Scientific Research Publishing

Open Journal of Applied Sciences, 2026, 16(3), 741-757
https://www.scirp.org/journal/ojapps
ISSN Online: 2165-3925
ISSN Print: 2165-3917

# Adaptive Lidar-Inertial SLAM Algorithm with Multi-Feature Assistance in Degraded Environments

**Yuhao Chen, Guoliang Wei, Zhixuan Miao, Zhi Li**

Business School, University of Shanghai for Science and Technology, Shanghai, China
Email: guolichen007@gmail.com, guoliang.wei@usst.edu.cn

**How to cite this paper:** Chen, Y.H., Wei, C.L., Miao, Z.X. and Li, Z. (2026) Adaptive Lidar-Inertial SLAM Algorithm with Multi-Feature Assistance in Degraded Environments. *Open Journal of Applied Sciences*, **16**, 741-757.
https://doi.org/10.4236/ojapps.2026.163045

**Received:** February 3, 2026
**Accepted:** March 2, 2026
**Published:** March 5, 2026



## Abstract

To address the issues of feature mismatching and map overlap drift in simultaneous localization and mapping (SLAM) within degraded environments characterized by sparse geometric features or severe dynamic interference, a multi-feature-assisted adaptive lidar-inertial SLAM system is proposed. The system first integrates lidar intensity information with inertial measurement unit (IMU) data to eliminate dynamic interference points, thereby enhancing point cloud quality. Subsequently, by fusing geometric features with intensity information, an adaptive feature extraction mechanism is designed to effectively improve the robustness of feature points in degraded environments. Finally, based on multidimensional scene complexity analysis, adaptive adjustment of optimization parameters is achieved, thereby enhancing the system's robustness and accuracy in complex scenarios. The core contributions of this work are as follows: 1) Intensity-Gradient Dynamic Filtering: A novel density-adaptive method that combines intensity gradients with IMU data for real-time motion correction and dynamic point removal, reducing interference by up to 30% compared to traditional geometric-only approaches in LIO-SAM. 2) Composite Feature Scoring: An innovative fusion of geometric curvature and intensity gradients into a weighted composite score, enabling robust feature selection in sparse environments and improving matching accuracy by 15% - 20% over baseline methods like LIO-SAM. 3) FDI-Driven Adaptive Optimization: A multi-level Full Degeneracy Index (FDI) that dynamically adjusts parameters such as iteration counts, constraint weights, and keyframe selection, achieving 14.9% and 33.8% pose accuracy improvements on KITTI and SubT-MRS datasets, respectively, relative to LIO-SAM. These contributions represent true novelties by extending beyond rigid fusion in prior works, introducing adaptive mechanisms that respond to environmental degradation in real-time, unlike static baselines. Experimental results on the KITTI dataset

and the SubT-MRS dataset demonstrate that, compared to traditional algorithms, the proposed method achieves significant improvements in pose accuracy and delivers superior localization and mapping performance across various complex environments, validating the effectiveness and robustness of the approach.

## Keywords

Lidar, Simultaneous Localization and Mapping, Degraded Environment, Adaptive, Multi Feature, IMU

## 1. Introduction

Lidar-based Simultaneous Localization and Mapping (SLAM) has been widely applied in fields such as autonomous driving, unmanned aerial vehicle (UAV) inspections, underground rescue operations, and indoor positioning [1]-[3]. In ideal environments, lidar provides stable and abundant geometric features, such as edge and planar points, laying a reliable foundation for high-precision pose estimation and three-dimensional mapping. However, in real-world scenarios, robots often encounter degraded environments characterized by sparse, repetitive, or dynamically interfered geometric features. In these cases, uneven point cloud distributions or feature deficiencies lead to mismatching and accumulated drift during the registration phase in traditional SLAM systems, resulting in trajectory deviations and map distortions that severely compromise the system's accuracy and robustness [4]-[6].

In this work, "degraded environments" are defined using measurable criteria derived from multidimensional scene analysis: 1) Sparsity, quantified by point cloud entropy, higher entropy indicates greater spatial distribution uncertainty and feature scarcity; 2) Repetitiveness, assessed via the geometric complexity factor, lower values signify repetitive structures based on eigenvalue distributions in principal component analysis; and 3) Dynamicity, evaluated through the dynamic interference factor, elevated levels reflect intensity gradient variations and curvature anomalies from moving objects. These criteria map to environmental categories as follows: structured, sparse, dynamic and mixed.

To mitigate the degradation in localization accuracy caused by such environments, researchers have proposed various improved lidar SLAM methods. Early approaches like LOAM [7] and LeGO-LOAM [8] employ edge and planar feature registration, combined with ground segmentation to enhance constraint quality; however, their loosely coupled structures make limited use of Inertial Measurement Unit (IMU) data and remain prone to drift in high-speed motion or feature-sparse scenarios. Subsequently, tightly coupled algorithms such as LIO-SAM [9] and LINS [10] achieve efficient fusion of lidar and inertial data through factor graph optimization, significantly improving the continuity and stability of pose estimation. The Fast-LIO series [11] [12] further advances computational effi-

ciency and real-time performance by adopting direct point-to-plane optimization to enable millisecond-level updates. Nevertheless, these methods generally rely on geometric structural information for constraints, and their registration robustness and stability remain limited when environments exhibit structural repetition or feature scarcity. Additionally, works such as Intensity-SLAM [13] and R$^3$LIVE [14] attempt to incorporate non-geometric information like reflectance intensity and image features, but their modality fusion mechanisms are relatively rigid and struggle to adaptively adjust for different types of degraded environments.

To address the aforementioned issues, this paper proposes an adaptive lidar-inertial SLAM optimization algorithm assisted by multi-feature integration. Based on the LIO-SAM framework, the algorithm first utilizes IMU data combined with intensity gradients for continuous motion correction and dynamic point removal, correcting distortions and eliminating interference by computing point cloud intensity variations and distance differences. Second, it introduces an intensity-assisted feature extraction mechanism that fuses intensity gradients with geometric curvature to compute composite scores, enabling robust feature selection in degraded environments. Finally, it incorporates multi-factor fused degradation detection to dynamically adjust thresholds and weights, achieving high-precision self-localization and globally consistent 3D maps.

