drones

MDPI

Article

R-LVIO: Resilient LiDAR-Visual-Inertial Odometry for UAVs in GNSS-denied Environment

Bing Zhang, Xiangyu Shao *, Yankun Wang, Guanghui Sun and Weiran Yao

Department of Control Science and Engineering, Harbin Institute of Technology, Harbin 150001, China; zhangbing0920@stu.hit.edu.cn (B.Z.); 20B904057@stu.hit.edu.cn (Y.W.); guanghuisun@hit.edu.cn (G.S.); yaoweiran@hit.edu.cn (W.Y.)
* Correspondence: xiangyushao@hit.edu.cn

Abstract: In low-altitude, GNSS-denied scenarios, Unmanned aerial vehicles (UAVs) rely on sensor fusion for self-localization. This article presents a resilient multi-sensor fusion localization system that integrates light detection and ranging (LiDAR), cameras, and inertial measurement units (IMUs) to achieve state estimation for UAVs. To address challenging environments, especially unstructured ones, IMU predictions are used to compensate for pose estimation in the visual and LiDAR components. Specifically, the accuracy of IMU predictions is enhanced by increasing the correction frequency of IMU bias through data integration from the LiDAR and visual modules. To reduce the impact of random errors and measurement noise in LiDAR points on visual depth measurement, cross-validation of visual feature depth is performed using reprojection error to eliminate outliers. Additionally, a structure monitor is introduced to switch operation modes in hybrid point cloud registration, ensuring accurate state estimation in both structured and unstructured environments. In unstructured scenes, a geometric primitive capable of representing irregular planes is employed for point-to-surface registration, along with a novel pose-solving method to estimate the UAV's pose. Both private and public datasets collected by UAVs validate the proposed system, proving that it outperforms state-of-the-art algorithms by at least 12.6%.

Keywords: Multi-sensor fusion; LiDAR-visual-inertial odometry; structure quantification; point-to-surface alignment

check for updates

Citation: Zhang, B.; Shao, X.; Wang, Y.; Sun, G.; Yao, W. R-LVIO: Resilient LiDAR-Visual-Inertial Odometry for UAVs in GNSS-denied Environment. *Drones* **2024**, *8*, 487. https://doi.org/10.3390/drones8090487

Academic Editor: Francesco Nex

Received: 17 August 2024
Revised: 8 September 2024
Accepted: 12 September 2024
Published: 14 September 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Unmanned aerial vehicle (UAV) navigation and localization systems must be stable and accurate to complete various tasks. UAVs generally adopt the global navigation satellite system (GNSS) as the core navigation technology. However, the open frequency bands of GNSS satellite signals make them susceptible to interference, which can result in UAVs failing to complete their planned missions or even return to base [1–4]. Multi-sensor fusion-based localization techniques are essential for UAVs to achieve state estimation in low-altitude, GNSS-denied scenarios. In multi-sensor fusion frameworks, UAVs are typically equipped with light detection and ranging (LiDAR), cameras, and inertial measurement units (IMUs) to collect multi-source information, achieving 6-degree-of-freedom (DoF) state estimation [5,6].

In recent years, numerous UAV sensor fusion frameworks based on LiDAR, cameras, and IMUs have been presented to achieve superior estimation accuracy in artificial environments, such as streets, campuses, and factories [7,8]. In these frameworks, LiDAR is usually used as the core sensor due to its high-fidelity measurements and wide-range sensing capabilities [9,10]. However, in unstructured environments like corridors, deserts, and stadiums, the lack of LiDAR return points leads to insufficient constraints for pose estimation. The challenges encountered in unstructured environments include the following. (1) Irregular planes are difficult to represent accurately. Ground points in unstructured scenes make up a relatively large portion of the LiDAR point cloud. The ground points can be fitted to similar planes due to minimal height differences. However, using planes as primitives results in the loss of the ground's uneven geometric properties. (2) Vertical constraints are inadequate for UAV height measurement. Unstructured scenes have too few LiDAR return points in the vertical direction, making the LiDAR module insensitive to altitude changes.

To tackle the problems in unstructured scenarios, we propose the following measures. (1) Select appropriate geometric primitives to represent irregular surfaces and adjust the point cloud alignment model. (2) Use short-term prediction of the IMU vertical direction instead of using the LiDAR module for height measurement. (3) Employ the IMU module as the core of the system to output the final state, thereby mitigating the risk of failure in the LiDAR or visual modules. This paper proposes a multi-sensor fusion-based odometry and mapping framework that relies on the complementary advantages of LiDAR, cameras, and IMUs, achieving low-drift and high robustness state estimation. The main contributions of the proposed system are as follows:

- Improve the accuracy of short-term IMU predictions by increasing the frequency of corrections from the LiDAR and visual modules. LiDAR pose frequency is boosted by sweep segmentation to synchronize the LiDAR input time with the camera sampling time.
- Devise an outliers rejection strategy of depth association between the camera image and LiDAR points to select accurate depth points by evaluating the reprojection error of visual feature points in a sliding window.
- Design a structure monitor to distinguish structured scenes and unstructured scenes by analyzing the vertical landmarks. The environmental structuring is quantified to switch the operating modes of the LiDAR module.
- Propose a novel point-to-surface model to register irregular surfaces in unstructured scenes, achieving three horizontal DoF state estimation. The vertical 3-DoF state is predicted by IMU relative measurement.

The paper is structured as follows. The related work of sensor fusion localization and point cloud registration is presented in Section 2. The problem statement is presented in Section 3. The details of the proposed system are illustrated in Section 4. Experimental results on public and private datasets are provided in Section 5. Conclusions and future work are given in Section 6. The notations are listed in Table 1.

Table 1. List of used notations with their descriptions.

Notations	Descriptions
t_i, t_j	Input time of camera image and LiDAR sweep
$\mathscr{X}$	Set of all states up to moment $t_n \in \mathscr{T}_n$
$\boldsymbol{x}_i$	State at time t_i
$\mathbf{R}_i$, $\mathbf{p}_i$ and $\mathbf{v}_i$	Rotation matrix, position vector, and linear velocity at time t_i
$\mathbb{C}_i$, $\mathbb{L}_i$	Observations of camera and LiDAR at time t_i
$\mathbb{I}_ij$	Set of IMU measurements between moments t_i and t_j
r_0, $r_{\mathbb{I}_ij}$, $r_{z_{ic}}$, $r_{z_{il}}$	Residuals of prior, IMU preintegration, and visual and LiDAR feature
$\hat{\boldsymbol{a}}_k^B$, $\hat{\boldsymbol{w}}_k^B$	Measurements value of accelerometer and gyroscope at time t_k
$\boldsymbol{a}_k^B$, $\boldsymbol{w}_k^B$	Acceleration and angular velocity of platform motion
$\boldsymbol{g}$	gravity
$\boldsymbol{b}_{a_k}$, $\boldsymbol{b}_{w_k}$, $\boldsymbol{b}_a$, $\boldsymbol{b}_w$	Biases and noise of accelerometer and gyroscope
$\Delta\bar{\mathbf{R}}_{ij}$, $\Delta\bar{\mathbf{p}}_{ij}$, $\Delta\bar{\mathbf{v}}_{ij}$	Preintegrated measurements for orientation, translation, and velocity
$r_{\Delta\mathbf{R}_{ij}}^\top$, $r_{\Delta\mathbf{v}_{ij}}^\top$, $r_{\Delta\mathbf{p}_{ij}}^\top$	Preintegration error
$\Delta\mathbf{R}_{ij}^O$, $\Delta\mathbf{p}_{ij}^O$, $\Delta\mathbf{v}_{ij}^O$	Prediction states from IMU
E_{ij}^C, E_{ij}^L	IMU measurement errors with VIO and LIO
Ω_{ij}^C, Ω_{ij}^L	Uncertain matrix of LiDAR and camera poses
E_m	Error of marginalized prior

Table 1. *Cont.*

Notations	Descriptions
e_i^{rp}	Reprojection residual error of the visual feature z_{ic}
$\mathscr{C}_{po}$	Constraint from the back-propagated LiDAR pose
$\mathbb{T}^C$	Set of camera poses
$\mathcal{S}(\boldsymbol{p}_i)$	Smoothness of the LiDAR point $\boldsymbol{p}_i$
$\\|\boldsymbol{r}_i\\|$	The range of the point $\boldsymbol{p}_i$.
$\mathbb{F}_n^e, \mathbb{F}_n^p$	Edge and plane points of n-th sweep
$\mathbb{F}_n$	All feature points of n-th sweep
$\mathscr{U}_n$	Clustering results of n-th sweep
$\boldsymbol{V}_i$	The weighted distance vector of the i-th sector
$\mathscr{C}_n$	Environmental structuring of n-th sweep
$\check{\mathbf{T}}_k^L$	Initial LiDAR pose from prediction state
$\mathcal{S}^{\mathbb{F}_k}$	Weighted point-to-feature distance
r_{pf}	Feature registration error
r_{pg}	Point-to-Gaussian surface error
$\boldsymbol{W}, \boldsymbol{N}, \Lambda$	Covariance matrix, eigenvector matrix, and diagonal matrix
$\mathbf{T}_n^L$	LiDAR pose at time t_n
$\mathbf{R}(\gamma_n)$	Rotation matrix on *yaw* angle
$\tilde{\boldsymbol{J}}, \tilde{\boldsymbol{H}}$	Jacobian matrix and Hessian matrix by differentiating the projected point $\tilde{\boldsymbol{f}}_j$ to the pose $\hat{\mathbf{T}}_n^L$

2. Related Works

2.1. Sensor Fusion Localization System

Recently, significant efforts have been made in the field of sensor fusion localization. Nguyen proposes VIRAL-fusion [11], which combines an IMU, an ultra wideband ranging sensor, and multiple on-board visual-inertial and LiDAR-ranging subsystems to implement an optimization-based comprehensive state estimator on UAVs. This estimator effectively mitigates the problem of pose position drift and robustness in low-texture environments. LIC-fusion2 [12] introduces a novel sliding-window planar feature tracking technique based on online spatio-temporal calibration for efficient processing of 3D LiDAR point clouds. A novel outlier rejection criterion is proposed in planar feature tracking to initialize feature points belonging to the same plane for high-quality data correlation. Shao [13] proposes a integrated LiDAR-visual-inertial framework that conducts coupling optimization in a factor graph manner, enabling a refined system state. Lin [14] proposes a novel multi-sensor fusion framework, called R3LIVE, that leverages the measurement strengths of LIDAR, inertial, and visual sensors to enable real-time reconstruction of accurate, dense, 3D, RGB-colored maps of the surrounding environment. Zheng [15] proposes a fast LiDAR-inertial-visual odometry system based on two tightly coupled direct subsystems: a VIO subsystem and a LIO subsystem. The LIO subsystem registers new sweep points onto an incrementally constructed point cloud map. The points on the map are also appended with image patches, which are then used in the VIO subsystem to align the new image by minimizing photometric errors. The system combines the advantages of sparse direct image alignment and raw point direct alignment to achieve accurate and reliable attitude estimation with low computational cost.

