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

LiDAR-Visual-Inertial Tightly-coupled Odometry with Adaptive Learnable Fusion Weights

Vsevolod Hulchuk Jan Bayer Jan Faigl

***Abstract*— In this paper, we address the sensitivity of the 3D LiDAR-based localization to environmental structural ambiguity. Although existing approaches employ additional sensors, such as cameras and inertial measurement units, to account for such ambiguities, multi-sensor localization is still an open problem. Limitations are from the need to tune fusion parameters to compensate for limited ambiguity detection manually. Therefore, we propose a feature-based localization method that learns the fusion parameters using ground truth and thus supports autonomous mobile robotic systems in new locations. The method combines planar surface LiDAR features with close and far camera features, and its further advantage is an online adjustment of the feature weights based on the measured environment ambiguity. The evaluation has been performed on the existing M2DGR dataset and custom dataset with geometrical ambiguities. The proposed method is competitive to or outperforms the existing LiDAR-based methods F-LOAM and LIO-SAM and the Visual-Inertial localization method VINS-Mono. Based on the reported results, the proposed method is a vital combination of LiDAR-based and visual features.**

I. Introduction

Vehicle localization is a critical problem studied in the context of autonomous navigation, especially in deployments without access to the Global Navigation Satellite System (GNSS), such as areas close to or under immense structures that include bridges, tunnels, or urban canyons. In GNSS-denied sites, a vehicle needs to use its onboard sensors to localize itself, and two main approaches can be found in the literature. If a prior map of the environment is available, localization may benefit from the map when estimating the robot pose [1]. However, when the map is not available, the Simultaneous Localization and Mapping (SLAM) [2] is a de facto standard approach to localize the robot within the map being created by the robot's exteroceptive sensors and range measurements. Besides, the estimation of the robot's relative ego-motion is an essential part of robot localization. It is to estimate the robot's transformation from the previous robot pose to the current one, and it is called odometry, regardless of whether it is based on visual image processing or wheels rolling.

Various onboard sensors can be used for localization. In this work, we focus on multi-modal sensing of the Light Detection and Ranging (LiDAR), camera, and Inertial Measurement Unit (IMU). LiDAR-based localization uses scan-to-scan or scan-to-map matching to estimate the ego-motion in a structured environment where it can provide sufficient performance [3]. However, it is sensitive to structure-less or degenerated parts of the environment, such as long corridors or open fields, where the scan alignment is ambiguous. Moreover, a proper matching initialization is needed to reach accurate results during rapid motions [4] that can be addressed by combining LiDAR with IMU as in LIO-SAM [5]. Ego-motion estimation from IMU measurements is not affected by the robot's surroundings, but it suffers from accumulated drift, leading to a high localization error in the long term. Therefore, cameras can be used in structure-less environments to support the localization as visual information might be robust against structural ambiguity despite the need for the presence of texture on the seen objects.

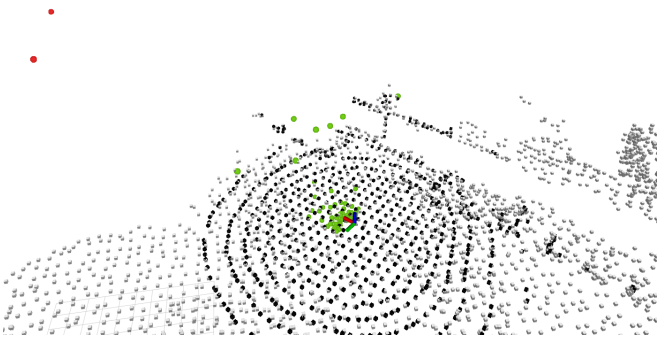


Fig. 1. LiDAR and Visual-Inertial features. Gray cubes represent the surface planar points map; black cubes are the current LiDAR features; green spheres are close visual features; and red spheres are far visual features.

Several methods [4] [6] [7] exist on multi-modal fusion; however, their proper settings can be laborious, and it might require a deep understanding of the particular method used. Therefore, we propose a LiDAR-Visual-Inertial fusion approach to combine 3D LiDAR scans, monocular camera images, and IMU measurements in odometry estimation that dynamically adjusts the weights of the sensors by changing the weight of the LiDAR features using measured structural ambiguity of LiDAR scans. It is explicitly called LiDAR-Visual-Inertial, as the IMU measurements are used in extracting 3D visual features from the camera images that are later fused with 3D LiDAR features. However, the sensor fusion is based on splitting visual features into far and close features used for rotation and full transformation estimation, respectively; see Fig. 1. Distance thresholds might be set empirically; however, we propose to learn them together with

The authors are with the Faculty of Electrical Engineering, Czech Technical University, Technická 2, 166 27, Prague, Czech Republic {hulchvse|bayerja1|faiglj}@fel.cvut.cz

The work has been supported by the Czech Science Foundation (GAČR) under research project No. 22-05762S and by the European Union under the project ROBOPROX - Robotics and advanced industrial production (reg. No. CZ.02.01.01/00/22_008/0004590). The support under grant No. SGS22/168/OHK3/3T/13 to the first author is also gratefully acknowledged.

sensor fusion weights to achieve improved performance in the target environment. In contrast with the LiDAR weight, which is adaptive within one environment depending on the current scan, the distance thresholds remain constant after being learned. The focus of the paper is on learning the fusion parameters to leverage the best out of the specific feature sets. Therefore, the comparative study is against the LiDAR method F-LOAM [8] and the Visual-Inertial method VINS-Mono [9], which use the fused features but separately, and not other LiDAR-Visual-Inertial fusion methods. The main contributions of the proposed work are considered as follows.

