SLAM with Wireframe Detections: Exploiting Geometrically Meaningful Features in Structured Environments

Amay Saxena
University of California, Berkeley
amaysaxena@berkeley.edu

Abstract

Standard landmark-based visual-SLAM approaches tend to utilize point features as the primary parameterization of the map. Additionally, an underlying conditional independence assumption between scene landmarks is present, largely because it is difficult to infer any nontrivial dependencies between separate landmarks in 3D space through classical methods from images alone. Recent advances in deep learning have led to the availability of powerful feature extractors that can infer "wireframe" representations of images, which encode structural relationships between the detected points and lines in the image. In this work, we propose a novel SLAM backend capable of consuming wireframe observations as the primary sensor measurements. Our proposed approach jointly deals with point features and line features, while also explicitly enforcing inter-feature constraints, which are rendered observable thanks to the rich wireframe measurement. We empirically evaluate the localization and mapping performance on synthetic data, and show that enforcing inter-feature constraints greatly improves scene reconstruction consistency. We also provide an empirical analysis of the relative usefulness of point features, line features, and incidence constraints between points and lines.

1. Introduction

Simultaneous Localization and Mapping (SLAM) is the name for a class of problems and algorithms that involve jointly estimating the state of a robotic platform along with a map of the environment of the robot, usually observed through means of an exteroceptive sensor such as a camera or laser scanner. When the robot or sensor platform is allowed to evolve in 3 dimensions, the 6DOF pose of the robot must be estimated, in addition to a 3D map of the environment. Optical cameras are low-cost and energy efficient sensors which also provide highly expressive environmental information. As such, vision aided localization and mapping techniques have recently prevailed (e.g. [19, 6, 7, 14, 12, 8, 17]).

The standard paradigm for vision aided localization and mapping is *landmark-based SLAM*, where the environment is modelled as a collection of unordered landmarks. Most current state-of-the-art vision aided SLAM systems focus on using *point features* as landmarks (e.g. [19, 6, 7, 12]). This is largely due to the fact that point features are easy to consistently and efficiently detect in images, as several mature keypoint feature extraction and description approaches exist in the literature, such as BRISK, ORB, SIFT, SURF, etc [11, 15, 9]. In such algorithms, the map is modelled as an undordered sparse point cloud, and post-processing is generally needed to produce a map that can be used by downstream tasks.

In urban environments, however, there are much more expressive features, such a lines and planes, that can be used. There exist approaches in the literature that utilize both line and point observations in the literature [21, 13, 20, 10]. Most such approaches share the fundamental assumption with landmark based SLAM - that landmark states are independent of each other given the robot state. This means that we only constrain landmark states through their image observations. This is generally because we cannot reliably detect structural relationships between landmarks observed in an image through classical techniques. Since classical feature extractors tend to work along small image patches, we cannot tell the difference between an observed incidence relationship (such as between two lines or between a point and a line) that is induced by occlusion and an incidence relationship that actually represents a geometric relationship between scene features.

However, recent advances in deep learning have led to powerful feature extractors that *can* infer structural relationships between features in the scene. In this work, we propose using one such representation, the *wireframe* to infer and enforce constraints between landmarks in the scene, thus explicitly enforcing dependencies directly between landmarks (instead of through image observations). Our work is motivated by the success of wireframe pars-

Figure 1. Example wireframe detections from [23]. Junctions in red represent C-junctions and those in blue represent T-junctions.

ing approaches such as [22, 23, 2]. These works use convolutional neural network to extract geometrically meaningful lines and junctions from an image in the form of a graph, as shown in figure 1. These techniques can also distinguish between junctions that arise due to occlusion and those that arise due to the actual incidence of two lines in 3D space. This makes the wireframe an extremely rich representation of the geometric structure in an image. In this work, we propose a novel SLAM backend capable of consuming wireframe observations as the primary exteroceptive sensor. Our proposed backend is capable of jointly estimating point features and line features over *inter-feature* constraints between points and lines inferred from the wireframe. In this way, we propose explicitly enforcing geometric relationships between features in structured environments where such relationships can be detected. We further provide experimental evaluation of the usefulness of point features, line features, and inter-feature constraints for localization and mapping.

2. Methods

In this section we lay out the proposed SLAM backend. First, we establish some notation. We denote $\|v\|_S^2$ to be the squared Mahalanobis norm of $v \in \mathbb{R}^n$, weighted by the positive semi-definite matrix $S \in \mathbb{S}^n$. When the arguments are 3D vectors, the multiplication symbol $\times$ represents the standard cross-product in 3D space. We denote by $(\cdot)^\wedge$ and $(\cdot)^\vee$ (called "hat" and "vee" operators) the canonical isomorphisms between $\mathbb{R}^3$ and $\mathfrak{so}(3)$ (the Lie algebra of the group $SO(3)$ of rotations). In particular, if $\omega = (\omega_1, \omega_2, \omega_3)^\top \in \mathbb{R}^3$, then

$$\omega^\wedge = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}$$

is the corresponding skew-symmetric *cross-product matrix*. For all $\mathbf{v} \in \mathbb{R}^3$, $\omega^\wedge \mathbf{v} = \omega \times \mathbf{v}$ holds. If $\widehat{\omega} \in \mathfrak{so}(3)$ is a skew symmetric matrix of the above form, then $\widehat{\omega}^\vee$ arranges the entries into a 3D vector, and is the inverse of $(\cdot)^\wedge$.

We also overload the hat and vee operators to denote the canonical isomorphisms between $\mathbb{R}^6$ and $\mathfrak{se}(3)$ (the Lie algebra of the group $SE(3)$ of rigid body transforms). Which interpretation is used is understood by the dimensionality of the argument.

2.1. Landmark-based SLAM

Landmark-based SLAM is a class of SLAM problems and algorithms where the map of the environment is stored as a collection of independent landmarks. The objective is then to jointly estimate the states of evolving robot at each time-step (localization) along with the states of every landmark in the scene (mapping).

