r}\mathbf{x}_{j,k-1}$ on $\mathbf{y}_{i,k}$ in I_k (see Fig. 4). The rationale is that we want to reward line correspondences which have larger distance in their perpendicular direction. If $\mathbf{y}'_{i,k}$ overlap with $\mathbf{y}_{i,k}$, their motion parallax should be zero.

With the motion parallax defined, the ideal line update can be performed in a similar fashion to the key point case, and thus skipped here.

Remark 2: Line segment nodes are also updated in this process. Since a line segment always has an ideal line parent, when an image ideal line is converted to a node, its associated line segments are also converted to line segment nodes. Their 3D positions are computed from the 3D ideal line parameters.

*3) **Vanishing Point Update**:* Updating vanishing point nodes is straightforward. Given an image vanishing point correspondence $\boldsymbol{\nu}_{i,k-1}\leftrightarrow\boldsymbol{\nu}_{i,k}$, if it is a re-observation of existing node $\mathbf{V}_j$, let $\mathbf{v}_{j,k}=\boldsymbol{\nu}_{i,k}$. Otherwise, establish a new vanishing point node $\mathbf{V}_j=[\boldsymbol{\nu}_{i,k}^\mathsf{T}\mathrm{R}_k,0]^\mathsf{T}$. It is trivial but important to update the edges between ideal lines and vanishing points whenever a new ideal line or vanishing point node is added.

*4) **Primary Plane Update**:* Finding new primary planes relies on detecting coplanar key points and ideal lines. Here we detect primary planes directly from 3D key points and ideal lines using RANSAC. To be specific, let $\mathcal{C}$ be the collection of 3D key points and ideal lines which are not yet associated with any primary plane. We brief two key steps of the RANSAC process below.

1) Compute a plane candidate $\boldsymbol{\Gamma}$ from a minimal solution set, which could include either 3 key points, or 2 parallel ideal lines, or 1 key point plus 1 ideal line.
2) $\forall c\in\mathcal{C}$, compute score $f(\boldsymbol{\Gamma},c)$. If c is a key point, $f(\boldsymbol{\Gamma},c)$ is the perpendicular distance from c to $\boldsymbol{\Gamma}$; if c is an ideal line, $f(\boldsymbol{\Gamma},c)$ is the average of the distances from its associated line segment endpoints to $\boldsymbol{\Gamma}$.

If the size of the largest consensus set is greater than a threshold N_{plane}, we add the corresponding plane candidate to $\mathcal{M}_k$ as a primary plane node, and establish edges between it and the key points and ideal lines in the consensus set. Moreover, when new key point or ideal line nodes are established, we check if they belong to existing primary planes similarly.

B. Local Bundle Adjustment

After MFG is updated, we want to refine the estimated camera pose and MFG nodes simultaneously using LBA. Along the lines of [8], we use w latest key frames to bundle adjust m latest camera poses and MFG nodes established since I_{k-m+1}, with $w\geq m$ usually. To account for the various feature types and geometric constraints in MFG, we need to define cost functions accordingly.

*1) **Key Point**:* Denote the re-projection of key point $\mathbf{P}_i$ in I_k by $\hat{\mathbf{p}}_{i,k}:=\mathrm{P}_k\mathbf{P}_i$. We assume zero-mean Gaussian noise for image point measurement, i.e., $\tilde{\hat{\mathbf{p}}}_{i,k}\sim\mathcal{N}(\mathbf{0},\Lambda_\mathbf{p})$. Define the cost function for $\mathbf{P}_i$ in I_k to be

$$\mathcal{C}_{\text{pt}}(\mathbf{P}_i,k)=(\tilde{\hat{\mathbf{p}}}_{i,k}-\tilde{\mathbf{p}}_{i,k})^\mathsf{T}\Lambda_\mathbf{p}^{-1}(\tilde{\hat{\mathbf{p}}}_{i,k}-\tilde{\mathbf{p}}_{i,k}). \tag{3}$$