- A tightly-coupled sensor fusion system for the ego-motion estimation with the adaptive fusion weights.
- A pipeline for the system to learn the fusion parameters to improve system performance in target environments, avoiding laborious empirical tuning.

The remainder of the paper is organized as follows. An overview of the related localization and sensory fusion approaches is presented in the following section. The proposed method is detailed in Section III. Achieved experimental results and comparison of the proposed method with existing approaches are reported in Section IV. Finally, the paper is concluded in Section V.

II. Related Work

Robot localization using onboard sensors is addressed by various methods benchmarked in publicly available datasets, such as KITTI [3]. Most of the top ten performing methods in the KITTI odometry benchmark [10] are LiDAR-based methods, including LOAM [11]. LOAM uses surface planar and edge features to estimate the robot's displacement as LiDAR-based odometry has become a standard approach for the LiDAR-based method, such as the LeGO-LOAM [12] that adds ground segmentation, and F-LOAM [8], making the method less computationally intensive. Although widely used, the KITTI dataset is captured in a city with a lack of challenging environments with high ambiguity of the LiDAR measurements, which can be encountered in monotonous or open areas, such as empty parking lots, fields, and areas close to water surfaces. Hence, custom datasets are collected for performance evaluation of state-of-the-art methods in a particular type of environments [13] and [14].

The ambiguity of LiDAR measurements using visual and inertial sensors is addressed in [15]. The combination is also used for visual and visual-inertial odometry in ORB-SLAM3 [16] and VINS-Mono [9]. Both methods are based on extracting the visual features from the camera images, tracking them among images, and estimating the camera motion based on the tracked features. However, IMU measurements can be used to determine the visual odometry calculated from monocular camera images [9].

The LiDAR-Visual-Inertial (LVI) odometry is of primary interest to take the best from each sensor by measurement fusion. An extensive review of the multi-sensor fusion methods, including LVI-based SLAM, is provided in [17] with two main classes of the LVI methods studied in the literature: loosely coupled and tightly coupled.

In the loosely coupled approach, each sensor's measurements are processed separately and fused at the top. Thus, the resulting estimation optimizes only measurements of one among multiple sensors. The authors of [18] propose loosely coupling of several localization sources. The first step of the coupling is the sanity check, where localization failures are identified for each localization source using the vehicle dynamic model. Next, the non-failing localization sources are scored by the Chamfer score, and the final localization is selected as the best-scoring localization source. In [19], the authors used short-term IMU-centric evaluation of each localization source to select between Visual and LiDAR-based odometry. Although a loosely coupled approach can boost each algorithm and pick the best out of two sources of odometry estimates, the measurements of the two sensors do not complement each other to produce principally improved results, which is important in the degraded scenarios. For example, in a long corridor, the estimation in the longitudinal direction would suffer from a high error using LiDAR data due to the ambiguity. Hence, camera measurements are preferred.

In contrast, tightly coupled approaches output pose estimation by jointly optimizing measurements of different sensors. In [13], factor graph optimization is used for coarse-to-fine estimation. The authors propose to detect a poorly constrained direction of 6 Degrees of Freedom (DoF) (3D position with orientations) optimization space of Lie group SE(3) for each sensor (LiDAR, camera, and IMU). Although the final estimation eventually accounts for all the sensors used, the approach relies on the optimization-based degeneration detector that selects the specific sensor as a source of optimization for each of the SE(3) directions. Therefore, in the proposed approach, the measurements of all the sensors are accounted for in all directions, and corresponding weights are adjusted based on the contextual information.

A principally different approach is LIMO [20], which uses pure visual features with the depth assigned from LiDAR scans; however, it skips pure LiDAR features. Besides, semantic information is considered when weighing vegetation landmarks differently, as they might not be persistent in feature tracking. In LVI-SAM [4], 3D-LiDAR scans are used to measure the depth of visual features, but the final estimation is produced by the scan-matching with visual estimation as an initial guess. The authors of [21] propose to use large planar segments for optimization in combination with visual features. The approach benefits from the high-level semantic information of the LiDAR scan, but it is limited to structured environments and ignores other semantic entries that appear in LiDAR scans.

In existing LVI methods, sensory fusion parameters are determined empirically, which puts high demands on user knowledge of the particular localization system. Besides, parameter tuning can be laborious in achieving the best possible results for a particular method in a given environment. Black-box automatic hyperparameter tuning for odometry

systems is proposed in [14] using sequential model-based optimization. Nevertheless, our proposed method learns the parameters in the order that takes benefits from the parameters' interdependencies. Further, fusion weights can be determined by deep reinforcement learning [7]; however, only fixed weights are learned. Therefore, we propose to utilize dynamic weighting based on the contextual properties of the features, allowing the weights to adapt to the current situation.

