2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

# A Point-Line Features Fusion Method for Fast and Robust Monocular Visual-Inertial Initialization

Guoqiang Xie[1], Jie Chen[1], Tianhang Tang[1], Zeyu Chen[1], Ling Lei[1] and Yiguang Liu[1*]

***Abstract*— Fast and robust initialization is essential for highly accurate monocular visual-inertial odometer (VIO), but at present majority of initialization methods rely only on point features, unstable in low texture and blurring situations. Therefore, we propose a novel point-line features fusion method for monocular visual-inertial initialization, as line features are more stable and provide richer geometric information than point features: 1) a closed-form line features initialization method is presented, and combined with point features to obtain a more integrated and robust linear system; 2) a monocular depth network is adopted to provide learned affine-invariant depth map, requiring only one prior depth map for the first frame, which can improve performance under low-parallax scenarios; 3) we can easily use RANSAC to reject outliers in solving linear system based on our formulation. Moreover, line feature re-projection residual is added to visual-inertial bundle adjustment (VI-BA) to obtain more accurate initial parameters. The proposed method is more accurate and robust than state-of-the-art methods due to the line features, especially under extreme low-parallax scenarios, and extensive experiments on popular datasets have confirmed, 0.5s initialization window on EuRoC MAV, 0.3s initialization window on TUM-VI, while the standard method normally waits for a window of 2s.**

## I. INTRODUCTION

Visual-inertial odometer utilizes images and IMU measurements to estimate camera motion in an unknown environment [1]. Simple integration of high-frequency IMU measurements is often corrupted by noise and bias, leading to drifts that make long-time estimation unreliable. The camera can constrain the integration results of the IMU by providing rich feature information about the environment, IMU compensates for the camera's instability at high speeds of motion, making them a naturally complementary pair of sensors. Lightweight, compact, and low-cost cameras and IMUs can be easily and quickly equipped on various devices, enabling the extensive use of VIO in emerging technologies such as AR/VR and autonomous driving [2], [3].

Optimization-based [4], [5] and filter-based [6], [7] are the two mainstream approaches for the VIO estimation. Importantly, a fast and robust initialization algorithm is indispensable for both approaches. The initialization task is to use the measurements to estimate the observable parameters (e.g. gravity, velocity, biases) as quickly and accurately as possible. Bad initialization may result in VIO non-convergence, adversely affecting the performance of localization accuracy. Many initialization methods assume that the initial motion is stationary, then calculate the parameters with IMU measurements [8], [9]. However, although these methods are simple, it is only effective when the system is in a strictly static state. In dynamic scenarios, the general approach is to build a linear system from measurements to solve for the initial parameters. However, it may fail, especially under low-parallax and low-excitation situations.

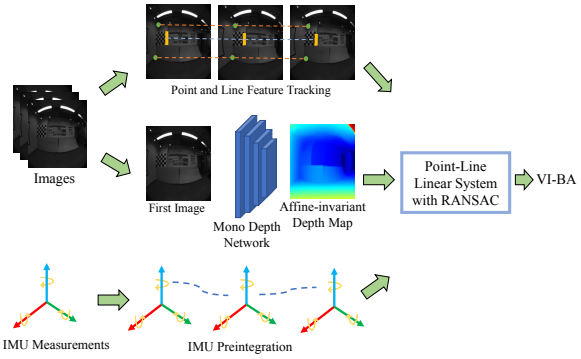


Fig. 1. Overview of the proposed point-line features fusion monocular visual-inertial initialization method. Our method performs point feature and line feature tracking and utilizes a monocular depth network to generate an affine-invariant depth map for the first image, which is then combined with IMU preintegration to build a robust linear system. Outliers are rejected during the solution process using RANSAC, and the results are fed to VI-BA with line feature re-projection residual for further refinement.

To address the initialization problem under low-excitation scenes, a recent method [10] utilizes learned monocular depth to provide additional constraints for point features, incorporating these into VI-BA to aid in monocular inertial initialization. This method requires computing a depth map for each keyframe. Another method [11] reduces the number of parameters by utilizing constraints between frames and employs the RANSAC to reject outliers. However, the performance of the method is prone to failure in cases where point features appear discontinuously (e.g. motion blur). In our work, as shown in Fig. 1, we first perform point and line feature tracking, and then combine the affine-invariant depth map of the first image output from the monocular depth network and the IMU preintegration to build a more robust point-line linear system with RANSAC. In the nonlinear optimization process, the line feature re-projection residual is added to the VI-BA to obtain more accurate results. Our main contributions are:

- We propose a novel closed-form method of line features, which can be conveniently integrated with the linear system based on point features.
- We build an integrated linear system with RANSAC of point-line features for monocular visual-inertial initialization, where the first frame of affine-invariant depth

[1]The authors are with the College of Computer Science, Sichuan University, Chengdu 610065, China. (Yiguang Liu is the corresponding author.)

map is used as a priori information.
- Experiments on the EuRoC MAV and TUM-VI datasets report performance under 0.5s and 0.3s windows with 5 keyframes, demonstrating fast and robust initialization capabilities.

## II. RELATED WORK

There has been considerable work on VIO initialization, generally divided into two different solutions: loosely-coupled methods [4], [12], [13], and tightly-coupled methods [14], [15], [16].

### A. Loosely-coupled Methods

The loosely-coupled methods obtain the up-to-scale trajectory by solving the Structure from Motion (SfM) problem using visual measurements. The relative motion is then obtained by integrating the IMU measurements, and the initial parameters can be solved by aligning the trajectories. This method is based on the accurate solving of the SfM problem. It was first proposed by Mur-Artal and Tardós [12] where the initial parameters are processed stepwise, and the initialization problem is divided into several sub-problems. However, this method requires 10-15 seconds for convergence, which is unacceptable in certain urgent situations. Qin et al. [4], [17] proposed a similar approach that ignores the accelerometer bias and greatly reduces the convergence time. The vast majority of advanced VIOs that rely on point-line features, such as EPLF-VINS [18] and PL-VINS [13], also use this point-only initialization method. The fatal drawback of the loosely-coupled methods is their reliance on a very significant parallax, which is challenging to achieve on an extremely short window.