*2) **Ideal Line & Collinearity**:* Denote the re-projection of ideal line $\mathbf{L}_i$ in I_k by $\hat{\mathbf{l}}_{i,k}:=\mathrm{P}_k\mathbf{Q}_i\times\mathrm{P}_k\mathbf{J}_i$. Since the observation of $\mathbf{L}_i$ in I_k, i.e. $\mathbf{l}_{i,k}$, is estimated from its supporting line segments $\{\mathbf{s}_{\iota,k}|\iota=1,\cdots\}$, we directly treat these line segments as its observations for cost function definition. The measurement noise of image line segment can be modeled in various ways. Here we adopt a simple but well-accepted modeling, which assumes each line segment endpoint is subject to a zero-mean Gaussian noise, i.e., $\tilde{\mathbf{d}}_{\iota j,k}\sim\mathcal{N}(\mathbf{0},\sigma_d^2\mathbf{I}_2)$, where $j=1,2$, and $\mathbf{I}_2$ is a 2×2 identity matrix. Define the cost function for $\mathbf{L}_i$ in I_k as

$$\mathcal{C}_{\text{ln}}(\mathbf{L}_i,k)=\sum_\iota\sum_{j=1}^{2}\left(\frac{d_\perp(\tilde{\mathbf{d}}_{\iota j,k},\hat{\mathbf{l}}_{i,k})}{\sigma_d}\right)^2, \tag{4}$$

where $d_\perp(\cdot,\cdot)$ denotes the perpendicular distance from a point to a line in image. This cost function effectively captures the *collinearity* constraint between ideal lines and line segments.

*3) **Vanishing Point & Parallelism**:* Let the re-projection of vanishing point $\mathbf{V}_i$ in I_k be $\hat{\mathbf{v}}_{i,k}:=\mathrm{P}_k\mathbf{V}_i$. The observation of $\mathbf{V}_i$ in I_k is $\mathbf{v}_{i,k}$ which is the intersection of image line segments from the same parallel group. Since $\mathbf{v}_{i,k}$ is estimated from line segments, its estimation covariance $\Lambda_{\mathbf{v}_{i,k}}$ can be easily derived as well [22]. Define the cost function for $\mathbf{V}_i$ in I_k by

$$\mathcal{C}_{\text{vp}}(\mathbf{V}_i,k)=(\hat{\mathbf{v}}_{i,k}-\mathbf{v}_{i,k})^\mathsf{T}\Lambda_{\mathbf{v}_{i,k}}^{-1}(\hat{\mathbf{v}}_{i,k}-\mathbf{v}_{i,k}). \tag{5}$$

In particular, for all ideal lines $\{\mathbf{L}_j\}$ connected to $\mathbf{V}_i$ in MFG, we enforce $\mathbf{L}_j=[\mathbf{Q}_j^\mathsf{T},\mathbf{V}_i^\mathsf{T}]^\mathsf{T}$ such that these lines are strictly parallel. Recall that $\mathbf{Q}_j$ is a finite point on $\mathbf{L}_j$. This parameterization and cost function together account for the *parallelism* relationship in MFG.

*4) **Primary Plane & Coplanarity**:* Primary plane $\boldsymbol{\Pi}_i$ has neither re-projection nor direct observation in image space. Therefore, we define its cost function by leveraging 3D key points and ideal lines, respectively. For key point $\mathbf{P}_j$ and primary plane $\boldsymbol{\Pi}_i$, define

$$\mathcal{C}_{\text{pl}}(\mathbf{P}_j, \mathbf{\Pi}_i) = \begin{cases} [\delta_\perp(\mathbf{P}_j, \mathbf{\Pi}_i)]^2 & \text{if } \mathbf{P}_j \in \mathbf{\Pi}_i \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

where $\delta_\perp(\cdot,\cdot)$ denotes the perpendicular distance from a point to a plane in 3D, and $\mathbf{P}_j \in \mathbf{\Pi}_i$ indicates that $\mathbf{P}_j$ is connected with $\mathbf{\Pi}_i$ in MFG.

For ideal line $\mathbf{L}_j$ and primary plane $\mathbf{\Pi}_i$, define

$$\mathcal{C}_{\text{pl}}(\mathbf{L}_j, \mathbf{\Pi}_i) = \begin{cases} \frac{1}{n}\sum_{\iota=1}^{n}[\delta_\perp(\mathbf{D}_\iota, \mathbf{\Pi}_i)]^2 & \text{if } \mathbf{L}_j \in \mathbf{\Pi}_i \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

where $\{\mathbf{D}_\iota, \iota = 1, \cdots, n\}$ denote the endpoints of all the line segments that support $\mathbf{L}_j$. Eqs. (6) and (7) represent the *coplanarity* constraint in MFG.