III. Proposed LiDAR-Visual-Inertial Odometry

The proposed method is based on a tightly coupled sensor fusion of the 3D LiDAR, camera, and IMU measurements. The method structure is depicted in Fig. 2. A sequence of

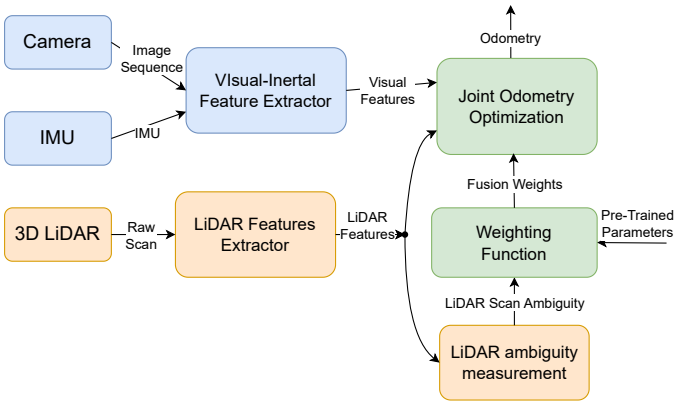


Fig. 2. The architecture of the proposed method.

the camera images synchronized with IMU measurements is processed by a visual-inertial feature detector and tracker, which outputs a set of current 3D features. These features are split into close and far sets based on their distance to the camera and fed into the optimization module. The 3D LiDAR scans are processed by the planar surface feature extractor and scan ambiguity detector. The LiDAR features are weighted based on the scene context and ambiguity. The weighted LiDAR features are fed into the optimization module to estimate the odometry from the joint weighted visual and LiDAR features. The output of the odometry estimation is the 6 DoF pose of the robot. The particular parts are detailed in the following sections.

A. Visual-Inertial Features

Visual-inertial feature extraction is inspired by VINS-Mono [9]. The features are detected using the Shi-Tomasi corner detector [22] and tracked using the Kanade-Lucas-Tomasi (KLT) tracker [23]. An example of the detected features is shown in Fig. 3.

For each feature tracked between consecutive frames, the 3D position is estimated using [9]. The tracked features are then used to estimate relative up-to-scale transformation between the frames and 3D feature poses using a five-point algorithm [24] and triangulation. The full bundle adjustment [25] is used to refine the estimated transformations.

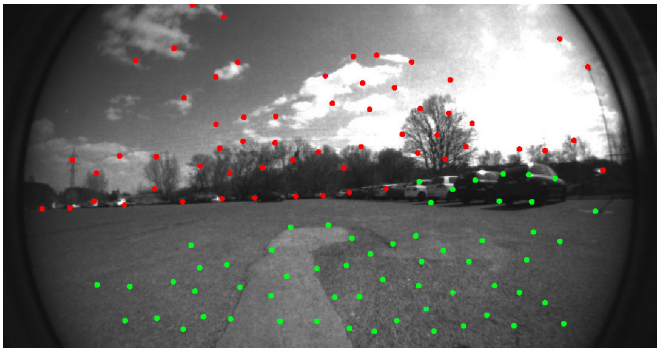


Fig. 3. Visual features in the fisheye camera image used in experimental evaluation of the proposed method. Green: close features, Red: far features, split with the threshold $\theta_{\text{visual}} = 11$ m.

Then, the up-to-scale transformations and 3D feature positions are jointly optimized with the pre-integrated IMU measurements using constrained graph optimization to fix the scale. As a result, 3D feature positions with respect to (w.r.t.) the pose of the robot are obtained for a camera frame received at the time instant t as

$$\mathbf{P}^t_{\text{vis}} = \{\mathbf{p}^t_i; i = 0, 1, \ldots, n; \mathbf{p}^t_i \in \mathbb{R}^3\}.$$

The localized 3D visual features are split into close features $\mathbf{P}^t_{\text{close}}$ and far features $\mathbf{P}^t_{\text{far}}$ based on the distance from the camera as

$$\begin{aligned} \mathbf{P}^t_{\text{close}} &= \{\mathbf{p}^t_{close,i} \in \mathbf{P}^t_{\text{vis}} \mid \left\|\mathbf{p}^t_{close,i}\right\| < \theta_{\text{visual}}\} \\ \mathbf{P}^t_{\text{far}} &= \{\mathbf{p}^t_{far,i} \in \mathbf{P}^t_{\text{vis}} \mid \left\|\mathbf{p}^t_{far,i}\right\| \geq \theta_{\text{visual}}\}, \end{aligned} \tag{1}$$

where θ_{visual} is the distinguishing threshold for the feature distance from the robot. It is assumed that close features can be localized more precisely so that they can be used for the estimation of the translation and rotation between consecutive frames. For far features, the feature's distance from the camera is estimated less precisely; the far features are used only for estimating orientation changes. The close and far features can be seen in Fig. 3.

B. LiDAR Features

The planar surface LiDAR features are extracted from the point cloud using [11] for each consequent LiDAR scan. The set of features extracted at the time t is denoted as

$$\mathbf{P}^t_{\text{LiDAR}} = \{\mathbf{p}^t_{\text{LiDAR},i}; i = 0, 1, \ldots, m; \mathbf{p}^t_{\text{LiDAR},i} \in \mathbb{R}^3\}.$$

An example of the planar features is depicted in Fig. 4.