### B. Tightly-coupled Methods

The tightly-coupled approaches build a linear system by combining visual and IMU measurements. This approach avoids solving the SfM problem and the inconsistencies caused by the step-by-step solution. Martinelli et al. [14] proposed an impressive method to estimate the initial parameters by combining point features and measurements from the accelerometer. This method assumes that the gyroscope is unbiased, which is clearly an unreasonable assumption. Liu et al. [15] proposed an initialization method that combines point and line features. An integrated linear system is first built from point and line features, and then the parameters are further refined using nonlinear optimization.

Benefiting from research on monocular depth estimation based on deep learning, Zhou et al. [10] proposed using learned measurements as high-level outputs to improve performance under low-parallax scenes. This method requires computing a depth map for each keyframe, which is computationally intensive. Merrill and Geneva et al. [11] reduced computational overhead by transferring the initial parameters to the inertial coordinate of the first frame, requiring the computation of the depth map for only the first image. It is worth noting that the vast majority of initialization methods rely solely on point features. However, in the case of poor lighting or drastic changes in viewing angle, point features may be unstable. Based on the method [11], we added line features to build a linear system combining point and line features, and added the line feature re-projection residual to the nonlinear optimization process to obtain more accurate and robust results.

## III. METHODOLOGY

According to [19], in the absence of any the prior information, the parameters desired to be recovered include:

$$\mathbf{x} = [{}^{I_0}\mathbf{p}_{f_1}^\top \quad \cdots \quad {}^{I_0}\mathbf{p}_{f_M}^\top \quad {}^{I_0}\mathbf{v}_{I_0}^\top \quad {}^{I_0}\mathbf{g}^\top]^\top \tag{1}$$

where $\{I_0\}$ represents the first IMU frame, ${}^{I_0}\mathbf{p}_{f_i}$ is the 3DoF feature point position with respect to $\{I_0\}$, ${}^{I_0}\mathbf{v}_{I_0}$ and ${}^{I_0}\mathbf{g}$ are the velocity of the system and local gravity in the $\{I_0\}$ frame, respectively. However, when given an affine-invariant depth map, $D$, in the frame $\{I_0\}$, we can reduce the number of initial parameters. We assume that the metric depth scalar $z_i$ of a feature point ${}^{C_0}\mathbf{p}_{f_i}$ can be expressed as $z_i = ad_i + b$, where $d_i = D(u_{i,0}, v_{i,0})$, $a$ and $b$ are the scale and shift parameters, and are constant for the whole depth map. The transformation between the camera and IMU is known. In this way, we can reduce the parameters to be estimated as:

$$\mathbf{x}' = [a \quad b \quad {}^{I_0}\mathbf{v}_{I_0}^\top \quad {}^{I_0}\mathbf{g}^\top]^\top \tag{2}$$

This is an $8\times1$ vector where all point features depth values in the frame $\{I_0\}$ are expressed as a linear function of the depth map. Using learned depth priors as a high-level input can improve performance under low-parallax scenarios. Next, we will introduce the inertial model and then detail the linear system built from point features and line features.

### A. Inertial Model

A typical 6-axis inertial measurement unit (IMU) can provide local angular velocity ${}^I\boldsymbol{\omega}_m$ and local acceleration ${}^I\mathbf{a}_m$ at high frequencies:

$${}^I\boldsymbol{\omega}_m(t) = {}^I\boldsymbol{\omega}(t) + \mathbf{b}_g(t) + \mathbf{n}_g(t) \tag{3}$$

$${}^I\mathbf{a}_m(t) = {}^I\mathbf{a}(t) + {}^I_G\mathbf{R}(t){}^G\mathbf{g} + \mathbf{b}_a(t) + \mathbf{n}_a(t) \tag{4}$$

where ${}^I\boldsymbol{\omega}$ and ${}^I\mathbf{a}$ are the true rotational velocity and translational acceleration in the IMU local frame $\{I\}$, ${}^G\mathbf{g} \simeq [0, 0, 9.81]^T$ denotes the gravitational acceleration expressed in the global reference frame $\{G\}$, $\mathbf{n}_g$, $\mathbf{n}_a$are white Gaussian noise, and ${}^I_G\mathbf{R}$ is the rotation matrix from global to IMU local frame. We assume that $\mathbf{b}_g$ and $\mathbf{b}_a$ are known. According to the continuous time IMU kinematics [20], by denoting $\Delta T = t_{k+1} - t_k$, we have the following equations:

$${}^{I_{k+1}}_{G}\mathbf{R} = {}^{I_{k+1}}_{I_k}\Delta\mathbf{R}\,{}^{I_k}_{G}\mathbf{R} \tag{5}$$

$${}^G\mathbf{p}_{I_{k+1}} = {}^G\mathbf{p}_{I_k} + {}^G\mathbf{v}_{I_k}\Delta T - \frac{1}{2}{}^G\mathbf{g}\Delta T^2 + {}^{I_k}_G\mathbf{R}^{\top I_k}\boldsymbol{\alpha}_{I_{k+1}} \tag{6}$$

$${}^G\mathbf{v}_{I_{k+1}} = {}^G\mathbf{v}_{I_k} - {}^G\mathbf{g}\Delta T + {}^{I_k}_G\mathbf{R}^{\top I_k}\boldsymbol{\beta}_{I_{k+1}} \tag{7}$$

where we define the preintegrated IMU measurements:

$${}^{I_k}\boldsymbol{\alpha}_{I_{k+1}} = \int_{t_k}^{t_{k+1}}\int_{t_k}^{s}{}^{k}_{u}\Delta\mathbf{R}({}^I\mathbf{a}_m(u) - \mathbf{b}_a(u) - \mathbf{n}_a(u))duds$$

$${}^{I_k}\boldsymbol{\beta}_{I_{k+1}} = \int_{t_k}^{t_{k+1}}{}^{k}_{u}\Delta\mathbf{R}({}^I\mathbf{a}_m(u) - \mathbf{b}_a(u) - \mathbf{n}_a(u))du$$

We remove the global frame by integrating relative to the first frame ${}^{I_0}_{G}\mathbf{R}$ [21]:

$$
\begin{aligned}
{}^{I_k}_{I_0}\mathbf{R} &\triangleq {}^{I_k}_{I_0}\Delta\mathbf{R} && (8)\\
{}^{I_0}\mathbf{p}_{I_k} &\triangleq {}^{I_0}\mathbf{v}_{I_0}\Delta T_k - \frac{1}{2}{}^{I_0}\mathbf{g}\Delta T_k^2 + {}^{I_0}\boldsymbol{\alpha}_{I_k} && (9)\\
{}^{I_0}\mathbf{v}_{I_k} &\triangleq {}^{I_0}\mathbf{v}_{I_0} - {}^{I_0}\mathbf{g}\Delta T_k + {}^{I_0}\boldsymbol{\beta}_{I_k} && (10)
\end{aligned}
$$

Note that the time offset $\Delta T_k$ is now equal to $t_k - t_0$. Now that all parameters are transformed to the frame $\{I_0\}$, after solving the velocity and gravity under the first frame, the initial parameters in the frame $\{G\}$ can be recovered by aligning the first frame with the global gravity.