Let the unknown pose of the robot at time t be $\mathbf{x}_t \in SE(3)$. Let $\mathcal{U}, \mathcal{L}, \mathcal{Z}$ be the spaces from which odometry measurements, landmark states, and landmark measurements respectively are drawn. We consider a SLAM set-up where we receive odometry measurements $\mathbf{u}_t \in \mathcal{U}$ and have knowledge of the motion model $g : SE(3) \times \mathcal{U} \to SE(3)$ of the robot. We assume the evolution of the robot is corrupted by some Gaussian noise $\mathbf{w}_t \sim \mathcal{N}(0, \Sigma_w)$, so that $\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t) \oplus \mathbf{w}_t$ where $\oplus$ is the increment operator on $SE(3)$ [18]. We assume the environment of the robot is composed of m *landmarks*, with states $\mathbf{L}_i \in \mathcal{L}$ for $i = 1, \cdots, m$. A landmark $\mathbf{L}_j$ can be measured from a robot state $\mathbf{x}_t$ to produce a measurement according to the known *measurement model* $h(\mathbf{x}_t, \mathbf{L}_j) \in \mathcal{Z}$. We assume the measurement is subject to some noise so that the true measurement $\mathbf{z}_{tj}$ is produced according to a process $\mathbf{z}_{tj} = h(\mathbf{x}_t, \mathbf{L}_j) + \mathbf{v}_{tj}$ where $\mathbf{v}_{tj} \in \mathcal{N}(0, \Sigma_v)$ is additive Gaussian noise. Let S be a set of index pairs such that $(t, i) \in S$ if and only if landmark i was measured from robot pose $\mathbf{x}_t$. Then the SLAM problem can be stated as finding the assignment to the unknown variables $X = \{\mathbf{x}_t\}_{t=0}^n$ and $L = \{\mathbf{L}_i\}_{i=1}^m$ that best explain the observations $\{\mathbf{u}_t\}_{t=0}^{n-1}$ and $\{z_{tj}\}_{(t,j)\in S}$. We state this estimation problem as the minimization over X, L of the following nonlinear least squares cost function

$$J(X, L) = \sum_{t=1}^{n} \|\mathbf{x}_t \ominus g(\mathbf{x}_{t-1}, \mathbf{u}_{t-1})\|_{\Sigma_w^{-1}}^2 + \sum_{(i,j)\in S} \|\mathbf{z}_{ij} - h(\mathbf{x}_i, \mathbf{L}_j)\|_{\Sigma_v^{-1}}^2 \quad (1)$$

which can also be seen as the Maximum A-Posteriori (MAP) estimate $\max_{X,L} \mathbb{P}(X, L \mid \{\mathbf{u}_t\}, \{\mathbf{z}_{ti}\})$. The above nonlinear least squares problem can be solved iteratively using gradient-based methods such as Gauss-Newton and Levenberg-Marquardt. Solvers that exploit the sparse structure of the problem (and hence, Jacobian matrices) can solve instances of the above problem with thousands of variables very efficiently.

From the above, we can see that in order to utilize a particular kind of landmark for SLAM, we need to define three

things: first: the space $\mathcal{L}$ in which the state of the landmark lives, second: the observation model h that predicts measurements of a given landmark from a given robot pose, and third: the space $\mathcal{Z}$ in which landmark measurements live. Since the minimization above is solved using gradient-based methods, we also need to compute the Jacobians of the measurement model h with respect to the robot pose and the landmark state. Note that in general, the robot pose and landmark state may live on some manifold other than Euclidean space (such as $SE(3)$ for robot poses) and so the Jacobians need to be taken with respect to tangent space elements of $SE(3)$ and $\mathcal{L}$ [5]. We will use the junctions and lines detected in the wireframe as our landmarks. In subsequent sections, we will define our measurement models for junctions and lines.

2.2. Wireframe Detection

Our work on localization and mapping using wireframes is motivated by the success of F-CLIP [2] and L-CNN [22]. F-CLIP is a fully convolutional neural network designed to directly predict the parameters of all salient wireframe lines in an input image in an end-to-end fashion that achieves state-of-the-art performance. L-CNN is a full end-to-end wireframe parsing network. Since the focus of this work is designing a SLAM back-end capable of consuming wireframe observations, we test our approach on synthetic data with ground-truth wireframe observations. However, our assumptions are based on the capabilities of the present wireframe parsing techniques.

The output of a wireframe detector is a graph (E, V), where each vertex $v_i \in V$ is a junction (i.e. the meeting point of one or more lines), and each edge $(i, j) \in E$ indicates that there is a line connecting junction v_i and v_j. Each junction is specified in terms of its location $v_i = (x_i, y_i)$ in image co-ordinates. Junctions observed in a wireframe can be one of two types:

1. *C-junctions:* These are "corner" junctions which represent real junctions in the scene where one or more structural lines actually meet.

2. *T-junctions:* These are "virtual" junctions that are produced in the image observation only due to occlusion, and do not represent any real feature of the scene.

For the purposes of this work, we will assume that we can classify detected junctions as either C or T junctions. A technique for jointly junction types in addition to the wireframe is exhibited in [23]. We are mainly interested in C-junction detections, since C junctions represent geometrically significant features in the scene; namely, they represent the points of intersection of two or more lines in the scene. This enforces a constraint between all the lines involved in the junction and the junction point itself, which can be exploited for more consistent reconstruction. The details of how these constraints are enforced are presented in section 2.4.

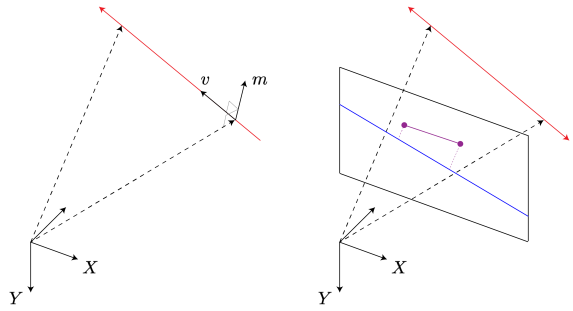

Figure 2. Summary of line measurement model. Left: the Plücker co-ordinates $(\mathbf{m}, \mathbf{v})$ of the 3D line in red. Right: an image measurement of the line. The blue line is the result of projecting the line onto the image plane, and the purple line is the detected line segment measurement. The error vector h_L comprises the distances from the two detected endpoints to the projected image (purple dotted lines).