The aforementioned systems enable accurate state estimation in structured environments. However, in unstructured scenes, the LiDAR module with a point-to-plane model cannot build a consistent surrounding map, and the overall system perhaps fails due to LiDAR pose drift or map divergence. To tackle this problem, some researchers have utilized the property of multi-sensor fusion, which involves discarding the output of the LiDAR module and using other sensors' poses as the system state. Zhang [16] introduces LiDAR-visual-IMU odometry, which starts with IMU preintegration measurement and ends with refined poses in a visual-inertial subsystem and LiDAR-inertial subsystem. Although the system fails to greatly improve the state accuracy compared with the LiDAR subsystem, the robustness is enhanced due to the coupling of the visual subsystem as a complement to the state estimation. Wisth [17] proposes VILENS, a factor graph-based odometry system for legged robots. By tightly fusing LiDAR, cameras, IMUs, and leg odometry together, reliable operation is achieved despite the fact that individual sensors can produce degraded estimates. To minimize legged odometry drift, the system extends the robot's state using a linear velocity deviation term, which is estimated online by preintegrating measurement with the visual, LiDAR, and IMU factors. The system exhibits excellent localization performance and strong robustness in unstructured environments. Based on the above study, we propose a method to compensate for LiDAR drift by combining short-term IMU predictions with the pre-drift LiDAR pose for state estimation. The proposed system achieves synchronization between LiDAR and camera by segmenting the LiDAR sweep, which improves the accuracy of short-term IMU predictions by boosting the correction frequencies. In addition, the undegraded LiDAR module is also used to constrain the pose estimation of the visual module to improve the localization accuracy of the visual sub-module. In the case of LiDAR failure, the visual-inertial module is employed to generate the final state.

2.2. Point Cloud Registration

The majority of existing work accomplishes point cloud alignment by sweep matching, which involves using iterative closest point (ICP) and normal distribution transform (NDT) algorithms to solve sweep-to-sweep or sweep-to-map transformations [18–20]. Point-to-point, point-to-edge, point-to-plane, and point-to-probability model techniques are among the geometric primitives that are employed in it. By fully considering the sparsity and scene complexity of LiDAR point clouds, Cui [21] provides a linear keypoint representation for 3D LiDAR point clouds, which minimizes keypoint-to-keypoint distance to efficiently perform sweep-to-sweep alignment. Zhang [22] distinguishes plane and edge points by calculating the local smoothness, and then minimizes the point-to-plane distance and point-to-edge distance to achieve accurate point cloud alignment. On this foundation, Guo [23] extracts edge and plane features by the principal component analysis (PCA) method, which is employed in two-stage alignment to achieve, without loss of real-time performance, improved odometry accuracy and consistent mapping.

Regularized planes are commonly used as geometric primitives for matching in the above methods. However, uneven surfaces are widely found in unstructured and undeveloped environments. Still, utilizing a planar model for point cloud matching will produce large random errors. Choi [24] presents a fast and generalized feature-based LiDAR odometry method using local quadratic surface approximation and point-to-surface alignment. Unlike most matching methods based on point-to-plane distances, the method approximates the local geometry of the LiDAR scan as a quadratic surface to minimize performance degradation due to the inconsistency of feature classes with the local geometry of the map. Chen [25] presents a lightweight front-end LiDAR odometry solution that uses a point-to-point probabilistic model in a generalized ICP-based direct point cloud matching method to yield accurate state estimation in unstructured subterranean environments. Combining these two approaches, this paper approximates an uneven surface as a Gaussian surface, which is formulated as a Gaussian probability function consisting of neighboring points. The point-to-Gaussian surface distance is employed to point cloud matching, achieving low-drift LiDAR odometry in unstructured scenes.

3. Problem Statement

The state variables are represented by the following in the proposed system:

$$\mathcal{X}_n = \{\boldsymbol{x}_i\}_{i\in\mathcal{T}_n} = \{\mathbf{R}_i, \mathbf{p}_i, \mathbf{v}_i, \mathbf{b}_{w_i}, \mathbf{b}_{a_i}\}_{i\in\mathcal{T}_n}. \tag{1}$$

where $\mathcal{X}_n$ represents the set of all states. $\mathbf{R}_i$, $\mathbf{p}_i$, and $\mathbf{v}_i$ denote the orientation, the translation, and the motion velocity of the UAV at moment t_i, respectively. $\mathbf{b}_{w_i}$ and $\mathbf{b}_{a_i}$ are the IMU biases.

The camera observations at time t_i are represented as $\mathbb{C}_i$, which include the extracted feature point, $\boldsymbol{z}_{ic}$, from the images. The measurement data from the LiDAR are represented as $\mathbb{L}_i$, which include the salient points, z_{il}, to be matched. IMU measurements between adjacent camera sampling moments t_i and t_j are represented as $\mathbb{I}_{ij}$. The UAV state is estimated by minimizing the sum of squared observation residuals, as follows:

$$\begin{aligned}\mathcal{X}_n &= \arg\min_{\mathcal{X}_n}(-lnP(\mathcal{X}_n|\mathcal{Z}_n)) = \arg\min_{\mathcal{X}_n}\|r_0\|^2 \\ &+ \sum_{(i,j)\in\mathcal{T}_n}\left\|r_{\mathbb{I}_{ij}}\right\|^2 + \sum_{i\in\mathcal{T}_n}\sum_{c\in\mathbb{C}_i}\|r_{z_{ic}}\|^2 + \sum_{i\in\mathcal{T}_n}\sum_{l\in\mathbb{L}_i}\|r_{z_{il}}\|^2.\end{aligned} \tag{2}$$

where r_0 is the prior error. $r_{\mathbb{I}_{ij}}$, $r_{z_{ic}}$, $r_{z_{il}}$ represent the residuals of the associated measurements. Residuals are functions of the state variables and observations, quantifying the mismatch between the observations and the estimated values under the current state and prior constraints.

4. Proposed Method

4.1. System Overview

The goal of our proposed system is to estimate the UAV state and construct the surrounding map. The intrinsic parameters of these sensors are assumed to be known. The extrinsic parameters between the three sensors have been calibrated to share a common coordinate system, with the IMU frame designated as the primary coordinate system. The camera and LiDAR frames are considered sub-coordinate systems. The definitions of these coordinate systems are illustrated in Figure 1. According to the figure, *C*, *L*, and *B* represent the coordinate systems of the camera, LiDAR, and IMU, respectively.

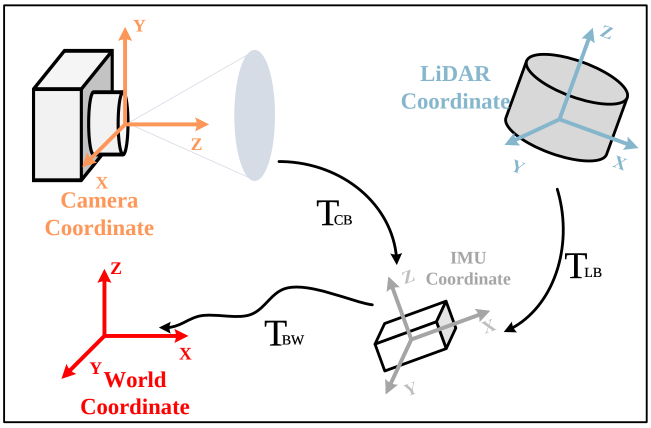


Figure 1. The schematic diagram of coordinate transformation. T_{CB} and T_{LB} represent the external parameter from camera and LiDAR to IMU. T_{BW} represents the transformation from the body frame to the world frame.

An overview of our system is illustrated in Figure 2. The IMU and visual modules provide prior poses and constraints to the LiDAR module, which is the main component of the system that uses motion estimation from coarse to fine. Initially, the camera input frequency reconstructs a LiDAR sweep to guarantee synchronized transmission between various sensors. Then, using cross-validation to assess the projection errors of visual feature-associated LiDAR points under various perspectives, the depth of visual features with large approximation errors and measurement noise is eliminated by outliers rejection. By utilizing temporal synchronization, the vision module receives the LiDAR pose from the previous instant, which helps to provide a position consistency constraint that enhances the accuracy of the camera pose estimation. To reduce the drift of the feature-based LiDAR module in scenes with fewer features, a direct point cloud registration method with IMU-constrained point-to-Gaussian surface error is proposed to be incorporated into the pose estimation.

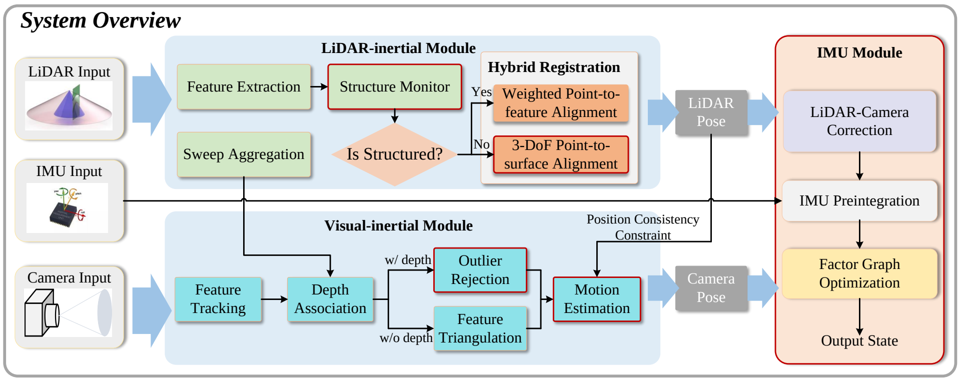


Figure 2. Pipeline of the proposed system. The proposed system is divided into the IMU module, the visual-inertial module, and the LiDAR-inertial module. Modules with red borders are highlighted in this paper. In detail, the LiDAR-inertial module provides depth measurements for visual features by aggregating recent multi-frame sweeps. Moreover, the motion estimation of the visual-inertial module is constrained by the back-propagated pose from the LiDAR-inertial module at the previous moment. The visual-inertial module provides the initial guess for the LiDAR-inertial module's point cloud matching. The camera pose and LiDAR pose are fed into the IMU module and form the measurement residuals with IMU preintegration, followed by minimizing the measurement residuals in the factor graph optimization to estimate the final state.

4.2. Imu Module with High Frequency Correction

In this module, the LiDAR point cloud and camera image are synchronized to implement the transmission between the LiDAR module and the visual module. IMU measurements between two consecutive frames are integrated to align with the LiDAR point cloud and camera image.