*5) **Overall Metric**:* Denote the last m camera poses by $\mathcal{S}_{\text{cp}}^k = \{\mathrm{R}_i, \mathrm{t}_i | i = k-m+1, \cdots, k\}$, and the last m key frames by $\mathcal{I}^k = \{I_i | i = k-m+1, \cdots, k\}$. The key points to be refined in LBA are those that can be observed in at least one frame of $\mathcal{I}^k$, and we denote them by $\mathcal{S}_{\text{pt}}^k$. Similarly we define $\mathcal{S}_{\text{ln}}^k$ and $\mathcal{S}_{\text{vp}}^k$ for ideal lines and vanishing points, respectively. The primary planes to be refined are those that have edges connected to key points from $\mathcal{S}_{\text{pt}}^k$ or ideal lines from $\mathcal{S}_{\text{ln}}^k$, and we denote them by $\mathcal{S}_{\text{pl}}^k$. Then the total cost function is defined in (8) in next page.

The MFG-LBA problem at time k is

$$\min_{\mathcal{S}_{\text{cp}}^k, \mathcal{S}_{\text{pt}}^k, \mathcal{S}_{\text{ln}}^k, \mathcal{S}_{\text{vp}}^k, \mathcal{S}_{\text{pl}}^k} \mathcal{C}_{\text{total}}(\mathcal{M}_k). \quad (9)$$

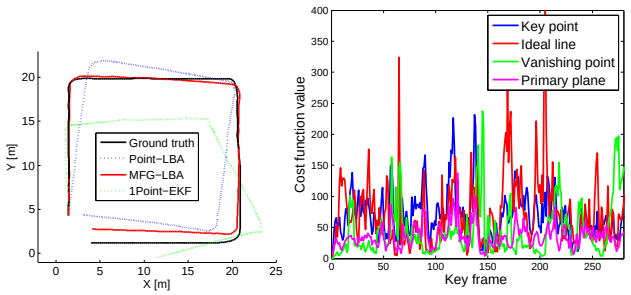
This problem can be solved using the Levenberg-Marquardt algorithm [21], and the solution provides refined camera poses $\mathcal{S}_{\text{cp}}^k$ and MFG nodes including key points $\mathcal{S}_{\text{pt}}^k$, ideal lines $\mathcal{S}_{\text{ln}}^k$, vanishing points $\mathcal{S}_{\text{vp}}^k$, and primary planes $\mathcal{S}_{\text{pl}}^k$.

VI. Experiments

We have implemented our algorithm using C++ in Windows 7. To validate the algorithm, we have compared it with state-of-the-art methods on different datasets. We choose the 1-Point RANSAC-based EKF-SLAM method [1] for comparison as it is state of the art in monocular visual navigation. Its MATLAB® code is available at the author's webpage [23]. In the following, we will refer to it as 1Point-EKF for brevity. The second method for comparison only uses SIFT points as landmarks and performs LBA as well, named Point-LBA. This method can be regarded as a degenerate version of MFG. Our proposed method is named MFG-LBA. It is worth mentioning that no loop closing is performed.

A. Indoor Experiments

*1) **Evaluation Metric**:* To evaluate the localization accuracy, we adopt the widely used absolute trajectory error (ATE) [1]. Since the ground truth and the estimation of camera poses are usually represented in different coordinate systems, we need to align them before computing ATE. Let $\mathbf{g}_k^{W'}$ be the ground truth of camera position at time k in a coordinate system $\{W'\}$ and $\mathbf{r}_k^W$ the estimated one in $\{W\}$. We need to find a similarity transformation that maps $\mathbf{r}_k^W$ to $\{W'\}$: $\mathbf{r}_k^{W'} := s\mathbf{R}_W^{W'}\mathbf{r}_k^W + \mathbf{t}_W^{W'}$, where the transformation is defined by rotation matrix $\mathbf{R}_W^{W'}$, translation vector $\mathbf{t}_W^{W'}$ and scaling factor s. Part of the transformation parameters (e.g. $\mathbf{t}_W^{W'}$) could be known *a priori*, and the unknown part would be obtained by minimizing $\sum_k \|\mathbf{r}_k^{W'} - \mathbf{g}_k^{W'}\|^2$. The ATE ε_k at time k is then defined as the metric distance between the estimation and the ground truth of camera position: $\varepsilon_k = \|\mathbf{r}_k^{W'} - \mathbf{g}_k^{W'}\|$.



(a) (b)

Fig. 5. (a) Estimated robot trajectories by different methods. (b) Cost function values of each component in MFG-LBA.