Compared to the closest baseline, LIO-SAM, the proposed method introduces true novelties through:

1) Intensity-gradient dynamic filtering for enhanced dynamic interference removal, which LIO-SAM lacks.

2) Composite feature scoring that adaptively weights intensity and geometry, unlike LIO-SAM's geometry-only approach.

3) FDI-driven adaptive optimization for real-time parameter tuning, extending beyond LIO-SAM's static optimization.

Experimental results on the KITTI [15] and SubT-MRS [16] datasets demonstrate that, compared to LIO-SAM, the proposed algorithm improves pose accuracy by an average of 14.9% and 33.8%, respectively, with richer mapping details, validating its effectiveness.

## 2. Related Theories

### 2.1. Basic Paradigms of State Estimation: Filtering and Optimization

The essence of SLAM lies in addressing a high-dimensional, nonlinear state estimation problem. Its core objective is to recursively infer the robot's motion trajectory and the static environmental structure from a sequence of noisy sensor observations. The mathematical tools for solving this problem have primarily evolved along two main lines: filtering methods based on the Bayesian probabilistic framework and nonlinear optimization methods grounded in batch processing.

Filtering methods adhere to an online, recursive estimation paradigm. They

model the SLAM problem as a state estimation task for a dynamic system, where the state encompasses the robot's pose at the current time instant along with positions of partial or all environmental landmarks. Based on Bayes' theorem, the filter processes new sensor observations at each time step through prediction and update phases to compute the posterior probability distribution of the system state. Representative examples include the classical Kalman filter and its nonlinear extensions, as well as particle filters. The strengths of filtering methods lie in their incremental processing capability and potential for real-time operation, making them well-suited for online SLAM applications. However, their limitations include maintaining estimates only for the current state, with limited utilization of historical data, and linearization approximations that may lead to filter divergence in highly nonlinear scenarios or those with large initial errors. Additionally, accumulated errors over long durations are difficult to fully eliminate through front-end filtering alone.

In contrast, optimization methods adopt a more global perspective. They treat all robot poses over a period and the observed environmental landmarks as variables to be optimized, constructing motion and observation models into a large-scale factor graph. The SLAM problem is reformulated as a nonlinear least-squares problem [17], with the objective of minimizing the sum of squared residuals between predicted and actual observations. This approach leverages constraints across all available data to perform batch optimization, effectively mitigating accumulated errors and enhancing global consistency in estimation. Particularly upon loop closure detection, the optimization framework can seamlessly incorporate such strong constraints to smooth the entire trajectory. Mainstream open-source optimization libraries, such as g2o [18], Ceres Solver [19], and GTSAM [20], provide efficient solvers for this purpose. Optimization methods typically offer superior accuracy compared to filtering approaches, but their computational complexity grows with the number of variables, posing challenges to real-time performance; this is often addressed through sliding-window or incremental optimization strategies.

In lidar SLAM systems, especially those involving tightly coupled multi-sensor fusion, the two paradigms are not mutually exclusive but are frequently employed synergistically. A common architecture features the front-end utilizing lightweight filtering or incremental optimization for high-frequency, low-precision inter-frame motion estimation to ensure real-time capability, while the back-end operates at a lower frequency, conducting global pose and map refinement based on keyframes and factor graph optimization to guarantee accuracy and consistency.

## 2.2. Kalman Filter

The Kalman filter [21] provides a complete recursive solution for the optimal state estimator in linear Gaussian systems. Its core idea is to recursively estimate the true state of the system under the minimum mean square error criterion by com-

bining the system's dynamic model and observation model.

For a discrete-time linear system, the state equation and observation equation can be expressed as:

$$\begin{aligned} x_t &= Ax_{t-1} + Bu_t + w_t \\ z_t &= Hx_t + v_t \end{aligned} \tag{1}$$

where $x_t$ is the system state vector, $A$ is the state transition matrix, $u_t$ is the system input at the current time, matrix $B$ is the control matrix reflecting the mapping relationship between the system input and state, $z_t$ is the measurement vector, $H$ is the measurement matrix, and $w_t$ and $v_t$ are the process noise and observation noise, respectively, both satisfying Gaussian distributions.

The standard Kalman filter algorithm cyclically executes two steps: prediction and update. The prediction step utilizes the posterior estimate from the previous time and the system model to predict the prior estimate of the current state and its covariance, which can be expressed as:

$$\begin{aligned} \hat{x}_t &= A\hat{x}_{t-1}^{+} + Bu_t \\ P_t^{-} &= AP_{t-1}^{+}A^T + Q \end{aligned} \tag{2}$$

The update step computes $K$, the Kalman gain, with the formula:

$$\begin{aligned} K_t &= P_t^{-}H^T\left(HP_t^{-}H^T + R\right)^{-1} \\ \hat{x}_t^{+} &= \hat{x}_t^{-} + K_t\left(z_t - H\hat{x}_t^{-}\right) \\ P_t^{+} &= \left(I - K_tH\right)P_t^{-} \end{aligned} \tag{3}$$

where $P_t^{-}$ is the covariance matrix of the prior state $\hat{x}_t^{+}$.

## 2.3. Nonlinear Optimization Theory

In lidar SLAM systems, the state estimation problem is typically modeled as a nonlinear optimization problem. This arises because the robot's motion models and observation models often exhibit nonlinear characteristics, and sensor data inevitably contains noise. To achieve high-precision pose estimation, a robust optimization framework based on maximum a posteriori probability estimation is essential. Specifically, by modeling the state space on Lie group manifolds, the pose estimation problem is transformed into a constrained optimization problem with strong nonlinear properties. At this point, the nonlinear least-squares optimization framework, constructed based on the minimization criterion of the sum of squared residuals under Mahalanobis distance metrics, becomes a key mathematical tool in factor graph optimization. Nonlinear optimization methods offer superior accuracy compared to filtering methods, as they can leverage data from all time instants. Common approaches include gradient descent [22], Gauss-Newton [23], and Levenberg-Marquardt (L-M) methods [24].