4.2.1. Time Synchronization Based on Sweep Segmentation

Temporal interpolation techniques are used in most existing research on sensor time synchronization and are effective for matching data between sensors with noticeably differing frequencies [26]. However, when using sensors with resembling input frequencies (e.g., LiDAR and camera), errors can arise because of the large gaps between their sample periods.

To solve the problem of information lag caused by different sampling rates among sensors, a LiDAR sweep reconstruction algorithm controlled by camera input frequency is adopted to avoid ambiguous transmission between the LiDAR module and the visual module. Specifically, image acquisition time serves as the starting point for reconstructing the LiDAR sweep, which is motivated by the continuous sampling nature of LiDAR. This synchronization allows for simultaneous processing of camera images and LiDAR sweeps, avoiding interpolation or approximation operations during LiDAR and visual information fusion. This not only facilitates the transfer of enhanced depth and backward propagated poses from LiDAR to the camera but also provides higher frequency corrections to the IMU bias to improve the accuracy of IMU short-term predictions.

4.2.2. IMU Kinetic Model

Let the timestamps of two consecutive frames, F_i and F_j, be denoted as t_i, t_j. The measurements of the accelerometer and gyroscope during the time interval are described as:

$$\hat{\boldsymbol{a}}_k^B = \boldsymbol{a}_k^B - \boldsymbol{g} + \boldsymbol{b}_{a_k} + \boldsymbol{n}_a, \quad \hat{\boldsymbol{w}}_k^B = \boldsymbol{w}_k^B + \boldsymbol{b}_{w_k} + \boldsymbol{n}_w. \tag{3}$$

where measurements between clock times t_i and t_j are denoted by the $k = 1, 2, \ldots, n$ index. Motion measurements include gravity, $\boldsymbol{g}$, motivation, $\boldsymbol{a}_k^B$, and angular velocity, $\boldsymbol{w}_k^B$. These measurements are interfered with biases $\boldsymbol{b}_{a_k}$, $\boldsymbol{b}_{w_k}$ and measurement noise $\boldsymbol{n}_a$, $\boldsymbol{n}_w$.

In this work, a discrete-time IMU preintegration method is employed to obtain the relative motion within the time interval $\left[t_i, t_j\right]$ [27]. The preintegrated measurements for orientation, $\Delta\bar{\mathbf{R}}_{ij}$, translation, $\Delta\bar{\mathbf{p}}_{ij}$, and velocity, $\Delta\bar{\mathbf{v}}_{ij}$, in the IMU frame are given by:

$$\begin{aligned}
\Delta\bar{\mathbf{R}}_{ij} &= \mathbf{R}_i^\top \mathbf{R}_j \doteq \prod_{k=i}^{j-1} \exp[(\hat{\boldsymbol{w}}_k - \boldsymbol{b}_{w_i})\delta t], \\
\Delta\bar{\mathbf{v}}_{ij} &= \mathbf{R}_i^\top (\mathbf{v}_k - \mathbf{v}_i - \boldsymbol{g}\delta t) \doteq \sum_{k=i}^{j-1} \Delta\mathbf{R}_{ik}(\hat{\boldsymbol{a}}_k - \boldsymbol{b}_{a_i})\delta t, \\
\Delta\bar{\mathbf{p}}_{ij} &= \mathbf{R}_i^\top \left(\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i\delta t - \frac{1}{2}g\delta t^2\right) \doteq \sum_{k=i}^{j-1} \left(\Delta\mathbf{v}_{ik}\delta t + \frac{1}{2}\Delta\mathbf{R}_{ik}(\hat{\boldsymbol{a}}_k - \boldsymbol{b}_{a_i})\delta t^2\right).
\end{aligned} \tag{4}$$

where δt is the time interval between adjacent IMU measurements. The preintegration error in the IMU frame $r_{\mathbb{I}ij} = \left[r_{\Delta\mathbf{R}_{ij}}^\top, r_{\Delta\mathbf{v}_{ij}}^\top, r_{\Delta\mathbf{p}_{ij}}^\top\right]^\top$ is naturally converted from (6):

$$\begin{aligned}
r_{\Delta\mathbf{R}_{ij}}^\top &= \log(\Delta\bar{\mathbf{R}}(b_{w_i}))\mathbf{R}_i^\top \mathbf{R}_j, \\
r_{\Delta\mathbf{v}_{ij}}^\top &= \mathbf{R}_i^\top (\mathbf{v}_j - \mathbf{v}_i - \boldsymbol{g}\Delta t_{ij}) - \Delta\bar{\mathbf{v}}_{ij}(\boldsymbol{b}_{w_i}, \boldsymbol{b}_{a_i}), \\
r_{\Delta\mathbf{p}_{ij}}^\top &= \mathbf{R}_i^\top (\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i\Delta t_{ij} - \frac{1}{2}g\Delta t_{ij}^2) - \Delta\bar{\mathbf{p}}_{ij}(\boldsymbol{b}_{w_i}, \boldsymbol{b}_{a_i}).
\end{aligned} \tag{5}$$

The relative states of the other sensors are predicted by the IMU preintegration measurements within the time interval between consecutive frames. Superscript O represents the target sensor frame including the LiDAR frame and camera frame. Then, the prediction state can be expressed as follows:

$$\begin{aligned}
\Delta\mathbf{R}_{ij}^O &= \left(\mathbf{R}_i^O\right)^\top \mathbf{R}_j^B \doteq \Delta\bar{\mathbf{R}}_{ij}^B \\
\Delta\mathbf{v}_{ij}^O &= \left(\mathbf{R}_i^O\right)^\top \left(\mathbf{v}_j^O - \mathbf{v}_i^O - g\delta t\right) \doteq \mathbf{R}_B^O \Delta\bar{\mathbf{v}}_{ij}^B \\
\Delta\mathbf{p}_{ij}^O &= \left(\mathbf{R}_i^O\right)^\top \left(\mathbf{p}_j^O - \mathbf{p}_i^O - \mathbf{v}_i^O\delta t - \frac{1}{2}g\delta t^2\right) \doteq \mathbf{R}_B^O \Delta\bar{\mathbf{p}}_{ij}^B
\end{aligned} \tag{6}$$

where $\mathbf{R}_B^O$ is the rotation matrix in the external parameter between the target frame and the IMU frame. The solved relative states are leveraged to the feature tracking process in the visual module and the distortion elimination process in the LiDAR module.

IMU odometry error is triggered by slowly varying random drift in the accelerometer bias, $\boldsymbol{b}_{a_i}$, and gyroscope bias, $\boldsymbol{b}_{w_i}$. IMU measurement biases are jointly corrected using the LiDAR and visual poses. Additionally, the system state is optimized using multiple IMU measurement residuals, which are equivalent to the errors $\left\{E_{ij}^C, E_{ij}^L\right\}$ between the IMU measurements and poses from the visual and LiDAR modules. The state estimation problem in Equation (2) is converted to minimize the IMU measurement residuals, as follows:

$$\mathcal{X}_n = \arg\min_{\mathcal{X}_n} \sum_{(i,j)\in\mathcal{T}_n} (E_{ij}^C)^\top \Omega_{ij}^C E_{ij}^C + (E_{ij}^L)^\top \Omega_{ij}^L E_{ij}^L + E_m. \tag{7}$$

where Ω_{ij} is the uncertainty matrix of LiDAR and visual poses, which can be calculated based on the sensor measurement noise. E_m is the marginalized prior, consisting of states and observations before the oldest state in the sliding window. The factor graph is shown in Figure 3.

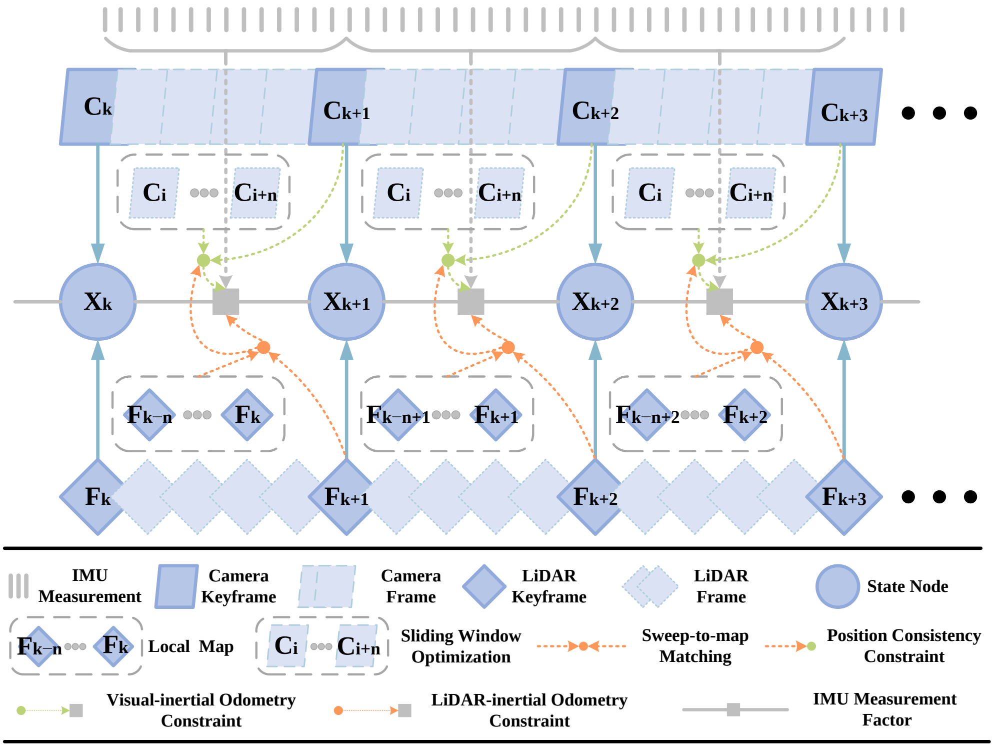


Figure 3. Factor graph of the system. The IMU module is constrained by LiDAR and visual modules, and ultimately outputs a refined system state.

4.3. Visual Module with Position Consistency Constraint

The visual-inertial module is implemented based on the LiDAR module assistance. In this module, the point reprojection method is designed to remove outliers of depth points. The position from LiDAR back-propagation is incorporated into visual-inertial optimization, achieving a refined camera pose.