*2) **Datasets**:* We have tested on two datasets: HRBB dataset and Bicocca dataset. The *HRBB dataset* is collected on the 4-th floor of H. R. Bright building at Texas A&M University using a PackBot. A Nikon 5100 camera with 60° HFOV is mounted on the PackBot. The dataset consists of 12000 raw frames with 1920×1080 resolution and 30 fps frame rate. In our experiments, we down-sample the images to 640×360 for faster computation. The robot trajectory covers around 70 meters. The ground truth of camera poses is obtained using artificial landmarks with error of ± 1 cm.

The *Bicocca dataset* used here is an image sequence from the publicly available Rawseeds datasets [24]. The image resolution is 320×240, and ground truth of camera positions is also provided. A sequence of 2000 frames are used for experiments, describing a trajectory of around 77 meters.

*3) **Results**:* Tabs. I(a) and I(b) show the ATE's for each method on two datasets. We can see that Point-LBA performs better than 1Point-EKF and this complies with the observation of [4]. Our algorithm MFG-LBA outperforms both of the other methods, achieving a relative mean ATE of 1.09% and 3.29% on each dataset, respectively. This implies that the mean ATE of our algorithm is 63.9% and 29.1% less than that of Point-LBA on each dataset, respectively. The larger ATE on the Bicocca dataset is due to lower image resolution.

Fig. 5(a) shows the estimated trajectories on the HRBB dataset. We can see that our algorithm suffers the smallest scale and angular drift. Fig. 5(b) illustrates the cost function values of each component of (8) over key frames, from which we see how each type of feature and constraint contributes to the LBA process.

B. Outdoor Test

We have collected an outdoor dataset containing 3240 frames with a hand-held camera (Nikon 5100). The camera trajectory covers around 150 meters on campus. The images are down-sampled to resolution 640×360. Although we do not have true camera trajectory, we have measured the plane normal directions of building facades on Google Maps™

$$\mathcal{C}_{\text{total}}(\mathcal{M}_k) = \sum_{\kappa=k-w+1}^{k} \left[\sum_{\mathbf{P}\in\mathcal{S}_{\text{pt}}^k} \mathcal{C}_{\text{pt}}(\mathbf{P},\kappa) + \sum_{\mathbf{L}\in\mathcal{S}_{\text{ln}}^k} \mathcal{C}_{\text{ln}}(\mathbf{L},\kappa) + \sum_{\mathbf{V}\in\mathcal{S}_{\text{vp}}^k} \mathcal{C}_{\text{vp}}(\mathbf{V},\kappa) \right] + \sum_{\mathbf{\Pi}\in\mathcal{S}_{\text{pl}}^k} \left[\sum_{\mathbf{P}\in\mathcal{S}_{\text{pt}}^k} \mathcal{C}_{\text{pl}}(\mathbf{P},\mathbf{\Pi}) + \sum_{\mathbf{L}\in\mathcal{S}_{\text{ln}}^k} \mathcal{C}_{\text{pl}}(\mathbf{L},\mathbf{\Pi}) \right] \quad (8)$$

TABLE I

(a) ATE of HRBB Dataset

Method	Mean ATE (m)	Std. of ATE (m)	Max ATE (m)	Mean ATE over trajectory length
1Point-EKF	4.37	2.01	8.25	6.24%
Point-LBA	2.11	1.06	3.52	3.01%
MFG-LBA	0.76	0.51	2.02	1.09%

(b) ATE of Bicocca Dataset

Method	Mean ATE (m)	Std. of ATE (m)	Max ATE (m)	Mean ATE over trajectory length
1Point-EKF	3.64	2.34	9.56	4.73%
Point-LBA	3.57	2.12	9.62	4.64%
MFG-LBA	2.53	1.69	8.77	3.29%

(c) Plane Normal Error (in $^\circ$)

Method	$\mathbf{\Pi}_1$	$\mathbf{\Pi}_2$	$\mathbf{\Pi}_3$
Point-LBA	1.10	1.71	4.38
MFG-LBA	1.09	1.56	3.70

and use them for evaluation. For the Point-LBA algorithm, 3D planes are detected from the resulting 3D points by finding coplanar points using RANSAC; the found planes are then re-estimated by optimization. Tab. I(c) shows the plane normal errors of reconstructed building facades. Our method produces smaller plane normal errors than Point-LBA, implying a smaller angular drift (see Fig. 1).