2.3. Measurement Models

Since our main sensor is a camera and the wireframe measurements are made in image space, our landmark measurements will all be defined in image space. The measurement models then, should project the landmark state onto the image plane of a camera at the specified pose. We assume our cameras are calibrated and rectified, so that the pinhole model intrinsic matrix K is known and has standard parameters (f_x, f_y, c_x, c_y), which are the two focal lengths and co-ordinates of the camera plane center, respectively.

2.3.1 Junction Measurement Model

Each C-junction $\mathbf{z}_{ti} \in \mathbb{R}^2$ measured in the image taken from pose $\mathbf{x}_t$ is treated as the observation of a point landmark $\mathbf{p}_i \in \mathbb{R}^3$, expressed in the world co-ordinate frame. The measurement model that generates this observation is the standard pinhole perspective projection:

$$h(\mathbf{x}_t, \mathbf{p}_i) = \frac{1}{{}^C\mathbf{p}_i^z} K \, {}^C\mathbf{p}_i, \quad {}^C\mathbf{p}_i = \mathbf{R}_t^\top (\mathbf{p}_i - \mathbf{T}_t)$$

where $\mathbf{x}_t = (\mathbf{R}_t, \mathbf{T}_t) \in SE(3)$ is the pose of the robot relative to the world frame, ${}^C\mathbf{p}_i = ({}^C\mathbf{p}_i^x, {}^C\mathbf{p}_i^y, {}^C\mathbf{p}_i^z)$ are the coordinates of $\mathbf{p}_i$ in the camera frame, and K is the camera matrix

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Then, the term added to our nonlinear least squares cost function is $\|\mathbf{z}_{ti} - h(\mathbf{x}_t, \mathbf{p}_i)\|^2_{\Sigma_v^{-1}}$, where Σ_v is a measurement covariance which is left as a design parameter.

To compute the measurement Jacobians, we do so in two steps. First, by direct computation we can see that

$$\frac{\partial h}{\partial {}^C\mathbf{p}_i} = \begin{bmatrix} f_x/{}^C\mathbf{p}_i^z & 0 & -(f_x\ {}^C\mathbf{p}_i^x)/({}^C\mathbf{p}_i^z)^2 \\ 0 & f_y/{}^C\mathbf{p}_i^z & -(f_y\ {}^C\mathbf{p}_i^y)/({}^C\mathbf{p}_i^z)^2 \end{bmatrix}$$

and

$$\frac{\partial {}^C\mathbf{p}_i}{\partial \mathbf{p}_i} = \mathbf{R}_i^\top$$

To compute the derivatives with respect to the robot pose, we need to compute it with respect to small deviations $\xi^\wedge \in \mathfrak{se}(3)$. Here, $\xi \in \mathbb{R}^6$ and $(\cdot)^\wedge$ is the standard isomorphism $\mathbb{R}^6 \to \mathfrak{se}(3)$. To do this, we will instead compute the derivative with respect to the rotational deviation $\delta\omega \in \mathbb{R}^3$ of the rotational component and with respect to the translation vector $\mathbf{T}_t$ separately, and then compose these with the derivatives of each of those quantities with respect to the full pose. It is easy to see that

$$\frac{\partial {}^C\mathbf{p}_i}{\partial \mathbf{T}_t} = -\mathbf{R}_t^\top$$

To compute the derivative with respect to the rotation, we perturb $\mathbf{R}_t$ by a small deviation $\exp(\delta\omega)$ and examine the first-order coefficient in the Taylor expansion

$$\begin{aligned} &(\mathbf{R}_t \exp(\delta\omega))^\top (\mathbf{p}_i - \mathbf{T}_t) \\ &\quad \approx (I + \delta\omega^\wedge)^\top R_t^\top (\mathbf{p}_i - \mathbf{T}_t) \\ &\quad = \mathbf{R}_t^\top (\mathbf{p}_i - \mathbf{T}_t) - \delta\omega^\wedge \mathbf{R}_t^\top (\mathbf{p}_i - \mathbf{T}_t) \\ &\quad = \mathbf{R}_t^\top (\mathbf{p}_i - \mathbf{T}_t) + (\mathbf{R}_t^\top (\mathbf{p}_i - \mathbf{T}_t))^\wedge \delta\omega \end{aligned}$$

from which the Jacobian can be read off as

$$\frac{\partial {}^C\mathbf{p}_i}{\partial \delta\omega} = (\mathbf{R}_t^\top (\mathbf{p}_i - \mathbf{T}_t))^\wedge$$

and then finally the chain rule gives us the required Jacobians

$$\begin{aligned} \frac{\partial h}{\partial \mathbf{p}_i} &= \frac{\partial h}{\partial {}^C\mathbf{p}_i} \frac{\partial {}^C\mathbf{p}_i}{\partial \mathbf{p}_i}, \\ \frac{\partial h}{\partial \mathbf{T}_t} &= \frac{\partial h}{\partial {}^C\mathbf{p}_i} \frac{\partial {}^C\mathbf{p}_i}{\partial \mathbf{T}_t}, \\ \frac{\partial h}{\partial \delta\omega} &= \frac{\partial h}{\partial {}^C\mathbf{p}_i} \frac{\partial {}^C\mathbf{p}_i}{\partial \delta\omega} \end{aligned}$$

2.3.2 Line Measurement Model

A line ξ_j in a wireframe detection is measured as a pair of end-points $(\mathbf{s}_{ti}, \mathbf{e}_{ti}) \in \mathbb{R}^2 \times \mathbb{R}^2$ (start and end). We model each line in the scene as an infinite line. Lines can be truncated to the required segment during post-processing by using the observations of the line's endpoints. To parameterize an infinite line in 3D space, we use Plücker co-ordinates [1]. In particular, each line ξ_j is represented by a pair of vectors $(\mathbf{m}, \mathbf{v}) \in \mathbb{R}^3 \times \mathbb{R}^3 \simeq \mathbb{R}^6$, where $\mathbf{v}$ is a unit vector in the direction of the line and $\mathbf{m}$ is the normal to the plane passing through the origin containing the line, with magnitude equal to the distance between the line and the origin. More simply, if v is the unit direction vector of the line and $\mathbf{q}$ is any point on the line, then the co-ordinates are

$$\xi_j = \begin{bmatrix} \mathbf{m} \\ \mathbf{v} \end{bmatrix} = \begin{bmatrix} -\mathbf{v} \times \mathbf{q} \\ \mathbf{v} \end{bmatrix}$$

where $\mathbf{q}$ is the standard cross product in $\mathbb{R}^3$. Equivalently, we associate each (directed) line with a unit revolute twist $\in \mathfrak{se}(3)$ [16].