### B. Point Features

For a feature point, we consider the following relation:

$$
\begin{aligned}
\mathbf{z}_{i,k} &= \begin{bmatrix} u_{i,k} \\ v_{i,k} \end{bmatrix} + \mathbf{n}_k \\
&= \boldsymbol{\Lambda}({}^{C_k}\mathbf{p}_{f_i}) + \mathbf{n}_k && (11)\\
{}^{C_k}\mathbf{p}_{f_i} &= {}^{C}_{I}\mathbf{R}\,{}^{I_k}_{I_0}\mathbf{R}({}^{I_0}\mathbf{p}_{f_i} - {}^{I_0}\mathbf{p}_{I_k}) + {}^{C}\mathbf{p}_{I} && (12)
\end{aligned}
$$

where $\boldsymbol{\Lambda}[x\ y\ z]^\top = [x/z\ y/z]^\top$, and $\mathbf{z}_{i,k}$ is the normalized feature bearing. Here we assume that the extrinsic transform between the camera and the IMU, $\{{}^{C}_{I}\mathbf{R}, {}^{C}\mathbf{p}_{I}\}$, is sufficiently accurate. After obtaining the depth map $D$ by inference through a monocular depth network, we can express the ${}^{I_0}\mathbf{p}_{f_i}$ as:

$$
\begin{aligned}
{}^{I_0}\mathbf{p}_{f_i} &= {}^{I}_{C}\mathbf{R}\,{}^{C_0}\mathbf{p}_{f_i} + {}^{I}\mathbf{p}_{C} \\
&= z_i\,{}^{I_0}\boldsymbol{\Xi}_{C_0\to f_i} + {}^{I}\mathbf{p}_{C} \\
&= (ad_i + b){}^{I_0}\boldsymbol{\Xi}_{C_0\to f_i} + {}^{I}\mathbf{p}_{C} && (13)
\end{aligned}
$$

where ${}^{I_0}\boldsymbol{\Xi}_{C_0\to f_i} = {}^{I}_{C}\mathbf{R}[u_{i,0}\quad v_{i,0}\quad 1]^\top$ is the bearing vector of the feature point rotated into the IMU frame. The normalized 2D coordinates of the feature point in the first camera frame can be obtained by the points tracking process. Eq. (11) can be re-written as:

$$
\begin{bmatrix} 1 & 0 & -u_{i,k} \\ 0 & 1 & -v_{i,k} \end{bmatrix} {}^{C_k}\mathbf{p}_{f_i} \triangleq \boldsymbol{\Gamma}_{i,k}{}^{C_k}\mathbf{p}_{f_i} = \mathbf{0}_2 \tag{14}
$$

This illustrates that ideally the normalized feature observation and projected feature differences should be zero. We use this constraint to build the point features linear system. We can substitute Eq. (12) and Eq. (13) to Eq. (14) to get:

$$
\boldsymbol{\Gamma}_{i,k}({}^{C}_{I}\mathbf{R}\,{}^{I_k}_{I_0}\mathbf{R}([(ad_i + b){}^{I_0}\boldsymbol{\Xi}_{C_0\to f_i} + {}^{I}\mathbf{p}_{C}] - {}^{I_0}\mathbf{p}_{I_k}) + {}^{C}\mathbf{p}_{I}) = \mathbf{0}_2 \tag{15}
$$

Eq. (9) is then substituted and rewritten to get the following linear system:

$$
\begin{aligned}
\mathbf{A}_{i,k}\mathbf{x}' &= \mathbf{b}_{i,k} && (16)\\
\mathbf{A}_{i,k} &= \boldsymbol{\Upsilon}_{i,k}[\mathbf{B}_i \quad -\boldsymbol{\Delta}\mathbf{T}_k \quad \frac{1}{2}\boldsymbol{\Delta}\mathbf{T}_k^2] && (17)\\
\mathbf{b}_{i,k} &= \boldsymbol{\Upsilon}_{i,k}{}^{I_0}\boldsymbol{\alpha}_{I_k} - \boldsymbol{\Upsilon}_{i,k}{}^{I}\mathbf{p}_{C} - \boldsymbol{\Gamma}_{i,k}{}^{C}\mathbf{p}_{I} && (18)\\
\mathbf{B}_i &= [d_i{}^{I_0}\boldsymbol{\Xi}_{C_0\to f_i} \quad {}^{I_0}\boldsymbol{\Xi}_{C_0\to f_i}] && (19)
\end{aligned}
$$

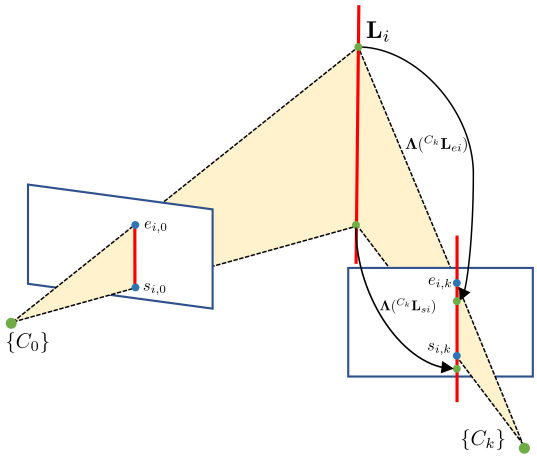


Fig. 2. The projection process of the 3D line. The 3D line $\mathbf{L}_i$ observed from both the $\{C_0\}$ and $\{C_k\}$ frame are shown. The projection point of the 3D line $\mathbf{L}_i$ should be on the line formed by the points $s_{i,k}$ and $e_{i,k}$.