4.3.1. Depth Association by Outliers Rejection

By utilizing external parameters between LiDAR and camera, the multiple LiDAR sweeps are projected on the image plane to generate depth map. Each feature depth is associated with three adjacent points on a unit sphere frame. The depth value is typically solved by spherical interpolation [28]. However, points with large incident angles of LiDAR will produce deviation. Therefore, the relative poses in the sliding window are leveraged to cross-cut evaluate the projection errors of depth points under different perspectives. In detail, the 3D landmark of the visual features z_{ic} and z_{jc} with depths d_i and d_j in frames F_i and F_j can be defined as $\mathbf{P}^C$. The reprojection residual error related to $\mathbf{P}^C$ between frames F_i and F_j in the sliding window can be written as:

$$r_{\mathbb{C}}(z_{ic}) = e_i^{rp}(\mathbf{P}^C, \boldsymbol{x}_i) = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} (\mathbf{P}^C - \pi_c(z_{jc})),$$
$$z_{jc} = \mathbf{R}_B^C(\mathbf{R}_W^{B_j}(\mathbf{R}_{B_k}^W(\mathbf{R}_C^B d_i \pi_c^{-1}(z_{ic}) + \mathbf{t}_C^B) + \mathbf{t}_{B_k}^W) + \mathbf{t}_W^{B_j}) - \mathbf{t}_B^C \tag{8}$$

where $\pi_c^{-1}(\cdot)$ is the back projection function. $\mathbf{R}_B^C$ and $\mathbf{t}_B^C$ are the rotation matrix and translation vector of the transformation between camera frame and IMU frame. $\mathbf{R}_B^W$ and $\mathbf{t}_B^W$ represent poses in the world frame. b_1 and b_2 are two orthogonal bases that span the tangent plane of z_{jc}.

The depth of projected points from different perspectives is used to quantify the depth association:

$$\tau_i = \exp(-\beta \cdot e_i^{rp}(\mathbf{P}^C, \boldsymbol{x}_i)). \tag{9}$$

where the corresponding projection points are considered outliers if τ is greater than the preset threshold. β is a decay coefficient that has been manipulated. The average depth value of the interior projected points is calculated as the final depth value. In addition, the visual features will be removed if the number of depth measurements is below a certain threshold. The final features with modified depth are further utilized to update the visual map.

4.3.2. Motion Estimation Assisted by LiDAR Odometry Back-propagation

The aims of the visual module are improving the robustness of the overall system and providing an initial guess for the LiDAR module. In our work, the LiDAR point clouds and visual images are synchronized and consistent. To improve the localization accuracy of the visual odometry, the position consistency constraint is proposed by incorporating backward propagated poses from the LiDAR module into the sliding window optimization. The definition of the above constraint is as follows:

$$\boldsymbol{C}_{po}(\mathbf{T}_k^C) = \left\| [\mathbf{T}_{k-1}^L \mathbf{T}_L^C (\mathbf{T}_{k-1}^C)^{-1} \mathbf{T}_k^C]_{:t} - (\mathbf{T}_k^C)_{:t} \right\|^2 \tag{10}$$

where $\boldsymbol{C}_{po}(\mathbf{T}_k^C)$ ensures that the location of the camera and the transformed LiDAR, using the external parameters $\mathbf{T}_L^C$, are consistent at the k-th keyframe. $\mathbf{T}_{k-1}^L$ and $\mathbf{T}_{k-1}^C$ are the 6-DoF poses of LiDAR and the camera frame at the $k-1$-th keyframe, respectively. $()_{:t}$ represents the translation vector of the transformation matrix.

For each newly acquired camera keyframe, motion estimation is implemented by jointly adjusting the camera poses $\mathbb{T}^C = (\mathbf{T}_1^C, \cdots, \mathbf{T}_n^C)$ and the 3D observations, $\mathbb{C}$. This process is formulated by minimizing the sum of the feature observation residual error, $r_{\mathbb{C}}(\boldsymbol{z}_{kc})$, IMU preintegration error, $r_{\mathbb{I}}$, and constraint, $\boldsymbol{C}_{po}$, as follows:

$$(\mathbf{T}_k^C)^* = \arg\min_{\mathbf{T}_k^C} \left\{ r_{\mathbb{I}} + + \sum_{c\in\mathbb{C}_k} \|r_{\mathbb{C}}(\boldsymbol{z}_{kc})\|^2 + \boldsymbol{C}_{po} \right\} \tag{11}$$

The IMU states and visual features of the regular frames and removed keyframes are marginalized as prior to constrain the sliding window optimization [29]. In addition, if the LiDAR module fails completely, the visual pose is output as the final state to ensure robustness.

4.4. Adaptive LiDAR Module with Hybrid Registration Mode

Existing feature-based LiDAR-inertial odometry may fail to work in geometrically uninformative environments, pushing optimization toward divergence along weakly constrained directions. In this paper, the LiDAR module adopts a hybrid registration method, which utilizes a feature-based approach to obtain the rough pose in high-level structured scenes and performs the point-to-surface matching algorithm under IMU constraints in unstructured scenes, achieving accurate and robust pose estimation.

4.4.1. Structure Monitor Based on Vertical Landmarks

After receiving a new sweep, the firstly performed feature extraction is employed to divide the original point cloud into edge points, $\mathbb{F}_n^e$, and plane points, $\mathbb{F}_n^p$, according to the smoothness of the local surface [30]. The set of total features is denoted as $\mathbb{F}_n$ at the n-th frame. The smoothness, $\mathcal{S}(\boldsymbol{p}_i)$, of the LiDAR point $\boldsymbol{p}_i$ is calculated as follows:

$$\mathcal{S}(\boldsymbol{p}_i) = \frac{1}{|\mathscr{B}_i|} \|\textstyle\sum_{j\in\mathscr{B}_i, j\neq i}(\boldsymbol{r}_j - \boldsymbol{r}_i)\|, r_b < \|\boldsymbol{r}_i\| \le r_m. \tag{12}$$

where $\|\boldsymbol{r}_i\|$ represents the range of the point $\boldsymbol{p}_i$. r_b and r_m are the blind distance and the maximum distance of the return points, respectively. $\mathscr{B}_i$ is the set of consecutive points of p_i from the same scan. $|\mathscr{B}_i|$ is the number of points in $\mathscr{B}_i$.

The edge points are extracted if $\mathcal{S}(\boldsymbol{p}_i)$ greater than preset threshold, $\mathcal{S}_{th}$. Then, all edge points of the current sweep are projected onto the horizontal plane, which is divided into N_s regions. The direction from the LiDAR origin to the edge points is used to allocate the edge points on the horizontal plane. Assume the positions of the edge features in the horizontal plane follow a Gaussian distribution. In each region, the 2D positions will be used as samples for local kernel density estimation, iteratively moving in the direction of increasing density. As a result, the sample points will eventually converge at the local maximum density, and the points that converge to the same local maximum are considered members of the same cluster. The final clustering result is the set of points in the region with the highest density of edge points. Figure 4 illustrates the aforementioned aggregation process. The N_s-dimensional vector is obtained by forming the results of all regions:

$$\mathcal{U}_n = [\eta_1 \boldsymbol{V}_1, \eta_2 \boldsymbol{V}_2, \cdots, \eta_{N_s} \boldsymbol{V}_{N_s}]. \tag{13}$$

where η_i is the normalized factor: $\eta_i = N_i / N_{tatal}$. N_i represents the number of points in the i-th sector in the horizontal plane. N_{tatal} is the number of all points in the horizontal plane. If $N_i < N_{th}$, η_i is set to zero, where N_{th} is the preset threshold of points number in one sector. The weighted distance vector of the i-th sector can be calculated by:

$$\boldsymbol{V}_i = \sum_{k=1}^{\boldsymbol{N}_i} \exp(-\frac{\boldsymbol{d}_k^h}{\sigma^2}). \tag{14}$$

where N_i is the number of points in the i-th sector in the horizontal plane. $\boldsymbol{d}_k^h$ is the horizontal range of the k-th point in the i-th sector. σ is the attenuation factor for adjusting distance weights.

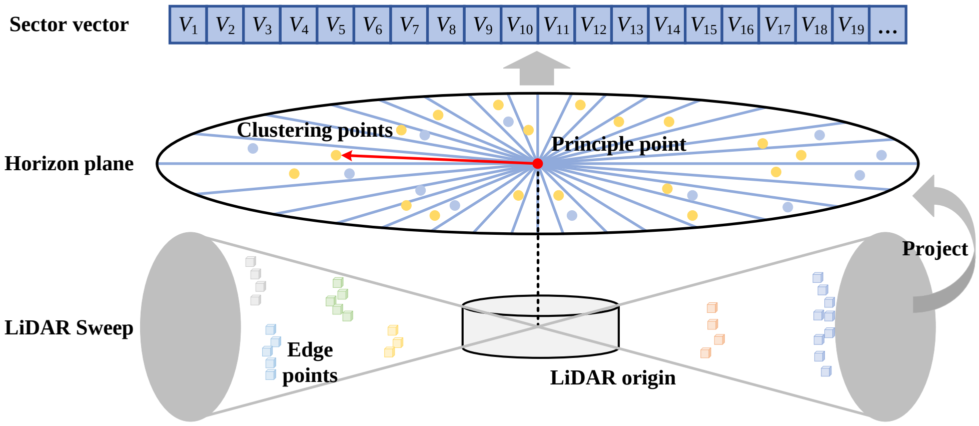

Sector vector
V_1 V_2 V_3 V_4 V_5 V_6 V_7 V_8 V_9 V_{10} V_{11} V_{12} V_{13} V_{14} V_{15} V_{16} V_{17} V_{18} V_{19} ...
Horizon plane
Clustering points
Principle point
Project
LiDAR Sweep
Edge points
LiDAR origin

Figure 4. The illustration of edge points aggregation. Different colored edge dots indicate different ranges. Vertical observations are projected onto the segmented horizontal plane for clustering.

The environmental structure, $\mathcal{C}_n$, is then quantified by the dispersion of the N_s-dimensional vector. A greater degree of dispersion indicates a higher level of environmental structuring. The details are illustrated by:

$$\mathcal{C}_n = var(\mathcal{U}_n) = \frac{1}{N_s} \sum_{i=1}^{N_s} (\eta_i \boldsymbol{V}_i - \frac{1}{N_s} \sum_{j=1}^{N_s} \eta_j \boldsymbol{V}_j). \tag{15}$$

The threshold $\mathcal{C}_{th}$, used in this method to distinguish between environmental structures, can be empirically set to 5. The quantitative environmental structuring is used to adjust the LiDAR module's operating mode.

4.4.2. Hybrid Point Cloud Alignment

The sparsity of LiDAR sweeps may lead to imprecise vertical constraints, especially in open and unstructured environments, failing to estimate the altitude variables *roll*, *pitch*, and z in point cloud alignment. To tackle these problems, we propose a hybrid point cloud alignment strategy that performs two modes according to the output of the structure monitor. In structured scenes, the point-to-feature model with distance weight is minimized to solve the LiDAR pose. In unstructured scenes, a novel point-to-surface model is generated to register the non-planar surface, achieving refined pose estimation. The point cloud alignment strategy is shown in Algorithm 1.