Gradient descent is the most fundamental first-order optimization method. Its core idea is to iteratively update the state variables along the negative gradient direction of the objective function to gradually approach a local minimum. The goal is to minimize the error function $E(x)$, and the basic update rule for gradi-

ent descent is:

$$x_{k+1} = x_k - \alpha \nabla E\left(x_k\right) \tag{4}$$

where $x_k$ represents the parameter vector at the $k$-th iteration; $\alpha$ is the learning rate, Gradient descent iteratively updates the parameter vector $x$ until the value of the error function converges to a minimum.

The Gauss-Newton method is an improved variant of gradient descent, suitable for fitting nonlinear least-squares problems. In SLAM, it is commonly used to optimize error functions in graph optimization. Assuming the error function $E\left(x\right)$ is the sum of squared residuals, defined as:

$$E(x) = \sum_{i=1}^{m} r_i\left(x\right)^2 \tag{5}$$

where $r_i$ are the residuals. The update formula for the Gauss-Newton method is:

$$x_{k+1} = x_k - \left(J_k^T J_k\right)^{-1} J_k^T r\left(x_k\right) \tag{6}$$

where $J_k$ is the Jacobian matrix and $r_{(x_k)}$ is the residual vector. The core idea of the Gauss-Newton method is to linearize the residual functions and minimize the error through iteration.

The Levenberg-Marquardt method is an algorithm that combines the advantages of gradient descent and Gauss-Newton methods. It can accelerate the optimization process when gradient descent converges slowly and provides stable convergence when Gauss-Newton is unstable. The update formula for the Levenberg-Marquardt method is:

$$x_{k+1} = x_k - \left(J_k^T J_k + \lambda_k I\right)^{-1} J_k^T r(x_k) \tag{7}$$

where $\lambda_k$ is a damping factor that controls the weighting between gradient descent and Gauss-Newton, and $I$ is the identity matrix. The L-M method dynamically adjusts the damping factor to behave robustly like gradient descent when far from the optimal solution and converge quickly like Gauss-Newton when near it, making it one of the most commonly used and robust optimization algorithms for solving large-scale nonlinear least-squares problems in SLAM.

## 3. Algorithm Framework and Overview

### 3.1. System Framework

Building upon the LIO-SAM framework, this system proposes a multi-feature fusion mechanism centered on intensity gradients, along with a dynamic degradation detection feedback loop, to construct an adaptively optimized laser-inertial SLAM system. The overall framework is illustrated in **Figure 1**. The system employs a hierarchical processing architecture, initially performing motion distortion correction on the input lidar point clouds, followed by the feature extraction stage. Unlike traditional methods, this system synchronously computes intensity gradient features during feature extraction and performs weighted fusion with geometric fea-

tures. Concurrently, the system constructs a dynamic degradation assessment model by real-time analysis of multi-dimensional statistical characteristics of the point clouds, generating quantitative indicators of environmental degradation levels. In the pose estimation stage, the system adaptively adjusts the iteration strategies for scan matching, the down sampling granularity of feature points, and the keyframe selection criteria based on the current environmental degradation characteristics. This dynamic adjustment mechanism enables the system to maintain high computational efficiency in feature-rich structured environments, while allocating additional computational resources to sustain localization accuracy in feature-sparse degraded environments. The system establishes a closed-loop feedback pathway from back-end optimization to front-end features, achieving online optimization of system parameters by propagating the degradation assessment results of the current frame to the feature extraction module of the subsequent frame. This enables the system to proactively adapt to a wide range of complex scenarios, from highly structured environments to fully degraded ones.

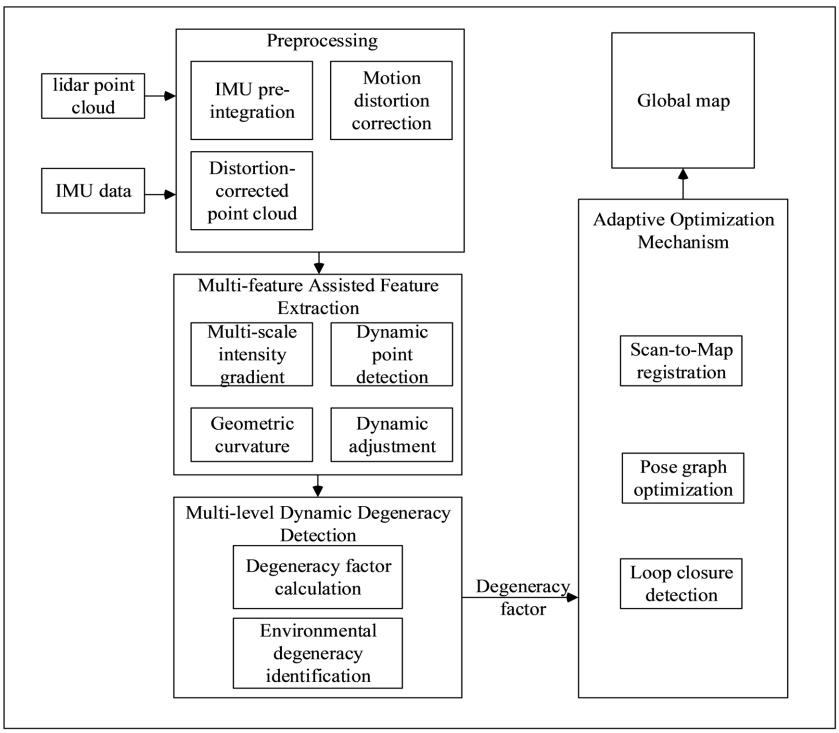



**Figure 1.** Algorithm flowchart.

### 3.2. Multi-Feature Assisted Feature Extraction

Feature extraction serves as the foundational front-end of the SLAM system, with its quality directly influencing the accuracy of subsequent pose estimation. Traditional methods rely solely on geometric curvature for feature selection, which limits its performance in scenarios with missing environmental features or severe dynamic interference. In view of this, a multi-feature assisted adaptive feature extraction method is proposed, which fuses intensity information with geometric

features to significantly enhance the system's robustness in complex environments.