where $\boldsymbol{\Delta}\mathbf{T}_k = \Delta T_k\mathbf{I}_3$ and $\boldsymbol{\Upsilon}_{i,k} = \boldsymbol{\Gamma}_{i,k}{}^{C}_{I}\mathbf{R}\,{}^{I_k}_{I_0}\mathbf{R}$. We stack the observations of all point features to obtain a more robust linear system:

$$
\mathbf{A}\mathbf{x}' = \mathbf{b} \tag{20}
$$

Suppose that $M$ point features are tracked stably from $N$ keyframes, $\mathbf{A} \in \mathbb{R}^{2MN\times 8}$ and $\mathbf{b} \in \mathbb{R}^{2MN}$. In this way, the state size remains constant regardless of the change in the number of features.

### C. Line Features

Line features can provide richer geometric information about the environment than point features. How to use the line features to build a linear system with a similar structure to the equations of the point features is the key to consistency. In our work, we build linear systems based on the re-projection of points on the 3D line to the normalized plane still on the observation of the 3D line. As shown in Fig. 2, $\mathbf{L}_i$ denotes a 3D line, ${}^{C_k}\mathbf{L}_{si}$ and ${}^{C_k}\mathbf{L}_{ei}$ are the start and end points of the 3D line in the $\{C_k\}$ frame (Note that the straight lines are infinite, in this case it is the 3D point that corresponds to the start and end points of the line in the normalized plane), and $s_{i,k}$ and $e_{i,k}$ denote the starting and ending measurements of the line feature in the $\{C_k\}$ frame. After a point on the 3D line $\mathbf{L}_i$ is projected onto the normalization plane, the projected point should be on the straight line joined by $s_{i,k}$ and $e_{i,k}$. Therefore, the line projection relation can be obtained:

$$
\begin{aligned}
\boldsymbol{\Phi}_{i,k}{}^{C_k}\mathbf{L}_{si} &= 0 && (21)\\
\boldsymbol{\Phi}_{i,k} &\triangleq [e_{yi,k} - s_{yi,k}, s_{xi,k} - e_{xi,k}, \\
&\qquad e_{xi,k}s_{yi,k} - s_{xi,k}e_{yi,k}] && (22)
\end{aligned}
$$

where $(s_{xi,k}, s_{yi,k})$ and $(e_{xi,k}, e_{yi,k})$ are the starting and ending coordinates of the observations of the 3D line $\mathbf{L}_i$ in the $\{C_k\}$ frame. Eq. (21) shows that the projected point is on the observation line. ${}^{C_k}\mathbf{L}_{si}$ is the 3D point corresponding to the line feature observation point, but is still a point with similar properties to ${}^{C_k}\mathbf{p}_{f_i}$. Then ${}^{C_k}\mathbf{L}_{si}$ also yields a similar

**Algorithm 1** Linear System with RANSAC

**Input:** Give $M$ point features and $Q$ line features from $N$ keyframes for $i \in \{1, \cdots, M\}$,$j \in \{1, \cdots, Q\}$, $k \in \{1, \cdots, N\}$,minimal problem size $M_{minp}$,$N_{minp}$, $M_{minl}$,$N_{minl}$, maximum number of iterations $K$, thresholds $d_p, d_l, \gamma_p, \gamma_l$

**Output:** Robust solution $\mathbf{x}'_{best}$

$e_{best} \leftarrow \infty$
**for** $i \in \{1, \cdots, K\}$ **do**
  $\mathcal{S}_p \leftarrow$ Rand. sample $N_{minp}$ meas. from $M_{minp}$ point feats.
  $\mathcal{S}_l \leftarrow$ Rand. sample $N_{minl}$ meas. from $M_{minl}$ line feats.
  $\mathbf{C}'_s, \mathbf{d}'_s \leftarrow$ Stack blocks $i, k \in \mathcal{S}_p$ and $j, k \in \mathcal{S}_l$
  $a, b, {}^{I_0}\mathbf{v}_{I_0}, {}^{I_0}\mathbf{g} \leftarrow \text{solve}(C'_s, d'_s)$
  **for** $i, k$ not in $\mathcal{S}_p$ **do**
    $\mathbf{r} \leftarrow \mathbf{A}_{ik}[a \quad b \quad {}^{I_0}\mathbf{v}_{I_0}^\top \quad {}^{I_0}\mathbf{g}^\top]^\top - \mathbf{b}_{ik}$
    **if** $||\mathbf{r}|| < \gamma_p$ **then**
      $\mathcal{S}_p \leftarrow \mathcal{S}_p \cup (i, k)$
    **end if**
  **end for**
  **for** $j, k$ not in $\mathcal{S}_l$ **do**
    $\mathbf{r} \leftarrow \mathbf{M}_{jk}[a \quad b \quad {}^{I_0}\mathbf{v}_{I_0}^\top \quad {}^{I_0}\mathbf{g}^\top]^\top - \mathbf{n}_{jk}$
    **if** $||\mathbf{r}|| < \gamma_l$ **then**
      $\mathcal{S}_l \leftarrow \mathcal{S}_l \cup (j, k)$
    **end if**
  **end for**
  **if** $|\mathcal{S}_p| \geq d_p$ and $|\mathcal{S}_l| \geq d_l$ **then**
    $\mathbf{C}', \mathbf{d}' \leftarrow$ Stack blocks $i, k \in \mathcal{S}_p$ and $j, k \in \mathcal{S}_l$
    $a, b, {}^{I_0}\mathbf{v}_{I_0}, {}^{I_0}\mathbf{g} \leftarrow \text{solve}(\mathbf{C}', \mathbf{d}')$
    $\mathbf{r} \leftarrow \mathbf{C}'[a \quad b \quad {}^{I_0}\mathbf{v}_{I_0}^\top \quad {}^{I_0}\mathbf{g}^\top]^\top - \mathbf{d}'$
    **if** $||\mathbf{r}|| < e_{best}$ **then**
      $e_{best} \leftarrow ||\mathbf{r}||$
      $\mathbf{x}'_{best} \leftarrow [a \quad b \quad {}^{I_0}\mathbf{v}_{I_0}^\top \quad {}^{I_0}\mathbf{g}^\top]^\top$
    **end if**
  **end if**
**end for**