The Plücker co-ordinates are a 6 dimensional over parameterization for the 4 dimensional manifold of lines in 3D space, as can be seen by the fact that there are two regular constraints $\|\mathbf{v}\| = 1$ and $\mathbf{m}^\top \mathbf{v} = 0$. To deal with this overparameterization, a 4 dimensional *orthogonal representation* was proposed [1], wherein each ξ_j is expressed as a pair $(\mathbf{U}_j, \mathbf{W}_j) \in SO(3) \times SO(2)$, where

$$\begin{aligned} \mathbf{U} &= \begin{bmatrix} \frac{1}{\|\mathbf{m}\|}\mathbf{m}, & \mathbf{v}, & \frac{1}{\|\mathbf{m}\|}\mathbf{m} \times \mathbf{v} \end{bmatrix} \\ \mathbf{W} &= \frac{1}{\sqrt{1 + \|\mathbf{m}\|^2}} \begin{bmatrix} 1 & -\|\mathbf{m}\| \\ \|\mathbf{m}\| & 1 \end{bmatrix} \end{aligned}$$

This mapping is a diffeomorphism in a neighbourhood of any line that does not pass through the origin. The inverse can be computed as follows. Let $\mathbf{e}_1^n, \mathbf{e}_2^n, \mathbf{e}_3^n$ be the standard basis vectors for $\mathbb{R}^n$. Then

$$\begin{aligned} \mathbf{v} &= \mathbf{U}\mathbf{e}_2^3 \\ \mathbf{m} &= \frac{w_2}{w_1}\mathbf{U}\mathbf{e}_1^3 \\ (w_1, w_2)^\top &= \mathbf{W}\mathbf{e}_1^2 \end{aligned}$$

The orthogonal representation gives the set of lines a Lie group structure inherited from $SO(3) \times SO(2)$. This representation is very convenient for optimization, since the Jacobians of the measurement vectors can be computed with respect to tangent space deviations in $SO(3) \times SO(2)$ in exactly the same way as we would generally compute Jacobians with respect to rotations.

Plücker lines transform according to the Adjoint map of $SE(3)$

$$\xi_C = \mathrm{Ad}_{\mathbf{x}_t^{-1}} \xi_W = \begin{bmatrix} \mathbf{R}_t^\top & -\mathbf{R}_t^\top \mathbf{T}_t^\wedge \\ \mathbf{0} & \mathbf{R}_t^\top \end{bmatrix} \begin{bmatrix} \mathbf{m}_W \\ \mathbf{v}_W \end{bmatrix} \quad (2)$$

where $\xi_C = (\mathbf{m}_C, \mathbf{v}_C)$ and $\xi_W = (\mathbf{m}_W, \mathbf{v}_W)$ are the representations of the line ξ in the camera and world reference frames respectively, and $\mathbf{x}_t = (\mathbf{R}_t, \mathbf{T}_t)$ is the

camera pose relative to the world frame. The line ξ_C projects onto the image plane of the camera to a line $\{\mathbf{z}$ (homogeneous image coordinates) : $\mathbf{l}^\top \mathbf{z} = 0\}$ given by $\mathbf{l} = K\mathbf{m}_C$ where

$$K = \begin{bmatrix} f_y & 0 & 0 \\ 0 & f_x & 0 \\ -f_y c_x & -f_x c_y & f_x f_y \end{bmatrix}$$

is a camera intrinsic matrix for lines [19]. Finally, we define a cost vector $h_L(\mathbf{s}_{ti}, \mathbf{e}_{ti}, \mathbf{x}_t, \xi_i)$ as the two distances between the projected line and the observed end-points. Our measurement dictates that both these distances should be zero.

$$h_L(\mathbf{s}_{ti}, \mathbf{e}_{ti}, \mathbf{x}_t, \xi_i) = \frac{1}{\sqrt{l_1^2 + l_2^2}} \begin{bmatrix} \mathbf{l}^\top \mathbf{s}_{ti} \\ \mathbf{l}^\top \mathbf{e}_{ti} \end{bmatrix}$$
$$\mathbf{l} = [\mathrm{K}\ \mathbf{0}_{3\times 3}]\ \mathrm{Ad}_{\mathbf{x}_t^{-1}}\, \xi_i$$

Then, the term added to our nonlinear least squares cost function is $\|h_L(\mathbf{s}_{ti}, \mathbf{e}_{ti}, \mathbf{x}_t, \xi_i)\|^2_{\Sigma_L^{-1}}$, where Σ_L is a measurement covariance which is left as a design parameter.

To compute the Jacobians of h_L with respect to the pose $\mathbf{x}_t$ and line ξ_i, we start by computing the Jacobians of the Plücker line representation with respect to small deviations $\delta u = (\delta\theta, \delta w) \in \mathbb{R}^3 \times \mathbb{R}$ in the tangent space of $SO(3) \times SO(2)$. Using $\mathbf{v} = \mathbf{U}\mathbf{e}_2^3$, we write down

$$\begin{aligned} \mathbf{U}\exp(\delta\theta)\mathbf{e}_2^3 &\approx \mathbf{U}(I + \delta\theta^\wedge)\mathbf{e}_2^3 \\ &= \mathbf{U}\mathbf{e}_2^3 + \mathbf{U}\delta\theta^\wedge\mathbf{e}_2^3 \\ &= \mathbf{U}\mathbf{e}_2^3 - \mathbf{U}(\mathbf{e}_2^3)^\wedge\delta\theta \end{aligned}$$

from which the Jacobian can be read off as

$$\frac{\partial \mathbf{v}}{\partial \delta\theta} = -\mathbf{U}(\mathbf{e}_2^3)^\wedge \implies \frac{\partial \mathbf{v}}{\partial \delta u} = \begin{bmatrix} -\mathbf{U}(\mathbf{e}_2^3)^\wedge & \mathbf{0}_{3\times 1} \end{bmatrix}$$