The contribution of the features to the estimation of the robot's odometry is weighted based on the geometrical properties of the robot's surroundings. For example, when the robot enters an open area, LiDAR features are treated as less valuable than visual features since LiDAR features provide less information about the robot's motion. We introduce the ambiguity factor $\mathcal{A}$ that reflects the asymmetry of the distribution of the LiDAR features in 3D space to evaluate the properties of the scene. It is computed as a ratio of the smallest eigenvalue $\lambda_{\min}$ of the covariance matrix C_P,

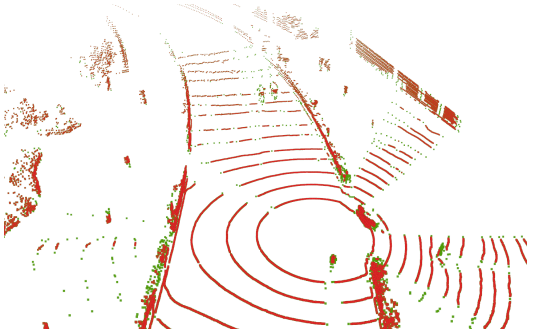


Fig. 4. LiDAR planar features cloud (red) and the rest of the points of the original scan (green).

and the largest eigenvalue $\lambda_{\max}$, where C_P is based on the distribution of the LiDAR features

$$\begin{aligned} \mathcal{A} &= \frac{\lambda_{\min}}{\lambda_{\max}} \\ C_P &= \frac{1}{N_{\text{LiDAR}}} \sum_{i=1}^{N_{\text{LiDAR}}} \mathbf{p}^t_{\text{LiDAR},i} \cdot (\mathbf{p}^t_{\text{LiDAR},i})^T \end{aligned}. \tag{2}$$

Based on the $\mathcal{A}$ value, we can distinguish three cases: *degenerated planar* case, *well-defined* case, and *transitional* case. Then, LiDAR features can be weighted differently depending on the case. It is because, in a well-defined scenario, the LiDAR features are more reliable than the visual ones, thus having a maximum weight $w^{\text{max}}_{\text{LiDAR}}$. However, in the case when LiDAR features form a plane, the LiDAR constraints noise in poorly defined directions can affect the better-defined visual optimization constraints. Thus, the LiDAR features are down-weighted to a smaller weight $w^{\text{min}}_{\text{LiDAR}}$, but still present in order to hold the planar constraint. In the transition between the well-defined and degenerated cases, the LiDAR features are gradually down-weighted using linear interpolation. Specifically, the cases are defined by comparing the degeneration factor $\mathcal{A}$ to the thresholds $\mathcal{A}_{\min}$ and $\mathcal{A}_{\max}$ as follows.

$$w_{\text{LiDAR}} = \begin{cases} w^{\text{min}}_{\text{LiDAR}} & \text{if } \mathcal{A} < \mathcal{A}_{\min} \text{ (degenerated)} \\ w^{\text{max}}_{\text{LiDAR}} & \text{if } \mathcal{A} > \mathcal{A}_{\max} \text{ (well-defined)} \\ \text{lerp}(\mathcal{A}) & \text{if } \mathcal{A}_{\min} < \mathcal{A} < \mathcal{A}_{\max} \text{ (transitional)} \end{cases} \tag{3}$$

where linear interpolation $lerp$ is defined by

$$lerp(\mathcal{A}) = w^{\text{min}}_{\text{LiDAR}} + \frac{\mathcal{A} - \mathcal{A}_{\min}}{\mathcal{A}_{\max} - \mathcal{A}_{\min}} \cdot (w^{\text{max}}_{\text{LiDAR}} - w^{\text{min}}_{\text{LiDAR}}). \tag{4}$$

The proposed ambiguity factor primarily detects open areas, where localization using only LiDAR-based features would be the most inaccurate.

C. Joint Optimization

Based on the three sets of the extracted features: $\mathbf{P}^t_{\text{close}}$, $\mathbf{P}^t_{\text{far}}$, and $\mathbf{P}^t_{\text{LiDAR}}$, we compute the displacement of the robot pose $\Delta\xi^t$ between the time instants $t-1$ and t of the consecutive frames by the following optimization. The optimization (5) minimizes the weighted sum of the cost functions J_{close}, J_{far}, and J_{LiDAR} that correspond to close visual, far visual, and LiDAR features, respectively.

$$\begin{aligned} \Delta\xi^t &= \arg\min_{\Delta\xi} J(\Delta\xi) \\ J(\Delta\xi) &= w_{\text{close}} J_{\text{close}} + w_{\text{far}} J_{\text{far}} + w_{\text{LiDAR}} J_{\text{LiDAR}} \end{aligned} \tag{5}$$

The optimization objective for the close visual features J_{close} is the minimization of the Euclidean distances between the observed landmarks $p^t_{\text{close, i}}$ and the corresponding landmarks in the previous frame $p^{t-1}_{\text{close, i}}$ as

$$J_{\text{close}}(\Delta\xi) = \sum_i \left\| \Delta\xi \cdot p^t_{\text{close, i}} - p^{t-1}_{\text{close, i}} \right\|_2, \tag{6}$$

used to optimize the whole transformation formed from both rotational and translational parts.