VII. Conclusions and Future Work

We presented a method utilizing heterogeneous visual features and their inner geometric constraints as the integrated high level landmarks to assist robot navigation. This was managed through a multilayer feature graph. Our method extended LBA framework by explicitly exploiting different features and their geometric relationships in an unsupervised manner. Physical experiments showed that our algorithm outperformed state of the art in localization and mapping accuracy. In the future, we will use MFG to facilitate loop closure detection and consider incorporating appearance information to enhance robustness.

Acknowledgment

We would like to acknowledge the insightful thoughts from Y. Xu, A. Perera, and S. Oh in Kitware. We also thank W. Li, M. Hielsberg, J. Lee, Z. Gui, M. Hirami, S. Mun, S. Jacob, and P. Peelen for their inputs.

References

[1] J. Civera, O. G. Grasa, A. J. Davison, and J. Montiel, "1-point RANSAC for extended Kalman filtering: Application to real-time structure from motion and visual odometry," *Journal of Field Robotics*, vol. 27, no. 5, pp. 609–631, 2010.

[2] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd, "Generic and real-time structure from motion using local bundle adjustment," *Image and Vision Computing*, vol. 27.

[3] A. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, oct. 2003, pp. 1403 –1410 vol.2.

[4] H. Strasdat, J. Montiel, and A. J. Davison, "Real-time monocular SLAM: Why filter?" in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 2657–2664.

[5] E. Royer, M. Lhuillier, M. Dhome, and J.-M. Lavest, "Monocular vision for mobile robot localization and autonomous navigation," *International Journal of Computer Vision*, vol. 74, no. 3, pp. 237–260, 2007.

[6] W. Li and D. Song, "Toward featureless visual navigation: Simultaneous localization and planar surface extraction using motion vectors in video streams," in *IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China*, May-June 2014.

[7] G. Sibley, C. Mei, I. Reid, and P. Newman, "Vast-scale outdoor navigation using adaptive relative bundle adjustment," *The International Journal of Robotics Research*, vol. 29, no. 8, pp. 958–980, 2010.

[8] J.-P. Tardif, Y. Pavlidis, and K. Daniilidis, "Monocular visual odometry in urban environments using an omnidirectional camera," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. IEEE, 2008, pp. 2531–2538.

[9] 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," *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 647–663, 2012.

[10] J.-S. Gutmann, M. Fukuchi, and M. Fujita, "3D perception and environment map generation for humanoid robot navigation," *The International Journal of Robotics Research*, vol. 27, no. 10, pp. 1117–1134, 2008.

[11] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*. IEEE, 2007, pp. 225–234.

[12] 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 (PGAI), AAAI Conference on Artificial Intelligence (AAAI)*, Atlanta, Georgia, USA, July 2010.

[13] A. Flint, C. Mei, I. Reid, and D. Murray, "Growing semantically meaningful models for visual SLAM," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 467–474.

[14] 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, 2012.

[15] H. Li, D. Song, Y. Lu, and J. Liu, "A two-view based multilayer feature graph for robot navigation," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. St. Paul, MN, USA: IEEE, May 2012, pp. 3580–3587.

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

[17] R. von Gioi, J. Jakubowicz, J.-M. 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, april 2010.

[18] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd, "Real time localization and 3D reconstruction," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1. IEEE, 2006, pp. 363–370.

[19] B. M. Haralick, C.-N. Lee, K. Ottenberg, and M. Nölle, "Review and analysis of solutions of the three point perspective pose estimation problem," *International Journal of Computer Vision*, vol. 13, no. 3, pp. 331–356, 1994.

[20] D. Nistér, "An efficient solution to the five-point relative pose problem," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 6, pp. 756–770, 2004.

[21] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge Univ Pr, 2003.

[22] Y. Xu, S. Oh, and A. Hoogs, "A minimum error vanishing point detection approach for uncalibrated monocular images of man-made environments," in *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2013, pp. 1376–1383.

[23] J. Civera, O. G. Grasa, A. J. Davison, and J. M. M. Montiel, "1-Point RANSAC Inverse Depth EKF Monocular SLAM Matlab Code." http://webdiis.unizar.es/~jcivera/code/1p-ransac-ekf-monoslam.html.

[24] G. F. M. M. D. G. S. Andrea Bonarini, Wolfram Burgard and J. D. Tardos, "Rawseeds: Robotics advancement through web-publishing of sensorial and elaborated extensive data sets," in *In proceedings of IROS'06 Workshop on Benchmarks in Robotics Research*, 2006.