To compute the Jacobians of $\mathbf{m}$, we need to consider the vectors $\mathbf{w} = (w_1, w_2)^\top = \mathbf{W}\mathbf{e}_1^2$ and $\widehat{\mathbf{m}} = \mathbf{U}\mathbf{e}_1^3$ so that $\mathbf{m} = (w_2/w_1)\widehat{\mathbf{m}}$. Then we can compute:

$$\frac{\partial \mathbf{m}}{\partial \widehat{\mathbf{m}}} = (w_2/w_1)\mathbf{I}_{3\times 3} \qquad \frac{\partial \widehat{\mathbf{m}}}{\partial \delta\theta} = -\mathbf{U}(\mathbf{e}_1^3)^\wedge$$
$$\frac{\partial \mathbf{m}}{\partial \mathbf{w}} = \begin{bmatrix} (1/w_1)\widehat{\mathbf{m}} & -(w_1/w_2^2)\widehat{\mathbf{m}} \end{bmatrix} \qquad \frac{\partial \mathbf{w}}{\partial \delta w} = \mathbf{W}\mathbf{e}_2^2$$

and since $\widehat{\mathbf{m}}$ and $\mathbf{w}$ are computed from independent components of $(\mathbf{U}, \mathbf{W})$, we have

$$\frac{\partial \mathbf{m}}{\partial \delta\theta} = \frac{\partial \mathbf{m}}{\partial \widehat{\mathbf{m}}}\frac{\partial \widehat{\mathbf{m}}}{\partial \delta\theta}$$
$$\frac{\partial \mathbf{m}}{\partial \delta w} = \frac{\partial \mathbf{m}}{\partial \mathbf{w}}\frac{\partial \mathbf{w}}{\partial \delta w}$$

which are the Jacobians that form the column blocks of the matrix $\partial\mathbf{m}/\partial\delta u$. Stacking this on top of $\partial\mathbf{v}/\partial\delta u$ gives us the Jacobian $\partial\xi/\partial\delta u$ of the Plücker line ξ_i with respect to a tangent deviation of the orthogonal representation. Finally, the Jacobians of the error vector h_L with respect to the Plücker line ξ_i can be computed straightforwardly. Let the measured endpoint coordinates be $\mathbf{s}_{ti} = (a_1, a_2)$ and $\mathbf{e}_{ti} = (b_1, b_2)$.

$$\frac{\partial h_L}{\partial \mathbf{l}} = \begin{bmatrix} -\frac{-a_1 {l_2}^2 + a_2 l_1 l_2 + l_1 l_3}{{l_1}^2 + {l_2}^2} & -\frac{-a_2 {l_1}^2 + a_1 l_2 l_1 + l_2 l_3}{{l_1}^2 + {l_2}^2} & 1 \\ -\frac{-b_1 {l_2}^2 + b_2 l_1 l_2 + l_1 l_3}{{l_1}^2 + {l_2}^2} & -\frac{-b_2 {l_1}^2 + b_1 l_2 l_1 + l_2 l_3}{{l_1}^2 + {l_2}^2} & 1 \end{bmatrix}$$
$$\frac{\partial \mathbf{l}}{\partial \xi_i} = \begin{bmatrix} K & \mathbf{0}_{3\times 3} \end{bmatrix} \mathrm{Ad}_{\mathbf{x}_t^{-1}}$$

Applying similar techniques to equation 2, we can write down the Jacobians with respect to the pose $(\mathbf{R}_t, \mathbf{T}_t)$, with respect to the deviations $(\delta\omega, \delta\mathbf{T})$ in the rotation and translation respectively.

$$\frac{\partial \mathbf{l}}{\partial \delta\omega} = \begin{bmatrix} K & \mathbf{0}_{3\times 3} \end{bmatrix} \begin{bmatrix} \mathbf{m}_C^\wedge \\ (\mathbf{R}_t^\top \mathbf{v}_W)^\wedge \end{bmatrix}$$
$$\frac{\partial \mathbf{l}}{\partial \mathbf{T}_t} = \begin{bmatrix} K & \mathbf{0}_{3\times 3} \end{bmatrix} \begin{bmatrix} \mathbf{R}_t^\top \mathbf{v}_W^\wedge \\ \mathbf{0}_{3\times 3} \end{bmatrix}$$

2.4. Inter-feature Constraints

Each C-junction that is detected in the image wireframe gives us geometric incidence information about the junction point and lines involved in that junction. In particular, if a detected junction i is a C-junction and the lines incident on it are lines $\{\xi_j\}$, then this measurement tells us that lines ξ_j all meet at the point $\mathbf{p}_i$ in 3D space. We propose adding a cost term for each such measurement that enforces this incidence relationship.

In particular, we propose processing each measured C-junction as follows. First, the junction is processed as a point landmark measurement $\mathbf{p}_i$ and a cost term is introduced according to the measurement model described above (c.f. sub-section 2.3.1). Then, each line ξ_j involved in the junction is processed as a line measurement and a cost term is introduced according to the line measurement model described above (c.f. sub-section 2.3.2). Finally, for each line ξ_j, another cost term is introduced which enforces that the distance between point $\mathbf{p}_i$ and line ξ_j must be zero. Note that such a cost term should only be introduced if such a term between the same point-line pair has not already been introduced by a previous measurement.

The distance between the point $\mathbf{p}_i$ and line $\xi_j = (\mathbf{m}_j, \mathbf{v}_j)$ is given by

$$d(\mathbf{p}_i, \xi_j) = \|\mathbf{v}_j \times \mathbf{p}_i + \mathbf{m}_j\|_2$$

So we propose defining the following cost vector h_{LP}:

$$h_{LP}(\mathbf{p}_i, \xi_j) = \mathbf{v}_j \times \mathbf{p}_i + \mathbf{m}_j$$

and adding a term of the form $\|h_{LP}(\mathbf{p}_i, \xi_j)\|^2_{\Sigma_{LP}^{-1}}$ to the cost function, where Σ_{LP} is a measurement covariance which is left as a design parameter. The required Jacobians are readily computed as

$$\frac{\partial h_{LP}}{\xi_j} = \begin{bmatrix} \mathbf{I}_{3\times 3} & -\mathbf{p}^\wedge \end{bmatrix} \qquad \frac{\partial h_{LP}}{\mathbf{p}_i} = \mathbf{v}_j^\wedge$$

Note that the above cost term directly constrains landmarks in 3D space; there is no camera pose involved in the cost term at all. This is a departure from the standard landmark-based SLAM paradigm where landmarks are assumed independent of each other given the camera pose, and so landmarks are only constrained to each other through image observations. The reason we were able to do this is that the wireframe measurements give us geometric information about the spatial dependencies between landmarks.