The optimization objective for the far visual features J_{far} is the minimization of the point-to-line errors e^i_{far} between the observed landmarks $p^t_{\text{far, i}}$ and the 3D lines defined by the robot's position $\vec{0}$ and the 3D positions of the corresponding landmarks in the previous frame $p^{t-1}_{\text{far, i}}$, expressed as

$$\begin{aligned} J_{\text{far}} &= \sum_i e^i_{\text{far}}(\Delta\xi) \\ e^i_{\text{far}}(\Delta\xi) &= e_{\text{point-to-line}}(\Delta\xi \cdot p^t_{\text{far, i}}, p^{t-1}_{\text{far, i}}, \vec{0}). \end{aligned} \tag{7}$$

Since far visual features have high uncertainty in depth, the point-to-line error $e_{\text{point-to-line}}$ is used instead of the Euclidean distance. The translation part of the objective's Jacobian is manually set to zero to avoid optimizing the translation part of the transformation.

The optimization objective for the LiDAR features $J_{\text{LiDAR}}(\Delta\xi)$ is the minimization of the point-to-plane errors $e^i_{\text{LiDAR}}(\Delta\xi)$ between the observed LiDAR features $p^t_{\text{LiDAR, }i}$ and the 3D planes $n^{t-1}_{\text{LiDAR, }i}$ defined by three closest point surface features in the LiDAR map

$$\begin{aligned} J_{\text{LiDAR}}(\Delta\xi) &= \sum_i e^i_{\text{LiDAR}}(\Delta\xi) \\ e^i_{\text{LiDAR}}(\Delta\xi) &= e_{\text{point-to-plane}}(\Delta\xi \cdot p^t_{\text{LiDAR, }i}, n^{t-1}_{\text{LiDAR, }i}). \end{aligned} \tag{8}$$

In (5), the LiDAR optimization objective J_{LiDAR} is weighted by the LiDAR features weight w_{LiDAR}, which is computed based on the ambiguity factor of the LiDAR features cloud $\mathcal{A}$.

The minimization according to (5) is performed using Ceres Solver [26], searching for a 6 DoF transformation describing the robot's motion between the time instants $t-1$ and t. The Jacobians of the error functions are calculated analytically for the sake of efficiency.

D. Parameters Learning

The parameters $\{\theta_{\text{visual}}, w_{\text{far}}, w_{\text{close}}, w^{\text{min}}_{\text{LiDAR}}, w^{\text{max}}_{\text{LiDAR}}, \mathcal{A}_{\min}, \text{and } \mathcal{A}_{\max}\}$ are learned using hyperparameter optimizer Optuna [27] with the experimentally obtained sensory data from the LiDAR, camera, and IMU further accompanied with the ground truth trajectory of the robot.

We propose to optimize the parameters in the order: θ_{visual}, w_{close}, w_{far}, $\mathcal{A}_{\min}$, $\mathcal{A}_{\max}$, $w^{\text{min}}_{\text{LiDAR}}$, $w^{\text{max}}_{\text{LiDAR}}$. The motivation is first to optimize the parameter that is used in splitting

the visual features θ_{visual}. When the features are classified, the importance of each feature type: w_{far}, w_{close} is optimized. Further, the ambiguity thresholds $\mathcal{A}_{\min}$ and $\mathcal{A}_{\max}$ are optimized to distinguish between the well-defined and ambiguous LiDAR features. Finally, the LiDAR weights values are optimized: $w^{\min}_{\text{LiDAR}}$, $w^{\max}_{\text{LiDAR}}$.

We propose to use a specific set of initial values that can be used for an arbitrary environment. Features importance is initialized to $w_{\text{far}} = w_{\text{close}} = w^{\max}_{\text{LiDAR}} = 1$. The visual distance threshold θ_{visual} can be initially set to a suitable value; however, it is not crucial as the optimization process finds it. The thresholds for the LiDAR features $\mathcal{A}_{\min}$ and $\mathcal{A}_{\max}$ are initialized to 0.0 so that every scan is weighted as well-defined (with $w^{\max}_{\text{LiDAR}}$) during the optimization of the preceding parameters. The initial value of $w^{\min}_{\text{LiDAR}}$ is set to a relatively small value of 0.5, which is important to be able to optimize the ambiguity threshold. The parameters learning is summarized in Algorithm 1.

Algorithm 1: Parameter Learning

Input: P_0 – Ordered parameters set with initial values, D_{train} – Training data.
Parameters: E_{RPE} – Estimation function.
Output: P – Optimizated parameters.