Algorithm 1 Hybrid Point Cloud Registration

```
Input: $\mathbb{F}_k, \mathscr{C}_k, \mathbf{T}^L_{k-1}, \mathbf{M}^l_{k-1}, \Delta\mathbf{R}^L_{(k-1,k)}, \Delta\mathbf{p}^L_{(k-1,k)}$
Output: $\mathbf{T}^L_k$
While $\mathbb{F}_k \neq \emptyset$ do
    $\check{\mathbf{T}}^L_k \leftarrow \text{InitialGuess}\left(\mathbf{T}^L_{k-1}, \mathbf{T}^C_{k-1}, \Delta\mathbf{R}^L_{(k-1,k)}, \Delta\mathbf{p}^L_{(k-1,k)}\right)$
    if $\mathscr{C}_k > \mathscr{C}_{th}$ then
        $S^{\mathbb{F}_k} \leftarrow \text{Point2FeatureDistance}\left(\mathbb{F}_k, \mathbf{M}^l_{k-1}\right)$
        $r_{pf} \leftarrow \text{WeightedRegistrationError}\left(S^{\mathbb{F}_k}, w(\mathbf{M}^l_{k-1})\right)$
        $\mathbf{T}^L_k \leftarrow \text{LiDARPose.minimize}\left(r_{pf}\right)$
    else
        $r_{pg} \leftarrow \text{Point2GaussianError}\left(\mathbb{F}_k, \mathbf{M}^l_{k-1}\right)$
        $\lambda(\mathbb{F}_k(\check{\mathbf{T}}^L_k)) \leftarrow \text{ResidualErrorTransform}(r_{pg})$
        $\mathbf{T}^L_k \leftarrow \text{LiDARPose.minimize}(\lambda(\mathbb{F}_k(\check{\mathbf{T}}^L_k)))$
    end if
    return $\mathbf{T}^L_k$
end
```

If $\mathscr{C}_n > \mathscr{C}_{th}$, the n-th sweep is considered structured. In this case, the j-th feature, f^L_j, in $\mathbb{F}_n$ is transformed from the m-th map point, $\mathbf{P}^G_m$, in the global frame by the pose $\mathbf{T}^L_n$, where $\mathbf{T}^L_n = (\mathbf{R}^L_n, \mathbf{t}^L_n)$. The feature in the global frame is defined by:

$$\mathbf{P}^G_m = \mathbf{R}^L_n \boldsymbol{f}^L_j + \mathbf{t}^L_n. \tag{16}$$

According to the widely used ICP method [31,32], the rigid transformation, $\mathbf{T}$, between the prior map, $\mathbf{M}^l$, and the feature cloud, $\mathbb{F}_n$, at the n-th frame can be solved by minimizing the feature registration error, $r_{pf}(\mathbf{P}^G_m)$, including the weighted point-to-feature distance $S^{\mathbb{F}_i}(\boldsymbol{f}_j, \mathbf{P}^G_m)$:

$$\begin{gathered} r_{\mathbb{L}}(z_{il}) = r_{pf} = \frac{\sum_{\boldsymbol{f}^L_j \in \mathbb{F}_i} w(\mathbf{P}^G_m) \cdot S^{\mathbb{F}_i}(\boldsymbol{f}^L_j, \mathbf{P}^G_m)}{\sum_{\boldsymbol{f}^L_j \in \mathbb{F}_i} w(\mathbf{P}^G_m)}, \\ S^{\mathbb{F}_i}(\boldsymbol{f}^L_j, \mathbf{P}^G_m) = \mathbf{n}^\top_m \left(\boldsymbol{f}^L_j - \mathbf{P}^G_m\right) + \mathbf{l}^\wedge_m \left(\boldsymbol{f}^L_j - \mathbf{P}^G_m\right). \end{gathered} \tag{17}$$

where $w(\mathbf{P}^G_m) = e^{-\left\|\mathbf{P}^G_m - c\right\|^2/\sigma^2}$ is the weight, which declines as distance is gained. The parameter σ is chosen to exclude features that are more than 3σ distances from the feature center, c, on the local map. $S^{\mathbb{F}_i}(\boldsymbol{f}^L_m, \mathbf{P}^L_m)$ is the distance function, where $\mathbf{n}_m$ and $\mathbf{l}_m$ represent dominant vectors of the corresponding feature.

Typically, point cloud alignment with point-to-feature error yields consistent matching results. However, in unstructured environments, edge points are insufficient to provide adequate constraints, and the numerous surfaces cannot be accurately represented by regular planes. The point-to-Gaussian distance is modeled to provide a more generalized representation with Gaussian mean and uncertainty. Figure 5 shows these two models in different scenes.

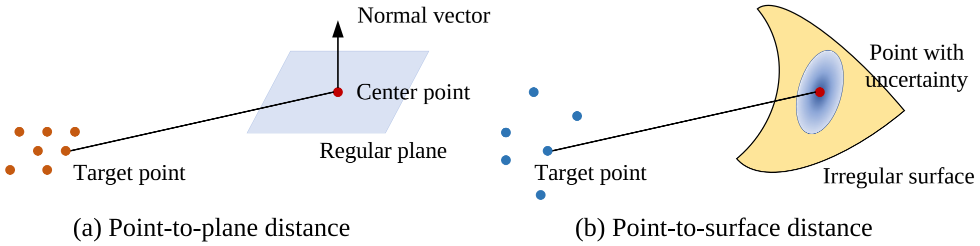


Figure 5. The model of the aligned point. (**a**) is the point-to-plane model that is employed in structured scenes. (**b**) is the point-to-surface model with uncertainty for aligning irregular ground points.

Let the neighbor region of $\mathbf{P}_m^G$ include a set of LiDAR points $\boldsymbol{q}_i (i = 1, \ldots, M)$. $\mathbf{P}_m^G$ has an uncertainty, $\Sigma_{\mathbf{P}_m^G}$, due to LiDAR measurement noise and position estimation errors. The uncertainty model of an irregular surface is illustrated in Figure 5b. The Gaussian mean of $\mathbf{P}_m^G$ is set to its 3D position. The uncertainty is the inverse of the covariance, $\boldsymbol{W}$, which is calculated from the neighbor points and is expressed as $\boldsymbol{W} = 1/M \sum_{i=1}^{M} (\boldsymbol{q}_i - \mathbf{P}_m^G)^\top (\boldsymbol{q}_i - \mathbf{P}_m^G)$. The cost function of the point-to-Gaussian surface is described as follows:

$$(\mathbf{T}_n^L)^* = \arg\min_{\mathbf{T}_n^L} r_{pg}(\mathbf{M}^l, \mathbb{F}_n, \mathbf{T}_n^L) = \arg\min_{\mathbf{T}_n^L} \sum_{j=1} (e^\top \boldsymbol{W}^{-1} e)_j \tag{18}$$

where the Mahalanobis distance between the target frame point and the corresponding Gaussian surface point is minimized rather than the Euclidean distance.

The inverse matrix of covariance $\boldsymbol{W}^{-1}$ is composed of an eigenvector matrix, $\boldsymbol{N}$, and a diagonal matrix, Λ, formed by eigenvalues. So, we can calculate the point-to-surface error by the decomposed W^{-1}:

$$\begin{aligned} r_{pg}(\mathbf{M}^l, \mathbb{F}_n, \mathbf{T}_n^L) &= \sum_{j=1} (e_j^\top \boldsymbol{N} \Lambda \boldsymbol{N}^\top e_j) \\ &= \sum_{j=1} e_j^\top [v_1, v_2, v_3] \mathrm{diag}(\lambda_1, \lambda_2, \lambda_3) [v_1, v_2, v_3]^\top e_j \\ &= \sum_{j=1} \lambda_1 (e_j^\top v_1 v_1^\top e_j) + \lambda_2 (e_j^\top v_2 v_2^\top e_j) + \lambda_3 (e_j^\top v_3 v_3^\top e_j). \end{aligned} \tag{19}$$

where $\lambda_1, \lambda_2, \lambda_3$ are descending eigenvalues of $\boldsymbol{W}^{-1}$, and v_1, v_2, v_3 are the corresponding eigenvectors. Then, the standard least squares definition of the point-to-surface cost function can be obtained:

$$r_{pg}(\mathbf{M}^l, \mathbb{F}_n, \mathbf{T}_n^L) = \sum_{k=1} \sum_{i=1} \lambda_k (\tilde{\boldsymbol{f}}_j (\check{\mathbf{T}}_n^L)) = \sum_{k=1} \sum_{j=1} \lambda_k \left\| v_k^\top (\check{\mathbf{R}}_n^L \boldsymbol{f}_j^L + \check{\mathbf{t}}_n^L - \mathbf{P}_m^G) \right\|_2^2 \tag{20}$$

where $\check{\mathbf{R}}_n^L$ and $\check{\mathbf{t}}_n^L$ are the rotation and translation part of the initial transformation, $\check{\mathbf{T}}_n^L$. The initial pose, $\check{\mathbf{T}}_n^L$, can be jointly calculated by the previous pose, $\mathbf{T}_{n-1}^L$, and the prediction state $\left\{ \Delta \mathbf{R}_{(n-1,n)}^L, \Delta \mathbf{p}_{(n-1,n)}^L \right\}$ from IMU. $\tilde{\boldsymbol{f}}_j$ is the transformed point from $\mathbb{F}_n$ by $\check{\mathbf{T}}_n^L$. Further, the second-order Taylor expansion of the eigenvalue matrix $\lambda_k (\tilde{\boldsymbol{f}}_j (\check{\mathbf{T}}_n^L))$ is denoted by:

$$\lambda_k (\tilde{\boldsymbol{f}}_j (\check{\mathbf{T}}_n^L)) \approx \lambda_k \left(\tilde{\boldsymbol{f}}_j \right) + \boldsymbol{J}(\tilde{\boldsymbol{f}}_j) \delta \boldsymbol{f} + \frac{1}{2} \delta \tilde{\boldsymbol{f}}^\top H(\tilde{\boldsymbol{f}}_j) \delta \boldsymbol{f} \tag{21}$$

where $\boldsymbol{J}(\tilde{\boldsymbol{f}}_j)$ and $\boldsymbol{H}(\tilde{\boldsymbol{f}}_j)$ are the Jacobian matrix and the Hessian matrix of $\lambda_k (\tilde{\boldsymbol{f}}_j (\check{\mathbf{T}}))$, respectively.