2.5. Point-Line SLAM with Inter-Feature Constraints

We are now ready to state the full nonlinear cost-function we will use, involving point landmarks, line landmarks, and inter-feature constraints between point and line landmarks. Let S_P be a set of index pairs such that $(i,t) \in S_P$ if and only if junction $\mathbf{p}_i$ was measured from camera pose x_t, and let S_L be a similar set for line landmark measurements. Let S_{LP} be a set of index pairs (i,j) such that junction $\mathbf{p}_i$ and line ξ_j were measured as being co-incident as part of a C-junction. Then our cost function reads

$$\begin{aligned} J(X,L) = &\sum_{t=1}^{n} \|\mathbf{x}_t \ominus g(\mathbf{x}_{t-1}, \mathbf{u}_{t-1})\|^2_{\Sigma_w^{-1}} \\ &+ \sum_{(i,t)\in S_P} \|\mathbf{z}_{it} - h(\mathbf{x}_t, \mathbf{L}_i)\|^2_{\Sigma_v^{-1}} \\ &+ \sum_{(i,t)\in S_L} \|h_L(\mathbf{s}_{ti}, \mathbf{e}_{ti}, \mathbf{x}_t, \xi_i)\|^2_{\Sigma_L^{-1}} \\ &+ \sum_{(i,j)\in S_{LP}} \|h_{LP}(\mathbf{p}_i, \xi_j)\|^2_{\Sigma_{LP}^{-1}} \end{aligned} \tag{3}$$

3. Experimental Results

We implement each of the cost functions (and their Jacobians with respect to landmarks and poses) described above and incorporate them into an incremental graph-SLAM algorithm. We use GTSAM [4, 3] as our optimization backend. The cost function is minimized using a Levenberg-Marquardt optimizer. The approach is validated in a simulated SceneCity-rendered dataset [23]. The scene is a city block with a number of buildings, and the camera takes a round of the whole block (figure 3). Odometry is taken to be noisy relative pose measurements. Landmarks are initialized when they are detected using a groundtruth depth image along with the current estimate of the camera pose (note that this means neither the ground truth pose nor the ground truth landmark state is used for initialization). However, ground-truth landmark correspondences between subsequent images are assumed known. The main focus of this work is the SLAM backend; designing a reliable front-end for landmark initialization and correspondence detection is the topic of ongoing follow-up work.

See figure 4 for localization accuracy on the test image sequence. We have plotted the localization error in translation and orientation estimates with respect to the ground-truth poses. We plot the performance of four separate estimators. First, as a baseline, the result of simply integrating the odometry measurements is plotted. Then, we plot the performance of four different SLAM backends. The first is a backend that utilizes only point landmarks (using just the C-junction detections). This has the structure of a usual visual SLAM backend that uses keypoint features. The second is a backend that utilizes only line landmarks, third a backend that uses both point and line landmarks (without any inter-feature constraints, so that each landmark is assumed independent given the robot pose) and finally the proposed backend that jointly utilizes point landmarks, line landmarks, and inter-feature constraints between point and line landmarks. We hencforth abbreviate these backends to P, L, PL, and PLC respectively.

We can see that the localization accuracy of the P, PL, and PLC backends are comparable, and all perform very well. The performance of the L backend is noticeably worse than the others. We attribute this to the fact that line measurements can in fact have very high uncertainty when they are measured as a very small line segment in the image. This happens in one of two ways. Either (1) the line segment in the environment is itself small, or (2) the line segment is being viewed almost along the direction of the line, so that even a long line segment is observed as a very short one. Recall that our cost function associated with line measurements assembles the distance between the projected line and the two endpoints in image space. Further recall that every line segment (regardless of observed length) is modelled as an infinite line in 3D space. When the line is observed to be really small, these endpoints are very close to each other, and so even very incorrect estimates of the line can result in small error values. This results in line measurements without point measurements (whose observations are more uniformly stable) providing more uncertain pose estimations, likely due to landmarks being initialized into the attraction basin of a different local minimum than the global minimum. This problem is fixed by the inter feature constraints: by constraining the line measurements to (more stable) point measurements, we can alleviate the local minimum issues.

A similar trend can be seen in the *mapping* or scene reconstruction performance of the various backends. Since

Figure 3. Example images with overlayed wireframe detections from the SceneCity dataset used for experimental validation (c.f. section 3).

we care about reconstructing the whole wireframe, we only consider the mapping performance of the PL and PLC backends. This also lets us effectively isolate the main strength of inter-feature constraints - consistent scene reconstruction. See figure 5 for a visualization of the reconstructed wireframes outputted by the PL and PLC backends. Note that the wireframe outputs were post processed to truncate the reconstructed lines according to the predicted location of the junctions lying on them. During estimation, all lines are treated as infinite lines.

In the PL reconstruction, each point and line are treated as independent landmarks and are constrained only through their image observations. On the other hand, the PLC backend enforces detected geometric constraints between the landmarks through the procedure described previously. It is clear that the latter approach leads to a much more consistent reconstruction. From the PL reconstruction, we can visually see the local minimum problem described earlier. There are many artefacts of small lines that are estimated to be at a location very far away from the structures they actually belong to. By comparison, longer line segments prove much easier to estimate correctly. This can be attributed to the instability of small line-segment measurements leading to the corresponding lines converging to an incorrect local minimum. On the other hand, these errors are completely fixed by the PLC backend. No more artefacts of small lines are present, and reconstructions of longer lines are much more consistent.

4. Discussion

It is clear from the localization and mapping performance that enforcing inter-feature constraints leads to more consistent reconstruction in environments where such information is available. In general, this suggests that there is merit in exploiting knowledge of geometric relationships between features of a structured environment for localization and mapping. These relationships are made observable through the use of deep neural network based feature extractors which can learn to detect such relationships from images alone. In the absence of learning-based extractors, it is not possible to efficiently detect relationships between geometric primitives in the scene: since classical feature extractors work along local image patches, they cannot distinguish between relationships induced by occlusion or other artefacts, and relationships that actually represent structure in the scene. In this work, we have shown that the wireframe representation can allow us to produce richer and more consistent maps of the environment, which can then be used for downstream tasks such as navigation or modelling.