RPE $\leftarrow$ E$_{RPE}(P_0, D_{\text{train}})$
$P \leftarrow P_0$
for $p \in P$ **do**
  $p_{\text{opt}} \leftarrow \text{OPTIMIZE\_PARAM}(p, P, D_{\text{train}}, E_{RPE})$
  $P[p] \leftarrow p_{\text{opt}}$
  RPE$_{\text{current}} \leftarrow$ E$_{RPE}(P, D_{\text{train}})$
  if *RPE$_{current}$* $<$ *RPE* **then**
    RPE $\leftarrow$ RPE$_{\text{current}}$
  else
    $P[p] \leftarrow P_0[p]$
return P

The parameters are learned by minimizing the Relative Pose Error (RPE) [28] between the ground truth and the estimated poses, defined by the equation

$$\mathbf{E}_i = (\mathbf{Q}_i^{-1}\mathbf{Q}_{i+\Delta})^{-1}(\mathbf{P}_i^{-1}\mathbf{P}_{i+\Delta}) \tag{9}$$

where $\mathbf{Q}$ is SE(3) pose of the ground truth trajectory and $\mathbf{P}$ is SE(3) pose of the trajectory estimate. RPE is a measure that quantifies the local consistency of trajectory, which is suitable for the estimation of a small portion of the robot trajectory. We set Δ to $1\,\text{m}$ to be able to train the parameters on a small portion of the trajectory.

For minimization, the hyperparameter optimizer Optuna [27] is used, which was initially designed for the optimization of the hyperparameters of the neural networks. The optimizer is based on the Tree-structured Parzen Estimator (TPE) algorithm, which fits the Gaussian Mixture Model (GMM) to the objective function and samples the next value of hyperparameter from the GMM.

The Absolute Trajectory Error (ATE) [29] is used for the test evaluation, as it quantifies the global consistency of the trajectory, defined by the equation

$$\mathbf{F}_i = \mathbf{Q}_i^{-1}\mathbf{S}\mathbf{P}_i, \tag{10}$$

where $\mathbf{S}$ represents the alignment of the trajectory estimate and the ground truth. ATE is also used to compare the final performance of the selected methods. Note that since the total station outputs are only 3D poses of the robot without orientation, the orientation is set identically to the trajectory estimate. Thus, during the ATE evaluation, the orientation error is not used; instead, we use the average of $\mathbf{F}_i$ translation as the statistical performance indicator. For both ATE and RPE, the final measure is computed as the RMSE (Root Mean Square Error) of the translational parts of the errors along the trajectory.

IV. Results

The proposed method has been experimentally evaluated in two datasets. The first is the publicly available M2DGR [30] dataset, captured in an urban environment with many obstacles and geometrical structures at various distances from a vehicle. The second dataset is our Custom dataset recorded in a large open field area, with obstacles only at the area's borders to challenge the proposed method in a scenario that contains ambiguous LiDAR scans. Both datasets are split into a training part and a testing part. The training part is used to learn the parameters of the proposed method, and the testing part is used to evaluate the odometry estimation. The performance of the proposed methods is compared with three state-of-the-art approaches.

In particular, we opted for F-LOAM [11] as a representant of the LiDAR-based method, which inspired the employed optimization. The vision-based method is represented by VINS-Mono [9], from which the visual feature extraction is utilized in the proposed method. Finally, LIO-SAM [5] is selected for a comparison of the LiDAR-Inertial method, and LVI-SAM [4] is included in comparison for the M2DGR dataset as an existing LiDAR-Visual-Inertial method.

TABLE I
Learned parameters for both datasets

Parameter	M2DGR Dataset	Custom Dataset
θ_{visual} [m]	10.0	11.0
w_{close}	0.0	0.4
w_{far}	0.0	0.2
$\log(\mathcal{A}_{\min})$	20.0	8.3
$\log(\mathcal{A}_{\max})$	20.0	11.0
$w^{\min}_{\text{LiDAR}}$	0.5	0.2
$w^{\max}_{\text{LiDAR}}$	0.5	0.5

ATE and RPE [29] are used as the performance indicators. The learned parameters of the proposed method are depicted in Table I. The results achieved in the M2DGR and Custom datasets are reported in the following sections.

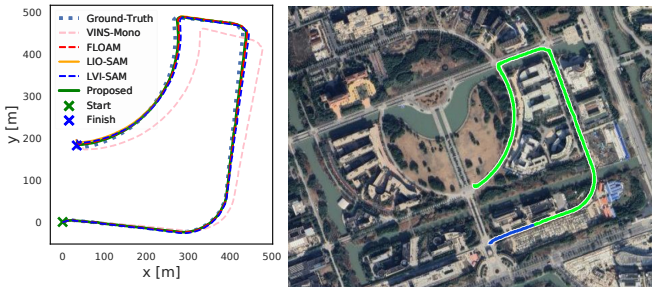
A. M2DGR Dataset

For the M2DGR dataset, we opted for the longest urban sequence `street02` with the length of $1.2\,\text{km}$. It is recorded for the Velodyne VLP-32C LiDAR running

TABLE II

PERFORMANCE INDICATORS IN M2DGR DATASET

Method	ATE [m]	RPE [m]
F-LOAM [11]	2.82	0.07
VINS-Mono [9]	24.16	0.25
LIO-SAM [5]	3.60	**0.05**
LVI-SAM [4]	3.75	0.07
Proposed	**2.72**	0.06



(a) Trajectories. (b) Bird's-eye view

Fig. 5. Trajectories and bird's-eye view of the M2DGR scenario, where the training set is in blue and the test set is in green.

at 10 Hz, a FLIR Pointgrey RGB camera running at 10 Hz, and 9-axis Handsfree A9 IMU running at 150 Hz. For the training set, the first 0.2 km long part of the sequence is used; thus, the evaluation is performed for the remaining 1.0 km, as illustrated in Fig. 5. The achieved ATE and RPE are listed in Table II, and projected position estimates are depicted in Fig. 5.