relationship to Eq. (12), which we combine with Eq. (13) and Eq. (21) to get:

$$\mathbf{M}_{i,k}\mathbf{x}' = n_{i,k} \tag{23}$$

$$\mathbf{M}_{i,k} = \mathbf{\Psi}_{i,k}[\mathbf{B}_i \quad -\mathbf{\Delta}\mathbf{T}_k \quad \frac{1}{2}\mathbf{\Delta}\mathbf{T}_k^2] \tag{24}$$

$$n_{i,k} = \mathbf{\Psi}_{i,k}\,{}^{I_0}\boldsymbol{\alpha}_{I_k} - \mathbf{\Psi}_{i,k}\,{}^{I}\mathbf{p}_C - \mathbf{\Phi}_{i,k}\,{}^{C}\mathbf{p}_I \tag{25}$$

where $\mathbf{\Psi}_{i,k} = \mathbf{\Phi}_{i,k}\,{}^{C}_{I}\mathbf{R}\,{}^{I_k}_{I_0}\mathbf{R}$. Similarly, a similar equation can be obtained for the end points, and the two equations stacked together form an equation for a single observation of a line feature. We can obtain the linear system consisting of line features by stacking all observations of all line features together:

$$\mathbf{M}\mathbf{x}' = \mathbf{n} \tag{26}$$

Suppose that $Q$ line features are tracked stably from $N$ keyframes, $\mathbf{M} \in \mathbb{R}^{2QN \times 8}$ and $\mathbf{n} \in \mathbb{R}^{2QN}$.

The linear system composed of line features is similar in spirit to the linear system of point features described above. We combine Eq. (20) and Eq. (26), which have the same unknowns, to obtain the complete linear system. We solve the linear system following the solution method of [21] in order to utilize our prior knowledge of the magnitude of gravity.

### *D. Outlier Rejection in Point-Line Linear System*

Because the linear system is built from a stack of observations of features, we can pick observations with smaller errors in each feature to build the linear system, which makes the method well suited to use RANSAC to improve robustness.

An overview of our RANSAC approach can be seen in Algo. 1. We build the minimum problem solving initial parameters by randomly selecting the observations of point features and line features, and then build the interior set by using the solved parameters to compute the re-projection error in the set of point features and the set of line features, respectively. We choose the interior set solution with the minimum error as the best result. The minimum problem solution for the RANSAC problem requires 2 features and 3 views [11], in practice we use 4 features (2 point features, 2 line features) and 3 views to improve the solution performance.

### *E. Nonlinear Optimization*

With nonlinear optimization, we further refine the initial parameters. We recover the 3D positions of the point features via Eq. (13), and solve the parameters in the $\{G\}$ frame by aligning the first frame of reference with that of gravity. In our work, We use two parameterizations to represent the 3D line, where Plücker line coordinates are used for transformations and projections, and a four-parameter orthonormal representation is used for optimization. The 3D spatial line $\mathbf{L}$ in the Plücker coordinates is denoted as $\mathbf{L} = (\mathbf{n}^\top, \mathbf{d}^\top)^\top \in \mathbb{R}^6$, with $\mathbf{n}^\top\mathbf{d} = 0$, where $\mathbf{n}$ is the normal vector of the plane determined by the line and the coordinate origin, and $\mathbf{d}$ is the direction vector of the line. The Plücker coordinates can be determined from the two endpoints of the line:

$$\begin{bmatrix} {}^{C_k}\mathbf{n}_i \\ {}^{C_k}\mathbf{d}_i \end{bmatrix} = \begin{bmatrix} \lfloor {}^{C_k}\mathbf{L}_{si} \times \rfloor\, {}^{C_k}\mathbf{L}_{ei} \\ {}^{C_k}\mathbf{L}_{ei} - {}^{C_k}\mathbf{L}_{si} \end{bmatrix} \tag{27}$$

where ${}^{C_k}\mathbf{n}_i$ and ${}^{C_k}\mathbf{d}_i$ denote the Pluck coordinates of the 3D line $\mathbf{L}_i$ in the $\{C_k\}$ frame, $\lfloor \times \rfloor$ is skew-symmetric matrix operator. We can obtain the Plücker coordinates of the 3D line by using Eq. (12), Eq. (13) and Eq. (27). Furthermore, the geometric transformation of the Plücker coordinates can be defined as:

$$\begin{bmatrix} {}^{G}\mathbf{n}_i \\ {}^{G}\mathbf{d}_i \end{bmatrix} = \begin{bmatrix} {}^{G}_{C_k}\mathbf{R} & \lfloor {}^{G}\mathbf{p}_{C_k} \times \rfloor\, {}^{G}_{C_k}\mathbf{R} \\ \mathbf{0} & {}^{G}_{C_k}\mathbf{R} \end{bmatrix} \begin{bmatrix} {}^{C_k}\mathbf{n}_i \\ {}^{C_k}\mathbf{d}_i \end{bmatrix} \tag{28}$$

The Plücker coordinates have implicit constraints that make it difficult to apply them in optimization. We follow [22] to use the orthonormal representation of the line $\mathcal{O} = [\boldsymbol{\psi}, \boldsymbol{\phi}]^\top \in \mathbb{R}^4$ in VI-BA, where $\boldsymbol{\psi}$ and $\boldsymbol{\phi}$ contain information about the rotation angle and distance, respectively. The re-projection

TABLE I

INITIALIZATION WINDOW ATE (DEG/M) ON EUROC MAV (0.5S, 5KFS)

| Algorithm | V101 | | V102 | | V103 | | V201 | | V202 | | V203 | | Average | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Ori. | Pos. | Ori. | Pos. | Ori. | Pos. | Ori. | Pos. | Ori. | Pos. | Ori. | Pos. | Ori. | Pos. |
| Merrill [11] | 1.111 | 0.019 | 1.065 | **0.018** | 2.234 | 0.030 | **0.882** | **0.011** | 1.442 | 0.015 | 1.778 | 0.040 | 1.419 | 0.022 |
| Baseline | 1.250 | 0.031 | 1.170 | 0.025 | **1.037** | **0.016** | 1.470 | 0.025 | 1.337 | 0.036 | 1.620 | 0.034 | 1.314 | 0.028 |
| Ours | **1.088** | **0.018** | **0.895** | 0.022 | 1.162 | 0.024 | 0.959 | 0.014 | **0.823** | **0.011** | **1.375** | **0.033** | **1.050** | **0.020** |

error of a linear measurement is defined as the distance from the endpoint to the projected line.