It should also be noted that due to the sparse nature of the wireframe representation, we are able to perform mapping and localization with a very small number of points and lines. This suggests the importance of exploiting geometrically meaningful features and constraints. Future work should address combining the present work with additional geometric features such as planes and vanishing points.

References

[1] A. Bartoli and P. Sturm. Structure-from-motion using lines: Representation, triangulation, and bundle adjustment. *Computer vision and image understanding*, 100(3):416–441, 2005.

[2] X. Dai, X. Yuan, H. Gong, and Y. Ma. Fully convolutional line parsing, 2021.

[3] F. Dellaert et al. Gtsam. *URL: https://borg. cc. gatech. edu*, 2012.

[4] F. Dellaert, M. Kaess, et al. Factor graphs for robot perception. *Foundations and Trends® in Robotics*, 6(1-2):1–139, 2017.

[5] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard. A Tutorial on Graph-Based SLAM. *IEEE Intelligent Transportation Systems Magazine*, 2(4):31–43, 2010.

[6] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis. Consistency analysis and improvement of vision-aided inertial navigation. *IEEE Transactions on Robotics*, 30(1):158–176, 2013.

[7] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis. Camera-imu-based localization: Observability analysis and consistency improvement. *The International Journal of Robotics Research*, 33(1):182–201, 2014.

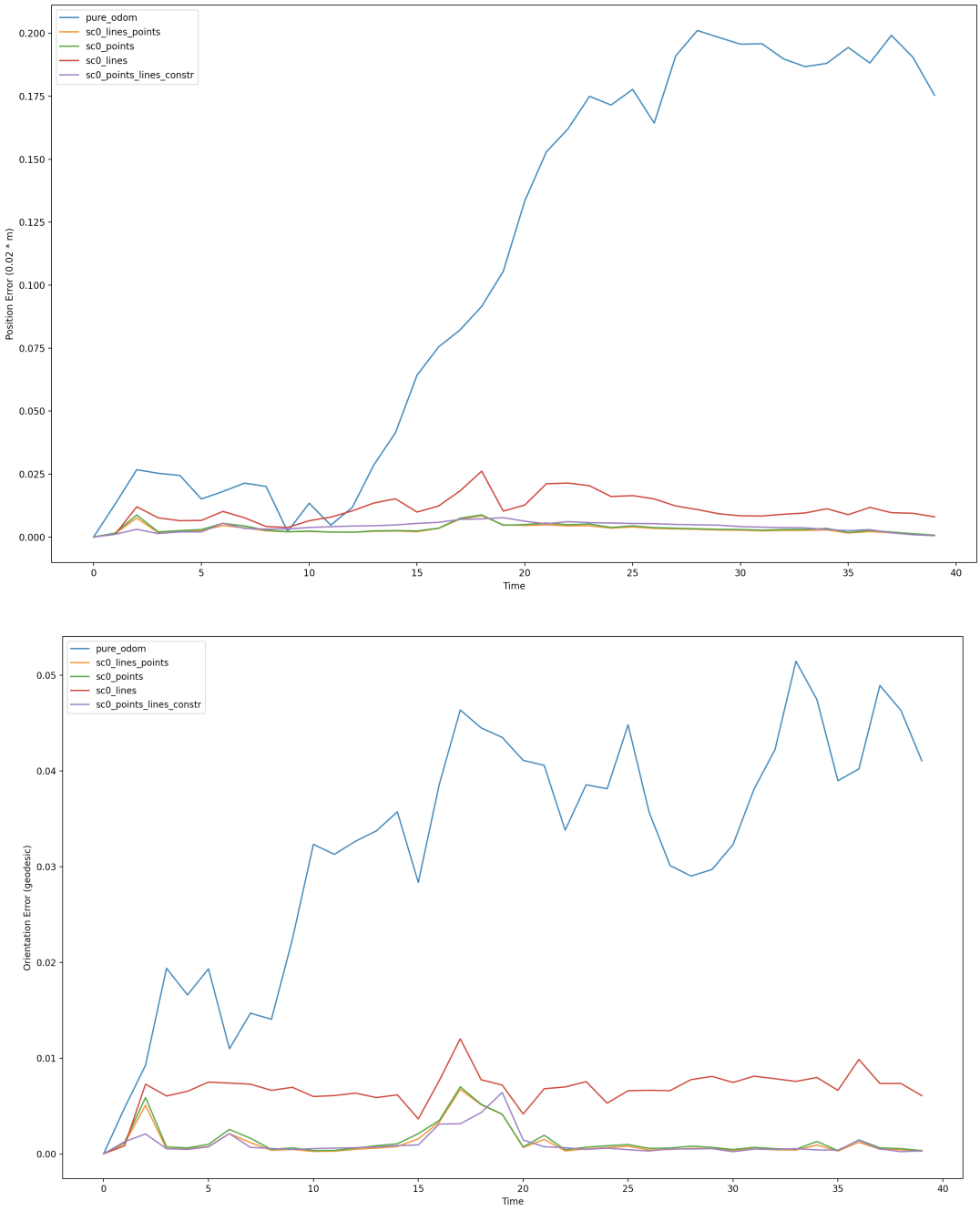

Figure 4. Localization performance for odometry integration and the four SLAM backends P, L, PL, and PLC. Top: Plot of position error against time, where the error metric is the Euclidean distance between the estimated position and ground truth position of the camera at each time, at 1/50th scale. Bottom: Plot of position error against time, where the error metric is the geodesic distance between the estimated and ground truth camera orientations in unit quaternion space.

[8] G. Huang, M. Kaess, and J. J. Leonard. Towards consistent visual-inertial navigation. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4926–4933. IEEE, 2014.

[9] E. Karami, S. Prasad, and M. Shehata. Image matching using sift, surf, brief and orb: performance comparison for distorted images. *arXiv preprint arXiv:1710.02726*, 2017.