Additionally, due to the insufficient vertical constraints in unstructured scenes, the point-to-surface model is modified for 3-DoF state estimation, that is, only the three horizontal DoF poses will be estimated. In this case, projecting the corresponding pose $\mathbf{T}_n^L$ of $\tilde{\boldsymbol{f}}_j$ on the tangent plane, we have:

$$\begin{gathered}
\mathbf{T}_n^L = \check{\mathbf{T}}_n \oplus \delta\mathbf{T} = \big(\mathbf{R}(\gamma_n)\exp(\delta\gamma_n^{\wedge}), \mathbf{t}_n|_{x,y} + \delta\mathbf{t}\big) \\
\mathbf{R}(\gamma_n) = \begin{bmatrix} \cos\gamma_n & -\sin\gamma_n & 0 \\ \sin\gamma_n & \cos\gamma_n & 0 \\ 0 & 0 & 1 \end{bmatrix} \approx \mathbf{I} + \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\gamma_n \\
\tilde{\boldsymbol{f}}_j = \mathbf{R}(\gamma_n)\exp(\delta\gamma_n^{\wedge})\boldsymbol{f}_j^L + \mathbf{t}_n|_{x,y} + \delta\mathbf{t}
\end{gathered} \tag{22}$$

By differentiating the projected point $\tilde{\boldsymbol{f}}_j$ with respect to pose $\hat{\mathbf{T}}_n^L$, we can obtain:

$$\begin{gathered}
\lambda(\check{\mathbf{T}}_n^L \oplus \delta\mathbf{T}) \approx \lambda\big(\check{\mathbf{T}}_n^L\big) + \underbrace{\boldsymbol{J}\mathscr{A}}_{\tilde{\boldsymbol{J}}}\,\delta\mathbf{T} + \frac{1}{2}\delta\mathbf{T}^{\top}\underbrace{\mathscr{A}^{\top}\boldsymbol{H}\mathscr{A}}_{\tilde{\boldsymbol{H}}}\,\delta\mathbf{T} \\
\mathscr{A} = \frac{\delta\tilde{\boldsymbol{f}}_j}{\delta\mathbf{T}} = \left[-(\boldsymbol{f}_j^L)^{\wedge} - \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\gamma_n(\boldsymbol{f}_j^L)^{\wedge} \quad I \right]
\end{gathered} \tag{23}$$

The objective function of point cloud registration can be formulated by:

$$(\tilde{\boldsymbol{H}}(\mathbf{T}_n^L) + \mu\mathbf{I})\delta\mathbf{T}^{*} = -\tilde{\boldsymbol{J}}(\mathbf{T}_n^L)^{\top}. \tag{24}$$

where μ is employed to adjust the iterative process. Finally, we minimize the cost function to refine pose $\mathbf{T}_n^L$ by repeatedly calculating its second-order derivative using the Levenberg–Marquardt (LM) approach.

5. Experimental Results

The full system's tests are conducted on an Intel Core i7-10700K CPU with 16 GB RAM. In this section, the accuracy and robustness of the proposed system are evaluated in off-line mode, and the runtime evaluation demonstrates that the proposed system can operate in real time. The UAV localization accuracy is tested between the proposed algorithm and the advanced algorithms in highly structured scenes. In robust testing, trajectory comparisons for structurally weaker scenes with two advanced LiDAR-visual-IMU odometry methods is performed. A public dataset and a private dataset are employed as test sets. The NTU-VIRAL dataset is collected by equipping the UAV with sensors such as 3D rotating LiDARs, global shutter cameras, IMUs, and ultra wideband ranging units [33]. This dataset records multiple sequences under several challenging indoor and outdoor conditions. In this experiment, nine outdoor sequences from NTU-VIRAL are used for localization accuracy evaluation. In addition, we acquire a campus sequence of the low-altitude environment for testing location accuracy, and construct a high-precision 3D map. Three structurally different outdoor scenes are captured using LiDAR, a pinhole camera, and IMU: grove, beach, and desert. The platform and the private dataset are shown in Figure 6.

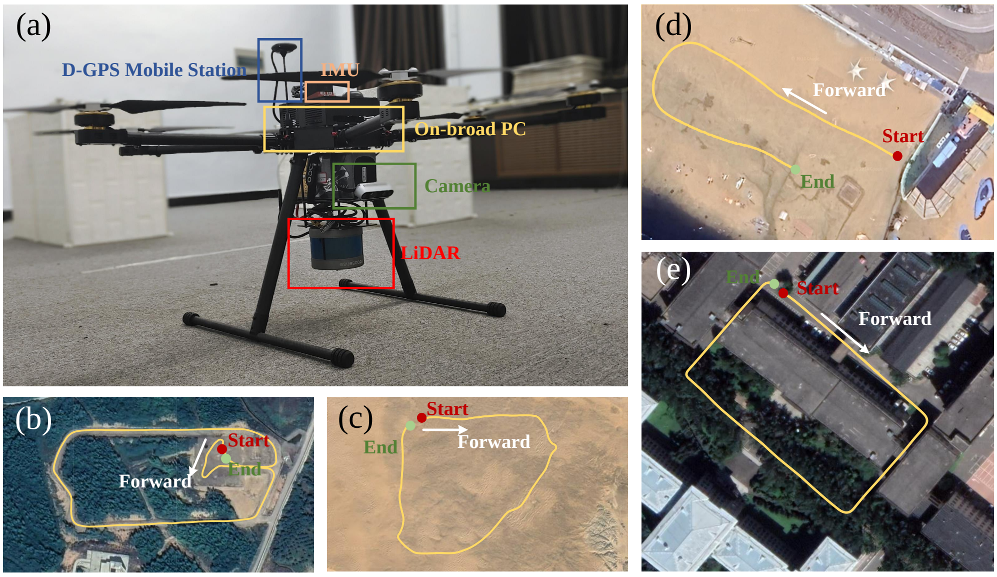


Figure 6. The platform is used for research and collection of the private dataset. The UAV in (**a**) is equipped with a GPS mobile station, LIDAR, on-board computer and pinhole camera. The world frame is defined as the first IMU frame. Satellite photographs (**b**–**e**) show four scenes. The orange curves represent the ground truth of these sequences as determined by the GNSS/IMU positioning system.

5.1. Structure Monitor Evaluation

In this section, a vertical observation-based structural monitor is employed to quantify the structure of different scenarios. The sequences from both public and private datasets are quantified separately, as shown in Table 2. A data sequence is defined, structured if its quantitative result exceeds 5; otherwise, it is regarded as unstructured. As shown in Table 2, the quantitative results indicate that all sequences in the public dataset and the campus sequence in the private dataset were collected in structured environments, whereas the grove, beach, and desert sequences reflect unstructured environments. The quantitative results of the unstructured scenes are show in Figure 7. In the structured environments, localization accuracy is the primary concern, while, in unstructured scenes, the focus shifts to ensuring stable system operation.

Table 2. Quantitative structuring on public and private datasets.

Data	rtp1	rtp2	rtp3	sbs1	sbs2	sbs3	tnp1	tnp2	tnp3	Camp	Grove	Beach	Desert
Result	5.21	5.53	5.39	6.03	5.86	5.15	6.84	6.67	5.51	6.30	4.13	2.26	0.04

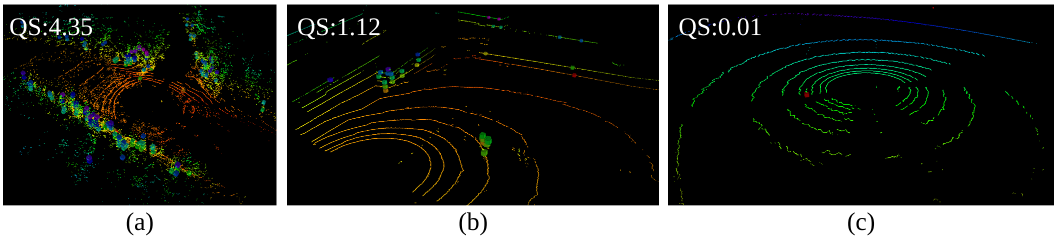


Figure 7. The qualitative structures of the private dataset are shown. (**a**–**c**) are grove sequence, beach sequence, and desert sequence, respectively. Cubes are salient edge points. The QS stands for quantitative structure.

5.2. System Localization Accuracy Evaluation

The evaluation results of localization accuracy are shown in Table 3. The localization error is dictated by the root mean square error (RMSE) of translation estimation. As shown in Table 2, the proposed method has higher localization accuracy than state-of-the-art LiDAR-visual-IMU odometry in structured scenes such as rtp, sbs, and tnp data sequences. In the NTU-VIRAL public dataset, the proposed system can maintain near-optimal localization performance in most data sequence tests. However, its performance in the rtp data sequence is slightly inferior to that of the LiDAR-IMU system. This is because strong lighting variations interfere with the visual module's localization accuracy, thus increasing the translation error. Additionally, the LiDAR module is not affected by sensor degradation in the rtp data sequence. Therefore, LiDAR-inertial odometry can obtain better location results than LiDAR-visual-inertial odometry with optical noise interference. Unlike other tightly coupled sensor fusion localization schemes such as FAST-LIVO and mVIL-Fusion [34], the proposed system uses the output of the visual module only as the initial value for the next pose estimation stage, making it less dependent on the visual module. Even if the visual module fails, the proposed system can still deliver accurate positioning. The real-time mapping results of rtp, sbs, and tnp are shown in Figure 8; the proposed system provides structured point clouds and signs without distortion to exhibit low-drift localization performance. The 3D point cloud map of the campus sequence in the private dataset is shown in Figure 9, where static objects can all be recognized clearly. In addition, Figure 10 illustrates the positioning error of the proposed system in the publicly available NTU-VIRAL dataset, with a focus on annotating the minimum error, average error, and maximum error between the proposed system and the ground truth trajectories. Overall, in testing with the public dataset, our proposed system reduced localization errors by 25.1%, 29.5%, 15.7%, 50.0%, and 55.8% compared with LIO-SAM, LVI-SAM, FAST-LIO2, FAST-LIVO, and mVIL Fusion, respectively, after excluding the failure and mismatch of each system.

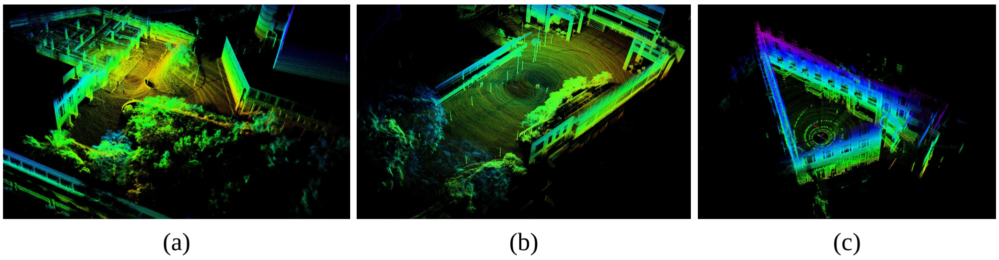


Figure 8. Point cloud maps of NTU dataset are shown. (**a**–**c**) are rtp sequences, sbs sequences, and tnp sequences, respectively.