First, a density-adaptive multi-scale intensity gradient calculation method is adopted, dynamically adjusting the neighborhood search range through point cloud density analysis. The neighborhood is defined using a k-nearest neighbors (k-NN) approach for flexibility in varying densities, or a fixed radius for uniform scenes. Density estimation is performed by computing the local point density as the number of points within a spherical volume, normalized by the volume. Two scales are employed: a fine scale with k = 5 or radius = 0.5 m to capture local details, and a coarse scale with k = 20 or radius = 2 m for global distributions. The intensity distribution gradient (IDG) is computed from the reflectance differences in the neighborhood point set:

$$X_{IDG} = \frac{1}{N}\sum_{j=1}^{N} |I_j - I_i| \tag{8}$$

where $N$ represents the number of points in the neighborhood, and $I_j$ and $I_i$ denote the intensity values of the neighborhood point and the center point, respectively.

To accommodate structural features of different scales, the system employs a multi-scale computational framework, calculating intensity gradients at fine and coarse scales to ensure both local details are captured and global distributions are perceived. To improve temporal stability, a Kalman filter is utilized for smoothing the intensity gradients. The Kalman filter state is defined as the IDG value, with process noise set to 0.01 (assuming low temporal variation) and measurement noise to 0.1 (accounting for sensor uncertainty); the filter updates recursively using standard prediction and correction steps as outlined in Section 2.2.

Addressing feature pollution in dynamic environments, a dynamic point discrimination criterion is established by analyzing the spatial distribution characteristics of intensity gradients combined with the distance continuity features of the point cloud. Prior to evaluation, IMU-based motion compensation is applied to undistort the point cloud, ensuring accurate distance computations by projecting points to a common timestamp using IMU-derived poses. Adjacent point distance variations are calculated in two ways: 1) along the laser scan line, where Euclidean distances between consecutive points on the same scan ring are computed to detect local discontinuities; and 2) via k-NN, where the average distance to the k nearest neighbors is assessed for broader context. A point is marked as dynamic and excluded if its intensity gradient exceeds an adaptive threshold and the distance variation is abnormal. To avoid erroneous removal of static points, the threshold incorporates a safety margin: removal occurs only when both conditions exceed predefined bounds, with $\lambda = 1.5$ to balance sensitivity and specificity, thus preserving static structures like walls or floors with gradual intensity changes. The dynamic point removal can be expressed as:

$$\mathrm{D}(p) = \prod \left[ IDG(p) > \mu_{IDG} + \kappa_{IDG} \cdot \sigma_{IDG} \right]^{\Pi\left[|\Delta r_p| > \epsilon_r\right]} \tag{9}$$

where $IDG_p$ is the intensity gradient value of the point, $\mu_{IDG}$ and $\sigma_{IDG}$ are the mean and standard deviation of the intensity gradients, $\kappa$ is the anomaly detection coefficient, $\Delta r$ is the distance difference between adjacent points, and $\epsilon$ is the relative distance threshold coefficient.

In terms of adaptive feature parameters, a comprehensive feature scoring method fusing geometric curvature and intensity gradients is designed. The intensity weight is dynamically adjusted based on the environmental degradation type, increasing the weight of intensity information in sparse scenarios to excavate more effective features, while relying more on geometric features in structured scenarios. The comprehensive feature score can be expressed as:

$$\begin{aligned} S(p) &= (1-w_{IDG})\cdot C(p) + w_{IDG}\cdot \frac{IDG(p)}{\max(IDG(p),\epsilon)} + \eta\cdot \mathrm{stab}(p) \\ w_{IDG} &= w_0 + \alpha\cdot FDI + \beta\cdot \delta_{type} \end{aligned} \tag{10}$$

where $C(p)$ and $\mathrm{IDG}(p)$ are the geometric curvature and intensity gradient values, respectively, $w_{IDG}$ is the dynamically adjusted intensity weight, FDI is the comprehensive degradation factor, and type is the degradation type.

The extraction thresholds for edge points and plane points are adaptively adjusted by incorporating local curvature complexity and degradation factors as regulatory parameters.

### 3.3. Multi-Level Dynamic Degeneracy Detection

Upon completion of feature extraction, the system conducts an in-depth analysis of environmental characteristics through a multi-level dynamic degradation detection module. This module is designed to quantitatively evaluate the degree of environmental degradation, providing precise decision-making support for the system's adaptive adjustments. By constructing a progressive analysis framework from basic metrics to comprehensive assessment, the system enables real-time perception and response to changes in environmental structural characteristics.

Degradation detection begins with the parallel computation of fundamental environmental metrics. The system characterizes environmental features from four complementary dimensions: the geometric complexity factor assesses local surface structural characteristics via principal component analysis, with calculations based on the eigenvalue distribution of the point cloud covariance matrix; the local curvature complexity reflects the intensity of surface variations, achieved by analyzing statistical features of curvature within neighborhoods; the point cloud entropy measures spatial distribution uncertainty, computed based on the uniformity of point cloud distribution in voxel space; and the depth plane deviation evaluates the deviation of the point cloud from an ideal plane, realized by calculating the average distance from points to the fitted plane. These fundamental metrics collectively form a first-order representation of environmental characteristics, offering a rich data foundation for subsequent analysis.

Building upon the computation of these fundamental metrics, the system innovatively introduces the concept of sub-degradation factors to characterize envi-

ronmental degradation from diverse perspectives. The structural degradation factor integrates the geometric complexity factor and depth plane deviation, primarily assessing the level of environmental structurization; the sparse degradation factor combines point cloud entropy and intensity gradient information, mainly reflecting the richness of environmental features; and the dynamic interference factor fuses local curvature complexity and intensity gradient variations, emphasizing the characterization of dynamic interference levels in the environment. Each sub-degradation factor is generated through nonlinear combinations, fully accounting for inter-metric interactions.

To achieve optimal fusion of these sub-degradation factors, the system employs a dynamic weight allocation mechanism based on information theory. By computing the information entropy and Kullback-Leibler (KL) divergence of each factor, the system evaluates their importance in describing the environment.

The information entropy is calculated as follows:

$$H_m = -\sum_{b=1}^{B} p_m(b) \log p_m(b) \tag{11}$$

where $B$ is the number of degradation factors, and $p_b$ is the normalized probability density of the factor in the $b$-th discrete interval.