[10] D. G. Kottas and S. I. Roumeliotis. Efficient and consistent vision-aided inertial navigation using line observations. In *2013 IEEE International Conference on Robotics and Au-*

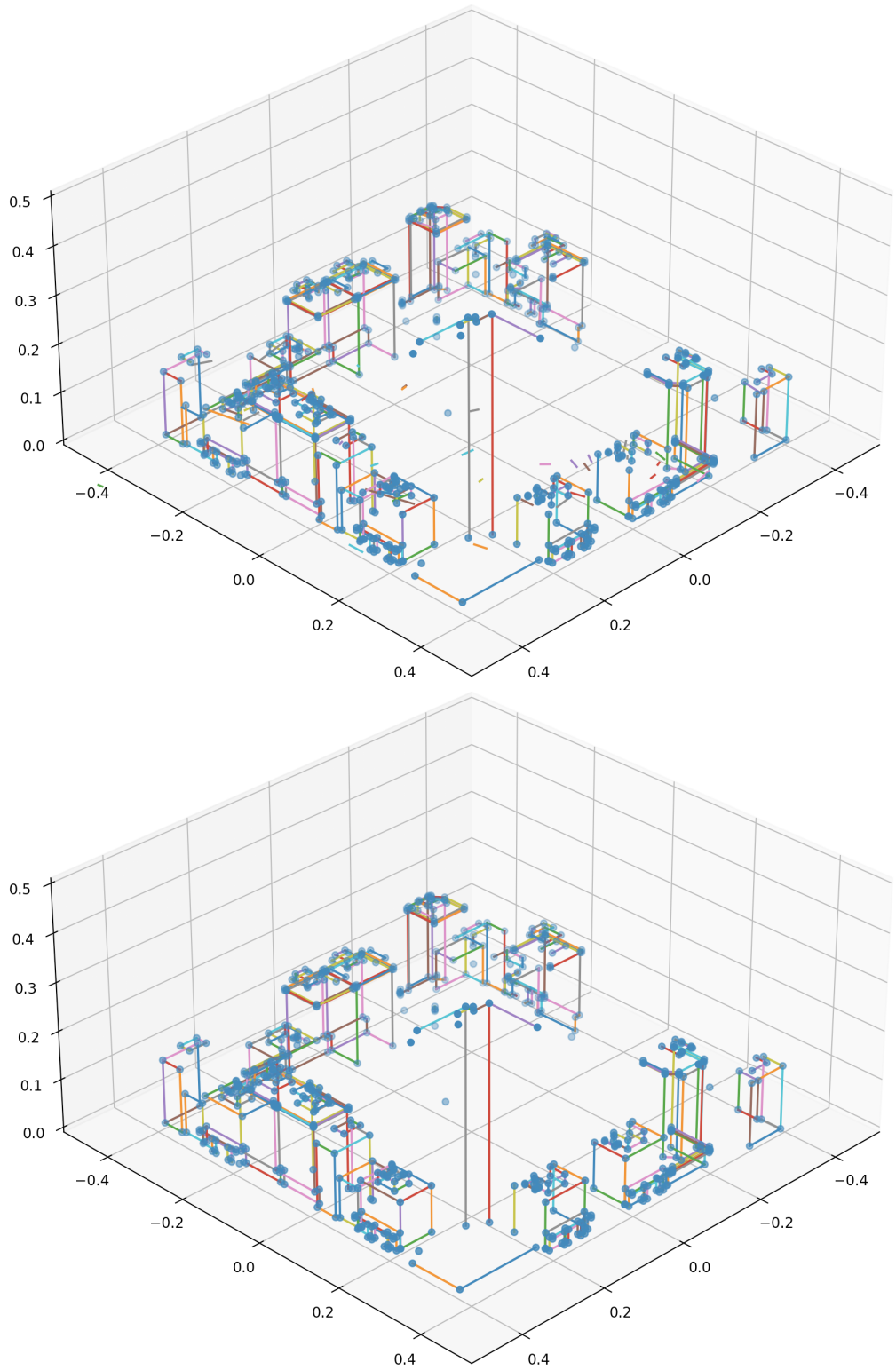

Figure 5. The reconstructed wireframe of the whole map from two different backends. Top: PL, Bottom: PLC. It is clear that the introduction of inter-feature constraints leads to a much more consistent reconstruction. Note that not all features in the ground truth wireframe are reconstructed since not all such features were visible from multiple camera poses.

tomation, pages 1540–1547. IEEE, 2013.

[11] S. Leutenegger, M. Chli, and R. Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *2011 International conference on computer vision*, pages 2548–2555. Ieee, 2011.

[12] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale. Keyframe-based visual–inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, 34(3):314–334, 2015.

[13] H. Li, J. Yao, J.-C. Bazin, X. Lu, Y. Xing, and K. Liu. A monocular slam system leveraging structural regularity in manhattan world. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2518–2525. IEEE, 2018.

[14] M. Li and A. I. Mourikis. High-precision, consistent ekf-based visual-inertial odometry. *The International Journal of Robotics Research*, 32(6):690–711, 2013.

[15] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.

[16] R. M. Murray, S. S. Sastry, and L. Zexiang. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Inc., USA, 1st edition, 1994.

[17] T. Qin, P. Li, and S. Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018.

[18] J. Sola, J. Deray, and D. Atchuthan. A micro lie theory for state estimation in robotics. *arXiv preprint arXiv:1812.01537*, 2018.

[19] Y. Yang and G. Huang. Observability analysis of aided ins with heterogeneous features of points, lines, and planes. *IEEE Transactions on Robotics*, 35(6):1399–1418, 2019.

[20] H. Yu and A. I. Mourikis. Vision-aided inertial navigation with line features and a rolling-shutter camera. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 892–899. IEEE, 2015.

[21] F. Zheng, G. Tsai, Z. Zhang, S. Liu, C.-C. Chu, and H. Hu. Trifo-vio: Robust and efficient stereo visual inertial odometry using points and lines. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3686–3693. IEEE, 2018.

[22] Y. Zhou, H. Qi, and Y. Ma. End-to-end wireframe parsing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 962–971, 2019.

[23] Y. Zhou, H. Qi, Y. Zhai, Q. Sun, Z. Chen, L.-Y. Wei, and Y. Ma. Learning to reconstruct 3d manhattan wireframes from a single image, 2021.