The results in Table II suggest the proposed method performs similarly to F-LOAM and LIO-SAM. The LiDAR-based methods perform significantly better than VINS-Mono since LiDAR measurements are precise and thus useful for localization given the absence of alignment in ambiguous areas.

LVI-SAM, which is an enhanced version of LIO-SAM by fusion with a variation of VINS-Mono, performs worse than LIO-SAM in the scenario. In contrast, as can be observed in Table I, the usage of visual features is discarded for the proposed method (parameters w_{close} and w_{far} are set to 0). The results support the benefit of the learning procedure to determine that the used visual features do not improve the performance in such a well-structured environment as the `street02` sequence.

B. Custom Dataset

The custom dataset is tailored to include locations that are challenging for the LiDAR-based methods. The dataset has been collected using four-wheeled robot Clearpath Husky, see Fig. 6a, with the LiDAR Ouster OS0-128 running at 10 Hz, single camera of the fisheye stereo pair of the Intel® RealSense™ Tracking Camera T265 (T265) running at 30 Hz, and Xsens MTi-30 IMU running at 200 Hz. All the sensors have been extrinsically calibrated w.r.t. the coordinate system of the LiDAR.

The dataset has been collected in an open field of a parking lot for two loops, see Fig. 6, with the total length of 160 m. The first 30 m long part is used for training and the rest

(a) Used Clearpath Husky (b) Bird's-eye view on a parking lot

Fig. 6. Used robot for Custom dataset collection and overview of the deployment location with visualization of the data collection path with highlighted training part in blue, testing part in green, and unused part of the trajectory in gray.

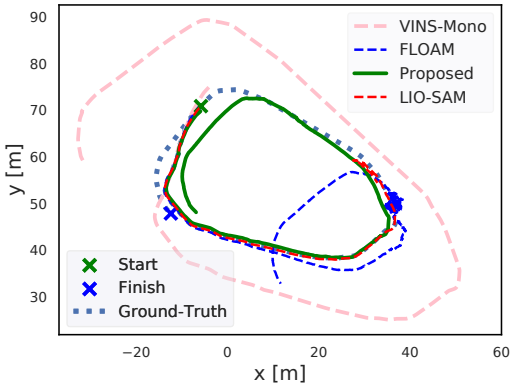


Fig. 7. Projected position estimates in Custom scenario.

for testing. The ground truth robot 3D position is captured by the Leica TS16 total station with centimeter precision. The performance indicators are depicted in Table III, and the estimated trajectories are in Fig. 7.

TABLE III

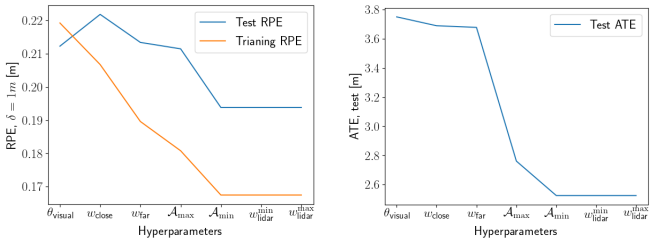
PERFORMANCE INDICATORS IN CUSTOM SCENARIO

Method	ATE [m]	RPE [m]
F-LOAM [11]	13.00	0.37
VINS-Mono [9]	13.50	0.39
LIO-SAM [5]	`Fail`	`Fail`
Proposed	**2.62**	**0.18**

`Fail` indicates the method has not been able to produce reasonable results.

The results show that LiDAR-based methods LIO-SAM and FLOAM performed relatively well until the robot reached the open area and degenerated for the LiDAR scan alignment. Then, both LiDAR-based methods failed to localize the robot further due to insufficient information from LiDAR scans. Specifically, LIO-SAM suffered from growing position jumps, and F-LOAM tended to stay around the last known position. VINS-Mono managed to estimate the robot's position for the whole path, including the open area, but with a noticeable drift. The best results are achieved for the proposed method that exploits the advantages of the LiDAR-based and visual features when needed.

The training progress of the learned parameters is further visualized in Fig. 8 as a sequence of training the particular hyperparameters in the order of $\theta_{\text{visual}}, w_{\text{close}}, w_{\text{far}}, \mathcal{A}_{\max}, \mathcal{A}_{\min}, w^{\min}_{\text{LiDAR}}, w^{\max}_{\text{LiDAR}}$. A significant



(a) Relative pose error (b) Absolute trajectory error

Fig. 8. Progress for training particular hyperparameters in the defined order on the Custom dataset. The training performance is in orange, and the testing is in blue. Training ATE is not reported as it is not being used.

improvement of the test ATE can be noticed after training the thresholds for the LiDAR ambiguity $\mathcal{A}_{\max}$ and $\mathcal{A}_{\min}$. It supports the proposed idea that accounting for the structure of the environment is an important part of odometry estimation. Moreover, training parameters based on RPE on the training set also lead to a decrease in ATE on the test set, which indicates that the learning procedure generalizes well between the sets.