In our optimization process, the state vector is defined as:

$$\mathbf{x}_{mle} = [\mathbf{x}_{I_0}^\top, \cdots, \mathbf{x}_{I_N}^\top, {}^G\mathbf{p}_{f_1}^\top, \cdots, {}^G\mathbf{p}_{f_M}^\top, {}^G\mathcal{O}_1^\top, \cdots, {}^G\mathcal{O}_Q^\top]^\top \tag{29}$$

$$\mathbf{x}_{I_k} = [{}^{I_k}_G\overline{q},\ {}^G\mathbf{p}_{I_k}^\top,\ {}^G\mathbf{v}_{I_k}^\top,\ \mathbf{b}_{g,k}^\top,\ \mathbf{b}_{a,k}^\top]^\top \tag{30}$$

Instead of optimizing the parameters $a$ and $b$ we directly optimize the recovered spatial position, which can more accurately help recover the coordinate values for each feature.

We define the following optimization problem:

$$\underset{x_{mle}}{\arg\min}\, \mathbb{C}_I + \mathbb{C}_C + \mathbb{C}_L + \mathbb{C}_P \tag{31}$$

where includes inertial $\mathbb{C}_I$, camera $\mathbb{C}_C$, line $\mathbb{C}_L$, and prior $\mathbb{C}_P$ cost terms. The inertial cost term is defined as follows [20]:

$$\mathbb{C}_I \triangleq \sum_k ||\mathbf{x}_{I_{k+1}} \boxminus \mathbf{f}(\mathbf{x}_{I_k}, {}^I\mathbf{a}_{m_k}, {}^I\boldsymbol{\omega}_{m_k})||^2_{\mathbf{Z}_k} \tag{32}$$

where $\mathbf{Z}_k$ is the linearized measurement noise covariance. The point re-projection cost is defined as [7]:

$$\mathbb{C}_C \triangleq \sum_{i,k} ||\mathbf{z}_{i,k} - \mathbf{h}(\mathbf{x}_{mle})||^2_{\mathbf{R}_i} \tag{33}$$

where $\mathbf{h}(\cdot)$ includes the intrinsic distortion, projection, and extrinsic transformation, and $\mathbf{R}_i$ is the image pixel noise covariance. The line re-projection cost is defined as [22]:

$$\mathbb{C}_L \triangleq \sum_{j,k} ||\mathbf{d}(\mathbf{L}_{j,k}, \mathbf{r}(\mathbf{x}_{mle}))||^2_{\boldsymbol{\Theta}_i} \tag{34}$$

where $\mathbf{r}(\cdot)$ is the projection function of the line, $\mathbf{d}(\cdot)$ is the distance function from the point to the line, and $\boldsymbol{\Theta}_i$ is the line feature noise covariance. In the prior cost, in addition to the unobservable initial global position and yaw rotation, since the gyroscope and especially accelerometer biases can nearly be unobservable under short windows, we also provide reasonable prior information for the gyroscope and accelerometer biases. The prior cost defined as [21]:

$$\mathbb{C}_P \triangleq ||\mathbf{x}_{mle} \boxminus \breve{\mathbf{x}}_{mle}||^2_{\boldsymbol{\Omega}_P^{-1}} \tag{35}$$

where $\breve{\mathbf{x}}_{mle}$ is a fixed linearization point and $\boldsymbol{\Omega}_P$ is the prior information matrix.

TABLE II

RESULTS OF ABLATION STUDY ON EUROC MAV (0.5S, 5KFS)

| | | point-only | line-only | point-line |
|---|---|---|---|---|
| V101 | Vel. | **0.063** | 0.180 | 0.082 |
| | Grav. | 3.180 | 3.282 | **2.890** |
| | Scal. | 9.561 | 164.870 | **7.671** |
| V102 | Vel. | 0.080 | 0.130 | **0.071** |
| | Grav. | 1.091 | 0.978 | **0.847** |
| | Scal. | 11.390 | 24.560 | **8.021** |
| V103 | Vel. | 0.120 | 0.081 | **0.080** |
| | Grav. | 1.453 | 1.555 | **1.132** |
| | Scal. | 13.850 | **11.270** | 11.620 |
| V201 | Vel. | 0.062 | 0.050 | **0.038** |
| | Grav. | 1.113 | **0.907** | 0.979 |
| | Scal. | 16.501 | 16.920 | **12.200** |
| V202 | Vel. | 0.091 | 0.084 | **0.040** |
| | Grav. | 1.310 | 1.030 | **0.902** |
| | Scal. | 12.312 | 12.550 | **4.278** |
| V203 | Vel. | **0.050** | 0.120 | 0.080 |
| | Grav. | **1.082** | 1.717 | 1.349 |
| | Scal. | 7.691 | 15.870 | **8.270** |
| Average | Vel. | 0.078 | 0.108 | **0.065** |
| | Grav. | 1.538 | 1.578 | **1.350** |
| | Scal. | 11.884 | 41.007 | **8.677** |

## IV. EXPERIMENTS

We conduct experiments on two popular datasets EuRoC MAV and TUM-VI to validate our proposed VIO initialization method. Because we focus on monocular visual-inertial initialization, only the left camera image is used for both datasets.