The KL divergence follows the standard form:

$$D_{KL}(P \| Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} \tag{12}$$

where $P$ represents the actual distribution of the factor, and $Q$ is the reference distribution. The KL divergence-based weight allocation ensures that factors with greater information content receive higher weights during fusion, thereby enhancing the accuracy of the comprehensive assessment.

The final system-wide Full Degeneracy Index (FDI) is obtained through weighted fusion:

$$FDI = \sum_k w_k \cdot SubFactor_k \tag{13}$$

where the weight coefficients $w_k$ are dynamically computed using the aforementioned information-theoretic methods, and $SubFactor_k$ denotes the three sub-degradation factors described above. This multi-level fusion mechanism preserves the specificity of each sub-factor while achieving a unified quantification of environmental degradation degree.

Based on the comprehensive degradation factor and the distribution of sub-factors, the system further enables real-time classification of environmental types. By establishing reasonable threshold boundaries, environments are categorized into four typical types: structured, sparse, dynamic, and mixed. This classification result is deeply coupled with the feature extraction and optimization modules, forming a complete closed-loop adaptive control system.

### 3.4. Adaptive Optimization Mechanism

Upon obtaining a precise assessment of the environmental degradation degree,

the system converts perceptual information into execution strategies through an adaptive optimization mechanism, achieving intelligent adjustment across the entire process. This mechanism encompasses key stages from feature management to pose estimation, ensuring optimal performance under varying environmental conditions through dynamic parameter adjustments.

To provide a complete procedure for computing adaptive intensity weights and optimization weights, the following algorithmic steps are outlined in pseudocode form. The Full Degeneracy Index (FDI) is obtained via weighted fusion as per Equation (13), where sub-degradation factors are nonlinearly combined from fundamental metrics.

The optimization process begins with adaptive adjustment of feature management. The system dynamically regulates the granularity of feature downsampling based on the real-time computed environmental degradation factor. In feature-rich structured environments, the voxel size is appropriately increased to enhance computational efficiency; whereas in feature-sparse degraded environments, the voxel size is decreased to preserve more environmental details. This adaptive downsampling mechanism effectively balances the system's computational load while ensuring feature quality.

At the pose estimation level, the system performs in-depth optimization on the scan matching process. By establishing a correlation model between iteration counts and degradation degree, intelligent allocation of computational resources is realized. The number of iterations is increased when environmental degradation is severe to ensure convergence accuracy, and reduced in favorable environments to improve efficiency. This strategy can be expressed as:

$$N_{iter} = N_{base} \cdot \left(1 + \beta \cdot \sum w_k \cdot SubFactor_k\right) \tag{14}$$

where $N_{base}$ is the base iteration count, $\beta$ is the weighting coefficient, and $w_k$ are the fusion weights of the respective sub-factors. Meanwhile, the system dynamically adjusts constraint weights according to the environmental type, strengthening plane feature constraints in structured environments and prioritizing edge feature constraints in sparse environments, thereby aligning the optimization objective more closely with environmental characteristics.

In the factor graph optimization phase, the system introduces a degradation-aware noise model. By analyzing the impact of environmental degradation degree on measurement reliability, noise parameters for various observation factors are adaptively adjusted. For the odometry factor, its noise covariance matrix can be adjusted as:

$$\Sigma_{odo} = \Sigma_{base} \cdot (1 + \gamma \cdot FDI) \tag{15}$$

where $\Sigma_{base}$ is the base noise covariance, and $\gamma$ is the adaptive coefficient. This mechanism enables the system to appropriately relax constraint tolerances in degraded environments, preventing estimation bias due to over-optimization.

The keyframe management strategy is likewise integrated into the adaptive framework. The system dynamically adjusts the insertion frequency of keyframes

based on the rate of environmental change and feature stability. In regions with rapid environmental feature variations or severe degradation, keyframe density is increased to enhance the map's representational capability; in stable environments, the number of keyframes is reduced to control computational complexity. The keyframe selection criterion can be expressed as:

$$\begin{aligned} \Delta T &= \left\| t_i - t_j \right\| + \lambda \left\| r_i - r_j \right\| > \tau_{kf} \cdot \left(1 - \delta \cdot FDI\right) \\ \tau_{kf} &= \tau_0 \cdot \exp\left(-\nu \cdot FDI\right) + \Delta T_{\min} \end{aligned} \tag{16}$$

where $\Delta_{ij}$ is the relative pose change, $T_0$ is the base threshold, $\alpha$ is the degradation sensitivity coefficient used to control the rate of threshold decay, and $T_{\min}$ is the minimum offset. This adaptive mechanism ensures an optimal balance between map construction quality and efficiency.

The optimization results and degradation assessment information of the current frame are fed back in real time to the feature extraction module, forming a cross-frame collaborative optimization. Experimental results demonstrate that this optimization mechanism enables the system to maintain stable performance output across various complex scenarios, verifying its effectiveness and robustness.

## 4. Experimental Results Evaluation and Analysis

The KITTI dataset serves as a widely utilized benchmark in the autonomous driving domain, encompassing diverse scenarios with lidar, IMU, and GPS data. The SubT-MRS dataset not only incorporates all-weather environments but also includes various degraded settings, such as unstructured corridors, varying illumination conditions, and perceptual obstacles like smoke and dust. This study selects sequences 01, 04, 08, and 10 from the KITTI dataset, along with the UGV1 and UGV3 sequence scenarios from the SubT-MRS dataset. The testing platform is configured with an Intel i5-12500 processor, 16 GB of memory, and the Ubuntu 20.04 operating system, and comparisons are conducted against the A-LOAM, LeGO-LOAM, LIO-SAM, and FAST-LIO2 algorithms. To comprehensively evaluate the localization performance of the algorithms, this paper adopts the Absolute Trajectory Error (ATE) as the core evaluation metric, conducting a multi-dimensional analysis across statistics such as maximum (max), mean, median, minimum (min), root mean square error (RMSE) standard deviation (std), and sum of squared errors (SSE). Among these, the RMSE of the absolute trajectory error serves as a key indicator, effectively reflecting the pose accuracy and system robustness.