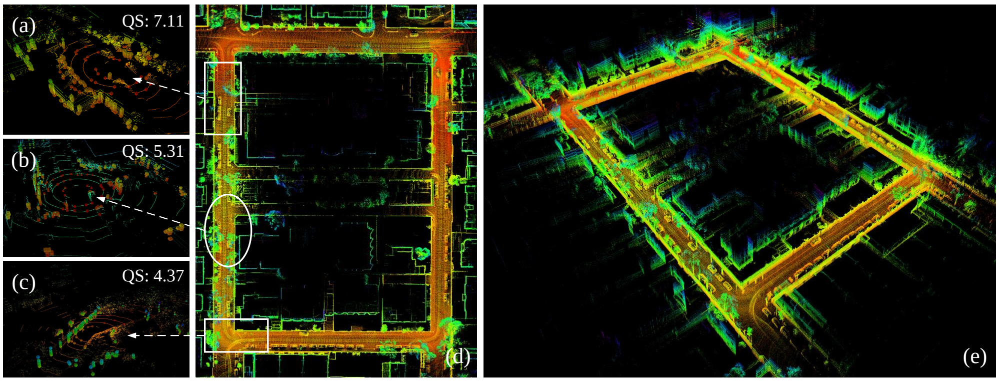
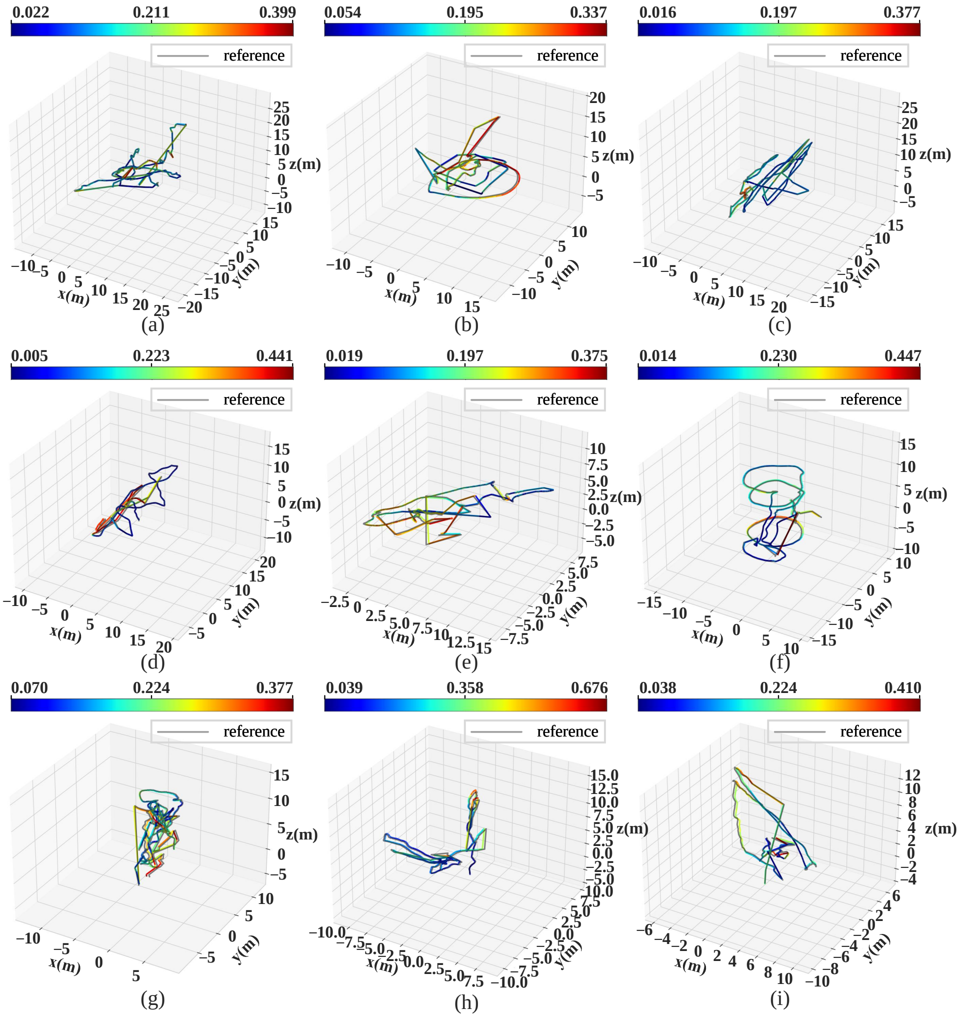


Figure 9. 3D bird's-eye view map of the campus sequence. The quantitative structure of the three locations is emphasized as (**a**–**c**). Figure (**d**,**e**) show point cloud maps from a bird's-eye perspective, presenting the consistent map without point cloud divergence.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 10. Localization accuracy experiments are performed on the NTU-VIRAL dataset. The trajectories' errors are compared with the ground truths, which are provided by the public dataset. Subfigures (**a**–**i**) represent the trajectory results for the rtp sequence, the sbs sequence, and the tnp sequence, respectively. The "reference" in the subfigure is the ground truth of the UAV trajectory. The heat map color of the estimated trajectories indicate the error level.

Table 3. Localization error on public and private datasets. (UNIT: Meter).

Data	LIO-SAM	LVI-SAM	FAST-LIO2	FAST-LIVO	mVIL-Fusion	Ours
rtp1	0.242 [1]	X [2]	**0.148** [3]	0.674	0.954	0.265
rtp2	**0.177**	X	0.195	0.861	0.673	0.201
rtp3	0.385	0.204	0.195	0.283	0.426	**0.141**
sbs1	0.214	0.215	0.223	0.351	0.213	**0.114**
sbs2	0.208	0.208	0.213	0.232	0.225	**0.200**
sbs3	0.179	X	0.210	0.210	0.193	**0.175**
tnp1	0.193	0.134	0.146	0.202	0.269	**0.214**
tnp2	0.192	0.180	0.169	**0.124**	0.229	0.168
tnp3	0.176	0.479	0.181	0.165	0.223	**0.109**
campus	0.466	1.641	0.457	0.395	- [4]	**0.312**
grove	5.237	6.318	5.047	5.215	-	**4.880**
beach	0.453	6.449	1.171	2.159	-	**0.241**
desert	X	X	X	X	-	**10.87**

[1] underlined numbers are the sub-optimal results. [2] X means that the system failed to complete the full localization. [3] **Bolded** numbers are the best ones. [4] - means that the system is not applicable.

5.3. System Robustness Evaluation

In this section, hierarchical unstructured scenarios are employed to test the localization accuracy of the proposed system. The localization error is shown in Table 2. The difficulty of grove, beach, and desert sequences increases progressively. The grove scene is the most structured, with minimal brightness variation, making the environment relatively easier. However, due to the grove sequence covering a distance of 983.5 m, the cumulative error is relatively large. The beach scene covers only 137.3 m and, like the *rtp* sequence in the public dataset, has strong visual interference. Due to its low level of structuring, the positioning error is larger compared with the *rtp* sequence. The total distance of the desert sequence is 988.3 m, with minimal changes in luminosity. However, the degree of structuring is extremely low, and there is almost no vertical observation. As a result, pose estimation can only rely on ground point clouds, leading to the complete failure of feature-based LiDAR odometry methods. Despite this, our method with point-to-surface registration can still maintain high accuracy in this scenario, ensuring that system localization does not fail. Figure 11 shows the positioning trajectory of the proposed system compared with FAST-LIVO and LVI-SAM in unstructured scenarios. Compared with these two state-of-the-art methods, the proposed system is closer to the ground truth and has relatively smaller positioning errors.

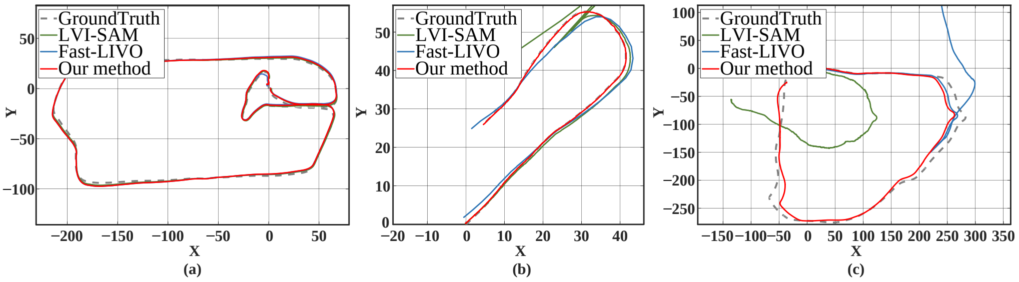


Figure 11. Trajectory comparisons on the private dataset. Subfigures (**a**–**c**) are indicate the comparisons on the grove, beach, and desert sequences. As the degree of non-structuring increases, the robustness of LVI-SAM and FAST-LIVO decreases. The proposed system can still maintain high localization accuracy.

5.4. Runtime Evaluation

The runtime of the proposed method is evaluated in the grove sequence. As shown in Table 4, the time consumption of the proposed system is divided into the visual module, LiDAR module, and IMU module. The visual module includes the extraction and matching of visual features, feature management, and data alignment, as well as image-based pose solving. The LiDAR module includes preprocessing, feature extraction, structure monitor, and hybrid point cloud tracking. The IMU module has an IMU measurement preintegration section and factor graph optimization section. These three modules run in parallel mode. According to the results in Table 3, the average running time of the vision module, LiDAR module, and IMU module is 48.1 ms, 39.17 ms, and 0.425 ms, respectively, which is lower than the acquisition time of the camera and LiDAR. Overall, the experimental results demonstrate that the proposed system can operate in real time.

Table 4. Time consumption on each component. (UNIT: MILLISECOND)

Component		Median	Mean	Std
Visual module	Feature tracking	17.11	17.26	5.16
	Feature management	17.22	18.31	4.39
	Pose estimation	13.19	12.53	5.33
LiDAR module	Preprocessing	9.49	11.61	3.92
	Feature extraction	1.15	1.29	1.72
	Structure monitor	0.15	1.56	3.67
	Hybrid point registration	22.55	24.71	17.43
IMU module	Preintegration	0.143	0.151	0.360
	Factor graph optimization	0.280	0.274	0.311

6. Conclusions

This article presents a sensor fusion system coupling LiDAR, cameras, and IMUs to reduce trajectory error and ensure robust operation of UAVs. The visual module is supported by LiDAR measurements and pose to estimate feature depth and visual pose. The estimated feature depth is validated by calculating the reprojection error using corresponding 3D observations within a sliding window in the visual module, which helps to remove outliers. Additionally, the LiDAR pose is back-propagated to constrain visual pose estimation, achieving an enhanced camera pose of the UAV. The LiDAR module employs a structure monitor to switch matching modes in various environments. In unstructured environments, we use Gaussian probability-based uncertainty to model irregular surfaces. This uncertainty is then decoupled into eigenvalues and eigenvectors, and a pose estimation objective function is constructed to achieve accurate localization. Finally, the IMU measurement errors with the LiDAR and visual modules are used to construct the odometry factor, which is incorporated into the factor graph optimization to complete the state solution of the UAV. Experimental results with UAVs demonstrate that the proposed system outperforms state-of-the-art algorithms by averagely reducing localization errors by at least 15.7%. In unstructured scenarios, the algorithm proposed in this paper leads the other control algorithms by at least 12.6%, effectively improving the localization accuracy of UAVs in unstructured environments. The system runs about 48.525 ms per frame, which meets the task requirements for real-time work. In future research, we will explore a sensor fusion architecture on UAVs that integrates satellite navigation and full-domain place recognition for long-term localization, aiming to maintain system accuracy and robustness.