V. Conclusion

We propose a tightly-coupled LiDAR-Visual-Inertial Odometry system that adapts the weights for the multi-modal sensor fusion based on the ambiguity measure of LiDAR scans. The developed localization system is parametrized with the distance thresholds for the visual features and quality of the LiDAR scan. The parameters are iteratively learned one by one using real data with the ground truth that shows to improve system performance in the target environment. The performance of the proposed method is evaluated in the M2DGR dataset and Custom dataset, which includes degenerated for LiDAR scans alignment open area environment. The results support the proposed ambiguity factor successfully recognizing open areas. In future work, we plan to focus on ambiguity factors in indoor environments to address monotonous long corridors.

References

[1] W. Burgard, D. Fox, and S. Thrun, "Active mobile robot localization," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 1997, pp. 1346–1352.

[2] S. Thrun, "Probabilistic robotics," *Communications of the ACM*, vol. 45, no. 3, pp. 52–57, 2002.

[3] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 3354–3361.

[4] T. Shan, B. Englot, C. Ratti, and D. Rus, "LVI-SAM: tightly-coupled lidar-visual-inertial odometry via smoothing and mapping," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 5692–5698.

[5] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "LIO-SAM: tightly-coupled lidar inertial odometry via smoothing and mapping," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 5135–5142.

[6] J. Zhang and S. Singh, "Laser–visual–inertial odometry and mapping with high robustness and low drift," *Journal of Field Robotics*, vol. 35, no. 8, pp. 1242–1264, 2018.

[7] Y. Jia, H. Luo, F. Zhao, G. Jiang, Y. Li, J. Yan, Z. Jiang, and Z. Wang, "Lvio-fusion: A self-adaptive multi-sensor fusion slam framework using actor-critic method," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 286–293.

[8] H. Wang, C. Wang, C.-L. Chen, and L. Xie, "F-loam: Fast lidar odometry and mapping," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 4390–4396.

[9] 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.

[10] "The KITTI vision benchmark suite: Visual odometry," cited on 2024-03-14. [Online]. Available: https://www.cvlibs.net/datasets/kitti/eval_odometry.php

[11] J. Zhang and S. Singh, "LOAM: lidar odometry and mapping in real-time." in *Robotics: Science and Systems*, 2014, pp. 1–9.

[12] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 4758–4765.

[13] J. Zhang and S. Singh, "Enabling aggressive motion estimation at low-drift and accurate mapping in real-time," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 5051–5058.

[14] K. Koide, M. Yokozuka, S. Oishi, and A. Banno, "Automatic hyper-parameter tuning for black-box lidar odometry," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 5069–5074.

[15] C. Debeunne and D. Vivet, "A review of visual-lidar fusion based simultaneous localization and mapping," *Sensors*, vol. 20, no. 7, p. 2068, 2020.

[16] 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.

[17] X. Xu, L. Zhang, J. Yang, C. Cao, W. Wang, Y. Ran, Z. Tan, and M. Luo, "A review of multi-sensor fusion slam systems based on 3d lidar," *Remote Sensing*, vol. 14, no. 12, p. 2835, 2022.

[18] A. Reinke, X. Chen, and C. Stachniss, "Simple but effective redundant odometry for autonomous vehicles," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 9631–9637.

[19] V. Hulchuk, J. Bayer, and J. Faigl, "Graph-based lidar-inertial slam enhanced by loosely-coupled visual odometry," in *European Conference on Mobile Robots (ECMR)*, 2023, pp. 1–8.

[20] J. Graeter, A. Wilczynski, and M. Lauer, "Limo: Lidar-monocular visual odometry," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 7872–7879.

[21] D. Wisth, M. Camurri, S. Das, and M. Fallon, "Unified multi-modal landmark tracking for tightly coupled lidar-visual-inertial odometry," *Robotics and Automation Letters*, vol. 6, no. 2, pp. 1004–1011, 2021.

[22] J. Shi *et al.*, "Good features to track," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1994, pp. 593–600.

[23] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *International Joint Conference on Artificial Intelligence (IJCAI)*, vol. 2, 1981, pp. 674–679.

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

[25] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—a modern synthesis," in *1999 Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms*. Springer, 2000, pp. 298–372.

[26] S. Agarwal, K. Mierle, and T. C. S. Team, *Ceres Solver*, 10 2023, cited on 2024-03-14. [Online]. Available: https://github.com/ceres-solver/ceres-solver

[27] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 2623–2631.

[28] J. Sturm, S. Magnenat, N. Engelhard, F. Pomerleau, F. Colas, D. Cremers, R. Siegwart, and W. Burgard, "Towards a benchmark for rgb-d slam evaluation," in *Rgb-d workshop on advanced reasoning with depth cameras at Robotics: Science and Systems (RSS)*, 2011.

[29] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 573–580.

[30] J. Yin, A. Li, T. Li, W. Yu, and D. Zou, "M2dgr: A multi-sensor and multi-scenario slam dataset for ground robots," *Robotics and Automation Letters*, vol. 7, no. 2, pp. 2266–2273, 2021.