In our experiments, we mainly evaluate the absolute trajectory error ATE. We align the ground truth and the estimated trajectory of the first frame, the gravity errors and scale accuracy will directly affect the orientation and position ATE. In the ablation study, we also calculate the scale accuracy, gravity error, and velocity error to evaluate the algorithm performance in detail. Similar to [10] [11], we divide each sequence into 10s windows run our initialization algorithm for each entry point, and average the results from each successful run. After the initial parameters are computed and the covariance is successfully recovered, it is fed into the OpenVINS [7] backend to get the VIO tracking accuracy. Bad initialization results will lead to poor VIO tracking accuracy. For line features, we adopt the LSD algorithm [23] available in OPENCV3 [24] to detect line segments and use LBD [25] to describe for matching. Fisheye distortion should be considered for experiments on line feature detection on the TUM-VI dataset. Line lengths shorter than 25 pixels are rejected and the detection and matching process is run in another thread.

For the monocular depth network, we directly use the

TABLE III

VISUAL-INERTIAL ODOMETRY TRACKING ATE (DEG/M) ON TUM-VI (0.3S,5KFS)

| Algorithm | Room1 | | Room2 | | Room3 | | Room4 | | Room5 | | Room6 | | Average | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Ori. | Pos. | Ori. | Pos. | Ori. | Pos. | Ori. | Pos. | Ori. | Pos. | Ori. | Pos. | Ori. | Pos. |
| Merrill [11] | 0.882 | 0.036 | **0.825** | 0.044 | 1.627 | 0.072 | 1.637 | 0.085 | 1.533 | 0.064 | **0.782** | 0.050 | 1.214 | 0.059 |
| Baseline | 1.931 | 0.081 | 3.366 | 0.170 | 1.508 | **0.049** | 1.632 | 0.115 | **1.328** | 0.067 | 1.887 | 0.196 | 1.942 | 0.113 |
| Ours | **0.827** | **0.032** | 0.836 | **0.037** | **1.371** | 0.060 | **0.896** | **0.029** | 1.405 | **0.062** | 1.030 | **0.045** | **1.061** | **0.044** |

TABLE IV

INITIALIZATION WINDOW ATE (DEG/M) AND SCALE ERROR(%) ON TUM-VI (0.3S, 5KFS)

| | | Merrill [11] | Baseline | Ours |
|---|---|---|---|---|
| Room1 | Ori. | 1.375 | 2.027 | **1.018** |
| | Pos. | 0.010 | 0.013 | **0.008** |
| | Scal. | **3.50** | 9.41 | 3.71 |
| Room2 | Ori. | **0.851** | 1.212 | 1.031 |
| | Pos. | **0.007** | 0.022 | 0.011 |
| | Scal. | **1.47** | 10.40 | 6.30 |
| Room3 | Ori. | 1.707 | **1.100** | 1.625 |
| | Pos. | 0.014 | **0.010** | 0.017 |
| | Scal. | 6.95 | **5.04** | 15.90 |
| Room4 | Ori. | 1.430 | 1.689 | **0.943** |
| | Pos. | 0.013 | 0.026 | **0.007** |
| | Scal. | 11.80 | 37.40 | **9.25** |
| Room5 | Ori. | 1.572 | 2.025 | **1.270** |
| | Pos. | 0.010 | 0.014 | **0.009** |
| | Scal. | **6.45** | 8.50 | 9.11 |
| Room6 | Ori. | **0.710** | 1.598 | 0.909 |
| | Pos. | 0.013 | 0.029 | **0.010** |
| | Scal. | **8.68** | 19.57 | 10.90 |
| Average | Ori. | 1.274 | 1.609 | **1.133** |
| | Pos. | 0.011 | 0.019 | **0.010** |
| | Scal. | **6.475** | 15.053 | 9.195 |

TABLE V

PERCENT OF SUCCESSFUL INITIALIZATIONS ON TUM-VI (AVERAGED OVER ALL ROOMS) WITH 5KFS AND 0.3S WINDOW

| Algorithm | 60 | 45 | 30 | 15 |
|---|---|---|---|---|
| Merrill [11] | **100.00** | **98.75** | 97.50 | 55.00 |
| Baseline | 73.97 | 68.49 | 64.38 | 17.80 |
| Ours | **100.00** | 98.63 | **98.63** | **84.93** |

pre-trained model V21 small from the MiDaS [26] family, which is a lightweight network designed specifically for mobile small devices. Since it will only be run once during initialization, we did not design a new thread for it. Note that this monocular depth network actually outputs affine-invariant inverse depth maps, which we scaled to $[1, 2]$ using min-max normalization before use to maintain numerical stability.

In addition, we use the initialization algorithm in OpenVINS [7] as baseline, which is an initialization method using only point features. Unless otherwise stated, 50 point features and 25 line features are used in our method and 75 point features are used in the baseline. Since the state-of-the-art method [11] is not open source, we cite its results for comparison where applicable and reasonable.

### A. EuRoC MAV Dataset

The EuRoC MAV dataset is visual-inertial data collected from a micro air vehicle (MAV) that provides ground truth of velocity, trajectory and biases. We selected 5 keyframes at even intervals from the 0.5s window for initialization. As shown in Table I, we first report the orientation and position of the ATE for comparison with the method [11]. Our method achieves optimal performance overall, with a large improvement in orientation ATE particularly.

To evaluate the proposed method in more detail, we also performed ablation study on the EuRoC MAV. We build linear system with RANSAC using only point features and only line features according to Eq. ( 20) and Eq. ( 26), respectively, and then evaluated the velocity error (m/s), gravity error (deg), and scale error (%) after VI-BA. We are still selecting 5 keyframes at even intervals from the 0.5s window. To ensure consistency, the same number of features are used, 45 points for point-only, 45 lines for line-only, and 30 points and 15 lines for point-line fusion. As shown in Table II, the point-only method is slightly better than the line-only method, but both are surpassed by the point-line fusion method. We show that the initialization performance of pure point features can be improved by adding line features.

### B. TUM-VI Dataset

The TUM-VI dataset is a comprehensive, multimodal dataset that is well suited for conducting research related to visual-inertial odometry. Without any additional training, the pre-trained MidaS [26] V21 small model still produces reasonable output on the TUM-VI dataset. The fisheye image needs to be undistorted before performing line feature detection on it.

To better validate our method, we designed a more challenging extreme low-parallax configuration, selecting 5 keyframes at evenly spaced intervals over a 0.3s initialization window. Table IV reports the ATE and scale error within the initialization window, where our method is overall superior in terms of orientation and position ATE, although slightly worse in terms of scale error.