Author Contributions: Conceptualization, B.Z. and X.S.; methodology, B.Z.; software, Y.W.; validation, B.Z., X.S. and G.S.; formal analysis, W.Y.; investigation, X.S.; resources, B.Z.; data curation, Y.W.; writing—original draft preparation, B.Z.; writing—review and editing, X.S.; visualization, Y.W.; supervision, W.Y.; project administration, G.S.; funding acquisition, G.S. All authors have read and agreed to the published version of the manuscript.'

Funding: This work was supported in part by the National Natural Science Foundation of China (U23A20346, 62033005, 62173107, and 62106062), in part by the Key Project of Natural Science Foundation of Heilongjiang Province (No. ZD2021F001), and in part by the Postdoctoral Fellowship Program of China Postdoctoral Science Foundation under Grant GZC20233462.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Liu, Y.; Bai, J.; Wang, G.; Wu, X.; Sun, F.; Guo, Z.; Geng, H. UAV Localization in Low-Altitude GNSS-Denied Environments Based on POI and Store Signage Text Matching in UAV Images. *Drones* **2023**, *7*, 451. [CrossRef]
2. Wang, R.; Deng, Z. Rapid Initialization Method of Unmanned Aerial Vehicle Swarm Based on VIO-UWB in Satellite Denial Environment. *Drones* **2024**, *8*, 451. [CrossRef]

3. Fu, J.; Yao, W.; Sun, G.; Ma, Z.; Dong, B.; Ding, J.; Wu, L. Multi-robot Cooperative Path Optimization Approach for Multi-objective Coverage in a Congestion Risk Environment. *IEEE Trans. Syst. Man Cybern. Syst.* **2024**, *54*, 1816–1827. [CrossRef]
4. Shao, X.; Sun, G.; Yao, W.; Liu, J.; Wu, L. Adaptive Sliding Mode Control for Quadrotor UAVs with Input Saturation. *IEEE-ASME T MECH.* **2022**, *27*, 1498–1509. [CrossRef]
5. Qin, T.; Li, P.; Shen, S. VINS-mono: A Robust and Versatile Monocular Visual-inertial State Estimator. *IEEE Trans. Robot.* **2018**, *34*, 1004–1020. [CrossRef]
6. Qin, C.; Ye, H.; Pranata, C.E.; Han, J.; Zhang, S.; Liu, M. LINS: A LiDAR-inertial State Estimator for Robust and Efficient Navigation. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation, Paris, France, 31 May–31 August 2020; pp. 8899–8906.
7. Shu, C.; Luo, Y. Multi-Modal Feature Constraint Based Tightly Coupled Monocular Visual-LiDAR Odometry and Mapping. *IEEE Trans. Intell. Veh.* **2023**, *8*, 3384–3393. [CrossRef]
8. Xie, J.; He, X.; Mao, J.; Zhang, L.; Hu, X. C2VIR-SLAM: Centralized Collaborative Visual-Inertial-Range Simultaneous Localization and Mapping. *Drones* **2022**, *6*, 312. [CrossRef]
9. Xu, W.; Zhang, F. FAST-LIO: A Fast, Robust LiDAR-inertial Odometry Package by Tightly-coupled Iterated Kalman Filter. *IEEE Robot. Autom. Lett.* **2021**, *6*, 3317–3324. [CrossRef]
10. Shan, T.; Englot, B.; Meyers, D.; Wang, W.; Ratti, C.; Rus, D. LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, NV, USA, 25–29 October 2020; pp. 5135–5142.
11. Nguyen, T.M.; Cao, M.; Yuan, S.; Lyu, Y.; Nguyen, T.H.; Xie, L. VIRAL-Fusion: A Visual-Inertial-Ranging-Lidar Sensor Fusion Approach. *IEEE Trans. Robot.* **2022**, *38*, 958–977. [CrossRef]
12. Zuo, X.; Yang, Y.; Geneva, P.; Lv, J.; Liu, Y.; Huang, G.; Pollefeys, M. LIC-Fusion 2.0: LiDAR-Inertial-Camera Odometry with Sliding-Window Plane-Feature Tracking. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, NV, USA, 25–29 October 2020; pp. 5112–5119.
13. Shan, T.; Englot, B.; Ratti, C.; Rus, D. LVI-SAM: Tightly-coupled Lidar-Visual-Inertial Odometry via Smoothing and Mapping. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation, Xi'an, China, 30 May–5 June 2021; pp. 5692–5698.
14. Lin, J.; Zhang, F. R3LIVE: A Robust, Real-time, RGB-colored, LiDAR-Inertial-Visual tightly-coupled state Estimation and mapping package. In Proceedings of the 2022 International Conference on Robotics and Automation, Philadelphia, PA, USA, 23–27 May 2022; pp. 10672–10678.
15. Zheng, C.; Zhu, Q.; Xu, W.; Liu, X.; Guo, Q.; Zhang, F. FAST-LIVO: Fast and Tightly-coupled Sparse-Direct LiDAR-Inertial-Visual Odometry. In Proceedings of the 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems, Kyoto, Japan, 23–27 October 2022; pp. 4003–4009.
16. Zhang, J.; Singh, S. Laser-visual-inertial Odometry and Mapping with High Robustness and Low Drift. *J. Field Robot.* **2018**, *35*, 1242–1264. [CrossRef]
17. Wisth, D.; Camurri, M.; Fallon, M. VILENS: Visual, Inertial, Lidar, and Leg Odometry for All-Terrain Legged Robots. *IEEE Trans. Robot.* **2023**, *39*, 309–326. [CrossRef]
18. Yuan, Z.; Wang, Q.; Cheng, K.; Hao, T.; Yang, X. SDV-LOAM: Semi-Direct Visual–LiDAR Odometry and Mapping. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 11203–11220. [CrossRef] [PubMed]
19. Shan, T.; Englot, B. LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, Madrid, Spain, 1–5 October 2018; pp. 4758–4765.
20. Chen, S.; Ma, H.; Jiang, C.; Zhou, B.; Xue, W.; Xiao, Z.; Li, Q. NDT-LOAM: A Real-Time Lidar Odometry and Mapping With Weighted NDT and LFA. *IEEE Sensors J.* **2022**, *22*, 3660–3671. [CrossRef]
21. Cui, Y.; Zhang, Y.; Dong, J.; Sun, H.; Chen, X.; Zhu, F. LinK3D: Linear Keypoints Representation for 3D LiDAR Point Cloud. *IEEE Robot. Autom. Lett.* **2024**, *9*, 2128–2135. [CrossRef]
22. Zhang, J.; Singh, S. LOAM: Lidar Odometry and Mapping in Real-time. In Proceedings of the 2014 Robotics: Science and Systems, Berkeley, CA, USA, 12–16 July 2014.
23. Guo, S.; Rong, Z.; Wang, S.; Wu, Y. A LiDAR SLAM with PCA-based feature extraction and two-stage matching. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–11. [CrossRef]
24. Choi, S.; Chae, W.H.; Jeung, Y.; Kim, S.; Cho, K.; Kim, T.W. Fast and Versatile Feature-Based LiDAR Odometry via Efficient Local Quadratic Surface Approximation. *IEEE Robot. Autom. Lett.* **2023**, *8*, 640–647. [CrossRef]
25. Chen, K.; Lopez, B.T.; Agha, M.; Ali, A.; Mehta, A. Direct LiDAR Odometry: Fast Localization with Dense Point Clouds. *IEEE Robot. Autom. Lett.* **2022**, *7*, 2000–2007. [CrossRef]
26. Chen, G.; Hong, L. Research on Environment Perception System of Quadruped Robots Based on LiDAR and Vision. *Drones* **2023**, *7*, 329. [CrossRef]
27. Forster, C.; Carlone, L.; Dellaert, F.; Scaramuzza, D. On-Manifold Preintegration for Real-Time Visual-inertial Odometry. *IEEE Trans. Robot.* **2017**, *33*, 1–21. [CrossRef]
28. Zuo, X.; Geneva, P.; Lee, W.; Liu, Y.; Huang, G. LIC-Fusion: LiDAR-Inertial-Camera Odometry. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, Macau, China, 4–8 November 2018; pp. 5848–5854.

29. Campos, C.; Elvira, R.; Rodriguez, G.J.J.; Montiel, M.M.J.; Tardos, J.D. ORB-SLAM3: An Accurate Open-source Library for Visual, Visual¨Cinertial, and Multimap SLAM. *IEEE Trans. Robot.* **2021**, *37*, 1874–1890. [CrossRef]
30. Xu, M.; Lin, S.; Wang, J.; Chen, Z. A LiDAR SLAM System With Geometry Feature Group-Based Stable Feature Selection and Three-Stage Loop Closure Optimization. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 1–10. Erratum in *IEEE Trans. Instrum. Meas.* **2023**, *7*, 524. [CrossRef]
31. Zhou, R.; Sun, H.; Ma, K.; Tang, J.; Chen, S.; Fu, L.; Liu, Q. Improving Estimation of Tree Parameters by Fusing ALS and TLS Point Cloud Data Based on Canopy Gap Shape Feature Points. *Drones* **2023**, *7*, 524. [CrossRef]
32. Baah, G.A.; Savin, I.Y.; Vernyuk, Y.I. Pollution from Highways Detection Using Winter UAV Data. *Drones* **2023**, *7*, 178. [CrossRef]
33. Nguyen, T.M.; Yuan, S.; Cao, M.; Lyu, Y.; Nguyen, T.H.; Xie, L. NTU VIRAL: A visual-inertial-ranging-lidar dataset, from an aerial vehicle viewpoint. *Int. J. Robot. Res.* **2022**, *41*, 270–280. [CrossRef]
34. Wang, Y.; Ma, H. mVIL-Fusion: Monocular Visual-Inertial-LiDAR Simultaneous Localization and Mapping in Challenging Environments. *IEEE Robot. Autom. Lett.* **2023**, *8*, 504–511. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.