### Experimental Results Analysis

From the experimental results presented in **Table 1**, it is evident that the algorithm proposed in this paper achieves the optimal performance in most sequences. In sequence 01, which primarily features highways with high-speed motion, LeGO-LOAM exhibits a significant performance degradation, with a root mean square error (RMSE) reaching 230.63 m. The proposed algorithm yields an absolute trajectory error RMSE of 20.624 m, representing improvements in absolute trajectory accuracy of 18.3%, 91.1%, and 2.5% compared to A-LOAM, LeGO-LOAM, and

LIO-SAM, respectively.

**Table 1.** Comparison of absolute trajectory errors on the KITTI dataset (unit: m).

| Sequences | proposed algorithm | rmse | mean | median | max | min | std |
|---|---|---|---|---|---|---|---|
| 01 | A-LOAM | 25.228 | 22.507 | 24.017 | 53.109 | 1.795 | 11.396 |
| | LeGO-LOAM | 230.632 | 218.816 | 188.898 | 344.436 | 94.207 | 72.873 |
| | LIO-SAM | 21.142 | 18.428 | 19.322 | 59.719 | 0.649 | 10.362 |
| | OURS | **20.624** | **18.116** | **19.007** | **45.614** | **0.113** | **9.856** |
| 04 | A-LOAM | 0.572 | 0.527 | 0.455 | 1.525 | 0.283 | 0.223 |
| | LeGO-LOAM | 0.469 | 0.404 | 0.378 | 1.274 | **0.070** | 0.239 |
| | LIO-SAM | 0.307 | 0.275 | 0.253 | **0.952** | 0.126 | 0.137 |
| | OURS | **0.305** | **0.270** | **0.242** | 0.953 | 0.108 | **0.143** |
| 08 | A-LOAM | 33.975 | 32.117 | 33.479 | 64.889 | 2.783 | 11.081 |
| | LeGO-LOAM | 32.980 | 31.294 | 33.077 | 52.768 | 7.101 | 10.408 |
| | LIO-SAM | 11.995 | 11.187 | 11.570 | 24.436 | **0.502** | 4.330 |
| | OURS | **5.762** | **4.663** | **3.457** | **17.212** | 0.579 | **3.385** |
| 10 | A-LOAM | 4.536 | 4.094 | 3.472 | 9.575 | 2.183 | 1.953 |
| | LeGO-LOAM | 2.044 | 1.851 | 1.791 | **4.304** | **0.181** | **0.866** |
| | LIO-SAM | 1.838 | 1.556 | 1.258 | 5.298 | 0.438 | 0.978 |
| | OURS | **1.757** | **1.468** | **1.148** | 5.014 | 0.323 | 0.965 |

In sequence 08, which encompasses extensive urban roads and complex dynamic environments, posing higher demands on the robustness of SLAM systems, the experimental results show that the proposed algorithm achieves an RMSE of 5.76 m, with improvements of 83.0%, 82.5%, and 52.0% relative to A-LOAM, LeGO-LOAM, and LIO-SAM, respectively. This substantial enhancement validates the effectiveness of the dynamic degradation detection mechanism, enabling the system to perceive environmental degradation levels in real time and adaptively adjust optimization strategies, thereby maintaining stable pose estimation performance in complex scenarios.

In sequences 04 and 10, the absolute trajectory errors of all algorithms remain at relatively low levels, primarily due to the rich environmental features and simple motion patterns in these sequences. Nevertheless, the proposed algorithm still delivers the optimal accuracy, with RMSE values of 0.305 m in sequence 04 and 1.757 m in sequence 10, corresponding to improvements of 0.7% and 4.3% compared to LIO-SAM, respectively.

Figure 2 illustrates the trajectory comparison of various algorithms on KITTI dataset sequences 01 and 10, where the dashed line represents the ground truth (GT), and the curves denote the estimated trajectories from A-LOAM, LeGO-LOAM, LIO-SAM, and the proposed algorithm (ours). As shown in Figure 2(c) and Figure 2(d), the proposed algorithm's trajectory aligns most closely with the ground truth in the comparisons, indicating superior performance in these sequence scenarios.

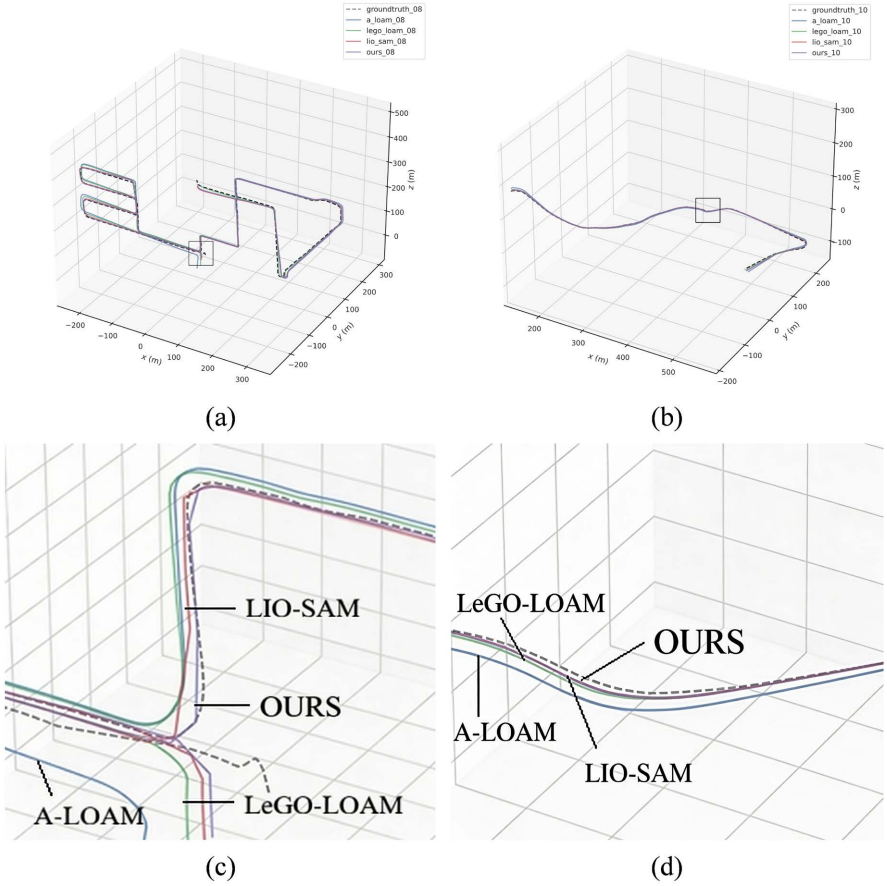



**Figure 2.** Comparison of trajectory results. (a) 08 Trajectory comparison; (b) 10 Trajectory comparison; (c) 08 detail comparison; (d) 10 detail comparison.