Furthermore, in order to more intuitively see the impact of initialization on tracking performance, we input the initial parameters after VI-BA to the backend to get the VIO tracking accuracy. Table III reports the VIO tracking accuracy after initialization. Our approach achieves the best performance, showing that this initialization method is significantly improved for downstream applications. Note that we count the results of a successful initialization, which is considered successful if all steps are completed and the covariance can be recovered from VI-BA. 73 sequences were obtained by dividing the dataset, our method successfully initialized 73 times and the baseline method successfully initialized

62 times, our method is better in terms of robustness and accuracy.

To verify the robustness of our method, we try to reduce the number of point features and line features used in the initialization process. Keeping the extreme low-parallax configuration constant and fixing the ratio of the number of points to the number of lines, we counted the success rate of initialization using 60 (40 points and 20 lines for ours), 45 (30 points and 15 lines for ours), 30 (20 points and 10 lines for ours), and 15 (10 points and 5 lines for ours) features. As shown in Table V, our method not only has an extremely high success rate when the number of features is sufficient, but also still guarantees the initialization success rate when the amount of features is only 15, demonstrating robustness to severe reductions in the number of features.

## V. CONCLUSIONS

In this paper, we have presented a state-of-the-art method for monocular visual-inertial initialization with the help of line features and learned depth map. Based on the solution of point features, a new closed-form solution of line features is proposed, which can be easily combined together to build a more robust linear system with RANSAC. The affine-invariant depth map is provided through a monocular depth network, where we express the feature depth as a linear function of the prior depth map, greatly reducing the number of initial parameters to be estimated. Experiments under extreme low-parallax scenarios (0.5s window on EuRoC MAV, 0.3s window on TUM-VI) are performed and our method can initialize more frequently, robustly, and accurately in different challenging scenarios. Extensive experiments have shown that using both point features and line features in the initialization can effectively improve the accuracy and robustness of the initialization. However, our method does not work under zero excitation, which is our future work. We also hope to further improve the accuracy and robustness by adding plane feature as additional constraints.

## ACKNOWLEDGMENT

This work is supported by the National Key R&D Program of China (No.2023YFF0615800), the funding from Sichuan Province under grant 2023YFG0334, as well as the funding from Sichuan University under grant 2020SCUNG205.

## REFERENCES

[1] G. Huang, "Visual-inertial navigation: A concise review," in *2019 international conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 9572–9582.

[2] K. J. Wu, C. X. Guo, G. Georgiou, and S. I. Roumeliotis, "Vins on wheels," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5155–5162.

[3] Google, "ARCore." https://developers.google.com/ar.

[4] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[5] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[6] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," in *Proceedings 2007 IEEE international conference on robotics and automation*. IEEE, 2007, pp. 3565–3572.

[7] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "Openvins: A research platform for visual-inertial estimation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4666–4672.

[8] K. Sun, K. Mohta, B. Pfrommer, M. Watterson, S. Liu, Y. Mulgaonkar, C. J. Taylor, and V. Kumar, "Robust stereo visual inertial odometry for fast autonomous flight," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 965–972, 2018.

[9] 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*, vol. 34, no. 3, pp. 314–334, 2015.

[10] Y. Zhou, A. Kar, E. Turner, A. Kowdle, C. X. Guo, R. C. DuToit, and K. Tsotsos, "Learned monocular depth priors in visual-inertial initialization," in *European conference on computer vision*. Springer, 2022, pp. 552–570.

[11] N. Merrill, P. Geneva, and S. K. C. C. G. Huang, "Fast monocular visual-inertial initialization leveraging learned single-view depth," in *Robotics: Science and Systems (RSS)*, vol. 2, 2023.

[12] R. Mur-Artal and J. D. Tardós, "Visual-inertial monocular slam with map reuse," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 796–803, 2017.

[13] Q. Fu, J. Wang, H. Yu, I. Ali, F. Guo, Y. He, and H. Zhang, "Pl-vins: Real-time monocular visual-inertial slam with point and line features," *arXiv preprint arXiv:2009.07462*, 2020.

[14] A. Martinelli, "Closed-form solution of visual-inertial structure from motion," *International journal of computer vision*, vol. 106, no. 2, pp. 138–152, 2014.

[15] H. Liu, J. Qiu, and W. Huang, "Integrating point and line features for visual-inertial initialization," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 9470–9476.

[16] G. Evangelidis and B. Micusik, "Revisiting visual-inertial structure-from-motion for odometry and slam initialization," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1415–1422, 2021.

[17] T. Qin and S. Shen, "Robust initialization of monocular visual-inertial estimation on aerial robots," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 4225–4232.

[18] L. Xu, H. Yin, T. Shi, D. Jiang, and B. Huang, "Eplf-vins: Real-time monocular visual-inertial slam with efficient point-line flow features," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 752–759, 2022.

[19] T.-C. Dong-Si and A. I. Mourikis, "Estimator initialization in vision-aided inertial navigation with unknown camera-imu calibration," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 1064–1071.

[20] K. Eckenhoff, P. Geneva, and G. Huang, "Closed-form preintegration methods for graph-based visual–inertial navigation," *The International Journal of Robotics Research*, vol. 38, no. 5, pp. 563–586, 2019.

[21] P. Geneva and G. Huang, "Openvins state initialization: Details and derivations," Tech. Rep. RPNG-2022-INIT, University of Delaware, 2022. Available: https …, Tech. Rep.

[22] Y. He, J. Zhao, Y. Guo, W. He, and K. Yuan, "Pl-vio: Tightly-coupled monocular visual–inertial odometry using point and line features," *Sensors*, vol. 18, no. 4, p. 1159, 2018.

[23] R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "Lsd: A fast line segment detector with a false detection control," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 4, pp. 722–732, 2008.

[24] A. Kaehler and G. Bradski, *Learning OpenCV 3: computer vision in C++ with the OpenCV library*. " O'Reilly Media, Inc.", 2016.

[25] L. Zhang and R. Koch, "An efficient and robust line segment matching approach based on lbd descriptor and pairwise geometric consistency," *Journal of visual communication and image representation*, vol. 24, no. 7, pp. 794–805, 2013.

[26] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 3, pp. 1623–1637, 2020.