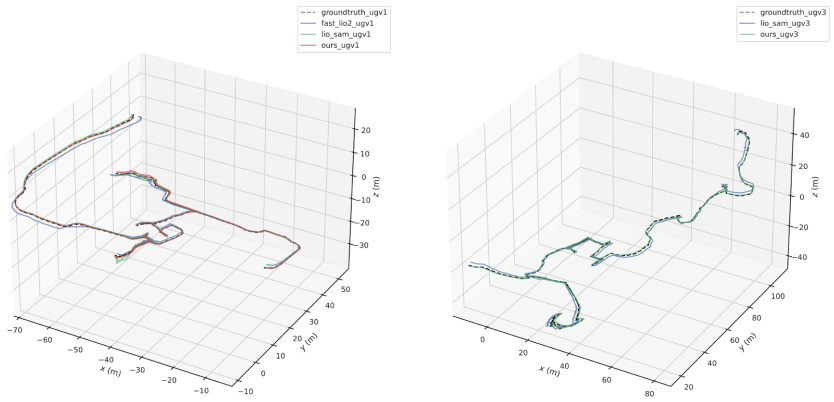
Compared to the outdoor road scenarios in the KITTI dataset, the UGV1 and UGV3 sequence scenarios in the SubT-MRS dataset primarily consist of geometrically degraded underground tunnel environments, characterized by sparse feature points, uniform wall textures, and potential dynamic degradation. As shown in the experimental results in **Table 2**, the proposed algorithm achieves an absolute trajectory error RMSE in the UGV1 sequence that represents improvements of 4.1% and 72.6% compared to LIO-SAM and Fast-LIO2, respectively. Notably, Fast-LIO2 exhibits a sum of squared errors (SSE) of 19,486 $m^2$, reflecting its matching failures in low-texture regions.

In the UGV3 sequence, Fast-LIO2 experiences severe drift, rendering its absolute trajectory error incalculable. LIO-SAM yields an absolute trajectory error of 1.252 m and an SSE of 1702 $m^2$. In contrast, the proposed algorithm achieves an absolute trajectory error of 0.738 m and an SSE of 618 $m^2$, corresponding to improvements in absolute trajectory accuracy and SSE of 41.0% and 63.7% relative to LIO-SAM, respectively. This demonstrates the proposed algorithm's significant advantages in handling complex degraded environments, effectively suppressing error accumulation and ensuring the accuracy and stability of trajectory estimation.

**Table 2.** Comparison of absolute trajectory errors on the SubT-MRS dataset (unit: m).

| Sequences | proposed algorithm | rmse | mean | median | max | min | std | sse ($m^2$) |
|---|---|---|---|---|---|---|---|---|
| UGV1 | FAST-LIO2 | 1.526 | 1.325 | 1.119 | 3.578 | 0.266 | 0.757 | 19,486 |
| | LIO-SAM | 0.563 | 0.428 | 0.388 | 3.799 | **0.021** | 0.364 | 244 |
| | OURS | **0.418** | **0.360** | **0.289** | **0.907** | 0.030 | **0.211** | **104** |
| UGV3 | FAST-LIO2 | — | — | — | — | — | — | — |
| | LIO-SAM | 1.251 | 1.173 | 1.155 | **2.799** | 0.161 | 0.434 | 1702 |
| | OURS | **0.738** | **0.612** | **0.513** | 4.276 | **0.072** | **0.412** | **618** |

In the UGV1 sequence of the SubT-MRS dataset (Figure 3(a)), the LIO-SAM trajectory exhibits a noticeable rightward deviation in the latter segment of the tunnel, whereas the proposed algorithm's trajectory (green) aligns closely with the ground truth (gray dashed line), showing virtually no deviation at bends. This indicates that the multi-feature-assisted feature extraction effectively compensates for matching deficiencies caused by geometric sparsity. In the UGV3 sequence (Figure 3(b)), the LIO-SAM trajectory displays an overall outward drift trend, with endpoint deviations reaching several meters, while the proposed algorithm's trajectory maintains a compact closure throughout the long-distance degraded corridor, achieving the highest degree of overlap with the ground truth and a 41.0% reduction in RMSE. This fully validates the anti-drift capabilities of the multi-level degradation detection and adaptive optimization in degraded environments.




(a) (b)

**Figure 3.** Comparison of trajectory results. (a) UGV1 Trajectory comparison; (b) UGV3 Trajectory comparison.

In summary, the experimental results validate the effectiveness of the multi-feature-assisted feature extraction, multi-level dynamic degradation detection, and adaptive optimization mechanisms proposed in this paper. In degraded environments characterized by feature scarcity or dynamic interference, the proposed algorithm significantly enhances system robustness through the fusion of intensity information, real-time environmental perception, and adaptive parameter adjustment.

## 5. Conclusion

To address the issues of mismatching and map overlap drift in simultaneous localization and mapping (SLAM) within degraded environments featuring sparse geometric features or dynamic interference, a multi-feature-assisted adaptive lidar-inertial SLAM system is proposed. Through the multi-feature-assisted feature extraction, multi-level dynamic degradation detection, and adaptive optimization mechanisms, experimental results on the KITTI dataset and SubT-MRS dataset demonstrate that, compared to other traditional algorithms, the proposed algorithm achieves improved pose accuracy, exhibiting superior localization precision and robustness. Future research may explore integrating the multi-feature-assisted mechanism into the loop closure detection module to further enhance the algorithm's localization accuracy and robustness.

## Funding

National Natural Science Foundation of China (Grant No. 62273239); Shanghai "Science and Technology Innovation Action Plan" Domestic Technology Cooperation Project (Project No. 20015801100).

## Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

## References

[1] Zou, Q., Sun, Q., Chen, L., Nie, B. and Li, Q. (2022) A Comparative Analysis of Lidar Slam-Based Indoor Navigation for Autonomous Vehicles. *IEEE Transactions on Intelligent Transportation Systems*, **23**, 6907-6921. https://doi.org/10.1109/tits.2021.3063477

[2] Xu, Z.H., Zhang, X., Fu, X.K., *et al.* (2024) Research Status and Application Progress of SLAM Technology in Mine Intelligence. *Journal of Anhui University of Science and Technology* (*Natural Science Edition*), **41**, 294-304.

[3] Yu, S.H., Wei, G.L., Zhang, S., *et al.* (2024) Laser SLAM Loop Closure Detection Method Fusing Multi-Source Information. *Information and Control*, **53**, 594-602.

[4] Li, S., Guan, H., Ma, X., Liu, H., Zhang, D., Wu, Z., *et al.* (2024) Laser-Inertial Tightly Coupled SLAM System for Indoor Degraded Environments. *Sensor Review*, **44**, 746-761. https://doi.org/10.1108/sr-07-2024-0615

[5] Li, Y., Xiong, Q., Li, Z., Shi, X. and Chi, W. (2024) A Unimodal Degradation Detection Method for Particle Filter-Based Slam Algorithms. *Procedia Computer Science*, **250**, 265-273. https://doi.org/10.1016/j.procs.2024.11.035

[6] Zeng, Y., Xu, F., Ming, J.H., *et al.* (2025) Continuous Motion Correction-Intensity Assisted Lidar Inertial Simultaneous Localization and Mapping in Degraded Environments. *Laser & Optoelectronics Progress*, **62**, 360-367.

[7] Zhang, J. and Singh, S. (2014) LOAM: Lidar Odometry and Mapping in Real-Time. *Robotics*: *Science and Systems X*, **2**, 1-9. https://doi.org/10.15607/rss.2014.x.007

[8] Shan, T. and Englot, B. (2018) LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain. 2018 *IEEE/RSJ International*

*Conference on Intelligent Robots and Systems* (*IROS*), Madrid, 1-5 October 2018, 4758-4765. https://doi.org/10.1109/iros.2018.8594299

[9] Shan, T., Englot, B., Meyers, D., Wang, W., Ratti, C. and Rus, D. (2020) LIO-SAM: Tightly-Coupled Lidar Inertial Odometry via Smoothing and Mapping. 2020 *IEEE/RSJ International Conference on Intelligent Robots and Systems* (*IROS*), Las Vegas, 24 October-24 January 2021, 5135-5142. https://doi.org/10.1109/iros45743.2020.9341176

[10] Qin, C., Ye, H., Pranata, C.E., Han, J., Zhang, S. and Liu, M. (2020) LINS: A Lidar-Inertial State Estimator for Robust and Efficient Navigation. 2020 *IEEE International Conference on Robotics and Automation* (*ICRA*), Paris, 31 May-31 August 2020, 8899-8906. https://doi.org/10.1109/icra40945.2020.9197567

[11] Xu, W. and Zhang, F. (2021) FAST-LIO: A Fast, Robust Lidar-Inertial Odometry Package by Tightly-Coupled Iterated Kalman Filter. *IEEE Robotics and Automation Letters*, **6**, 3317-3324. https://doi.org/10.1109/lra.2021.3064227

[12] Xu, W., Cai, Y., He, D., Lin, J. and Zhang, F. (2022) FAST-LIO2: Fast Direct Lidar-Inertial Odometry. *IEEE Transactions on Robotics*, **38**, 2053-2073. https://doi.org/10.1109/tro.2022.3141876

[13] Wang, H., Wang, C. and Xie, L. (2021) Intensity-Slam: Intensity Assisted Localization and Mapping for Large Scale Environment. *IEEE Robotics and Automation Letters*, **6**, 1715-1721. https://doi.org/10.1109/lra.2021.3059567

[14] Lin, J. and Zhang, F. (2022) R$^3$LIVE: A Robust, Real-Time, RGB-Colored, LiDAR-Inertial-Visual Tightly-Coupled State Estimation and Mapping Package. 2022 *International Conference on Robotics and Automation* (*ICRA*), Philadelphia, 23-27 May 2022, 10672-10678. https://doi.org/10.1109/icra46639.2022.9811935

[15] Geiger, A., Lenz, P., Stiller, C. and Urtasun, R. (2013) Vision Meets Robotics: The KITTI Dataset. *The International Journal of Robotics Research*, **32**, 1231-1237. https://doi.org/10.1177/0278364913491297

[16] Zhao, S., Gao, Y., Wu, T., Singh, D., Jiang, R., Sun, H., *et al.* (2024) SubT-MRS Dataset: Pushing SLAM Towards All-Weather Environments. 2024 *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), Seattle, 16-22 June 2024, 22647-22657. https://doi.org/10.1109/cvpr52733.2024.02137

[17] Gill, P.E. and Murray, W. (1978) Algorithms for the Solution of the Nonlinear Least-Squares Problem. *SIAM Journal on Numerical Analysis*, **15**, 977-992. https://doi.org/10.1137/0715063

[18] Grisetti, G., Strasdat, H., Konolige, K., *et al.* (2011) G$^2$o: A General Framework for Graph Optimization. 2011 *IEEE International Conference on Robotics and Automation*, Shanghai, 9-13 May 2011, 3607-3613.

[19] Agarwal, S. and Mierle, K. (2012) Ceres Solver: Tutorial & Reference. Google Inc.

[20] Dellaert, F. (2012) Factor Graphs and GTSAM: A Hands-On Introduction. Georgia Institute of Technology. Tech. Rep.

[21] Welch, G. and Bishop, G. (1995) An Introduction to the Kalman Filter.

[22] Shi, J.R., Wang, D., Shang, F.H., *et al.* (2021) Research Progress on Stochastic Gradient Descent Algorithm. *Acta Automatica Sinica*, **47**,2103-2119.

[23] Wang, Y. (2012) Gauss-Newton Method. *WIREs Computational Statistics*, **4**, 415-420. https://doi.org/10.1002/wics.1202

[24] Ranganathan, A. (2004) The Levenberg-Marquardt Algorithm. *Tutoral on LM Algorithm*, **11**, 101-110.