IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 8 July 2024, accepted 24 July 2024, date of publication 29 July 2024, date of current version 6 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3434578

APPLIED RESEARCH

Enhanced Mobile Robot Odometry With Error Kalman Filtering Incorporating 3D Point Cloud Intensity

XIONGWEN JIANG, TAIGA KUROIWA, HAOLAN ZHANG, TAKATO YOSHIDA, LINFENG SUN, YU CAO, HAOHAO ZHANG, TAKAHIRO KAWAGUCHI, (Member, IEEE), AND SEIJI HASHIMOTO, (Member, IEEE)

Division of Electronics and Informatics, Gunma University, Kiryu, Gunma 376-8515, Japan

Corresponding author: Seiji Hashimoto (hashimotos@gunma-u.ac.jp)

This work was supported in part by JSPS KAKENHI under Grant JP22K04150.

ABSTRACT Simultaneous localization and mapping (SLAM) based on light detection and ranging (LiDAR) is a trending research direction. This paper proposes a tightly coupled LiDAR inertial odometry (LIO) system that integrates geometric and textural information from the environment, aiming to address the key challenge of autonomous robot navigation. Most existing methods only consider geometric information and assume that reflective surfaces are ideal Lambertian surfaces, neglecting the variable reflective properties of surfaces at different angles and distances. Innovatively, we propose a new LiDAR intensity model that compensates for distortions in intensity values caused by incident angles and distances. Furthermore, we use intensity discontinuities as line features to reflect texture changes in the environment. These points are utilized to calculate residuals of line features, combined with surface features derived from geometric information, and integrated into the observation equations as part of an error-state Kalman filter sensor fusion scheme, thus constructing a tightly coupled LiDAR-inertial odometry system. Compared with the classic LOAM algorithm based on feature extraction and a state-of-the-art algorithm FAST-LIO, the results demonstrate that our system improves trajectory accuracy by approximately 16.1% with LOAM, 15.8% with LOAM+IMU and 11.2% with FAST LIO, rotation accuracy error precision by approximately 15.9% with LOAM, 15.3% with LOAM+IMU and 8.2% with FAST-LIO, demonstrate that proposed method significantly outperforms these methods in terms of localization accuracy and path generation precision.

INDEX TERMS Mobile robots, feature detection, sensor fusion, simultaneous localization and mapping (SLAM).

I. INTRODUCTION

Mobile robots have wide-ranging applications in fields such as logistics, catering, and industrial production. Reliable self-localization and mapping are crucial for the successful application of robots in various industries. As the robotics industry has evolved over the past few decades, robot localization has become increasingly challenging. The transition from familiar environments to unfamiliar ones, the evolution from simple scenarios to complex ones, the shift from static environments to dynamic ones, and the extension from short-term positioning to long-term positioning. Outdoor autonomous mobility usually relies on GPS systems. However, GPS measurements are more suitable for new area construction with wide views and fewer obstacles, field exploration positioning, etc. GPS systems have limitations in outdoor environments where signals may be weak or unavailable. For example in urban campuses outdoor scenes, GPS signals are often not received well in places blocked by buildings. These non-GPS-based odometry systems provide continuous and reliable positioning independent of satellite signals, making them particularly suitable for areas with signal obstruction or interference. Therefore, non-GPS-based odometers have significant advantages in outdoor scenarios with dense buildings.

In this context, Simultaneous Localization and Mapping (SLAM) systems have garnered great interest in the research community. SLAM technology, as a method for

The associate editor coordinating the review of this manuscript and approving it for publication was Li He.

pose estimation and localization in unknown environments, is widely used in mobile robotics [1], autonomous driving [2], and other fields [3]. Various SLAM systems based on sensors such as LiDAR, cameras, and Inertial Measurement Unit (IMU) has been developed. Depending on the perception system used, SLAM can be further categorized into Visual SLAM (V-SLAM) and LiDAR SLAM. Compared to V-SLAM, LiDAR SLAM is more accurate and robust in environmental changes such as weather and lighting. This paper focuses on the research of LiDAR SLAM.

Single-sensor systems are limited by assumptions about the robot's motion state, resulting in inaccuracies in 3D LiDAR self-localization and decreasing system accuracy. Existing robot self-localization systems must adhere to 6 degrees of freedom (DOF) constraints to provide accurate attitude estimates [4], including three translational degrees of freedom (along the x, y, and z axes) and three rotational degrees of freedom (about the row, pitch, and yaw axes). These constraints are crucial for describing and limiting the motion and attitude of objects in three-dimensional space. Accurate self-motion estimation is especially important when facing challenging environmental structures, such as complex terrain or obstacle layouts, aiding robots in better path planning, obstacle avoidance, and safe task completion. An Inertial Measurement Unit (IMU) is a combination sensor that provides information about object position, orientation, and movement by measuring acceleration, angular velocity and magnetic field direction. Combining LiDAR and IMU measurements offers low-drift self-motion estimation, particularly effective in scenarios with rapid movement, low light, and complex environments. Therefore, the combination of LiDAR and IMU not only enhances a robot's motion perception capabilities but also improves its adaptability and reliability in various engineering applications.

Traditional LiDAR-based SLAM methods heavily rely on prominent environmental features, potentially leading to reduced localization accuracy in environments lacking distinctive features. Assumptions about motion patterns constrain performance in fast or dynamic environments and exhibit sensitivity to LiDAR data quality and density, demanding significant computational resources and demonstrating limited adaptability in long-term or dynamic scenarios. Recent advancements in SLAM research have made significant expansions and improvements across several critical domains. In terms of accuracy, related studies address the limitations of single sensors through sensor fusion, integrating visual, inertial and GPS data to enhance localization accuracy and map quality. Efficiency-focused works have optimized algorithmic computational efficiency to achieve higher real-time performance, facilitating rapid adaptation to dynamic environments and large-scale data processing. Robustness improvements have concentrated on drift compensation and long-term consistency maintenance mechanisms within SLAM systems to meet the requirements of extended operational periods. Emphasis on cross-environment adaptability ensures stable performance across diverse and complex scenarios. These advancements enhance the potential applications of SLAM technology in fields like autonomous driving, robotic navigation, and augmented reality, promoting its widespread adoption and advancement in practical engineering and research.

However, the majority of these methods rely on the extraction of geometric features from LiDAR point cloud, which exhibit robustness and accuracy in structured environments. Nonetheless, in environments lacking structure, these methods may experience geometric degradation and failure. Traditional LiDAR SLAM methods primarily minimize discrepancies between point clouds by extracting the geometric three-dimensional coordinates of LiDAR points, often overlooking intensity channel information [5]. It is noteworthy that intensity information is correlated with material reflectivity, varying among different objects, and holds significance for localization and object recognition. Consequently, we advocate for LIO systems to consider this intensity information.

In the Lambertian model, LiDAR intensity calibration has been extensively researched in previous works. However, this model has several key limitations for LiDAR intensity calibration. It assumes surfaces to be ideal Lambertian surfaces, neglecting variations in surface reflectivity at different angles and inadequate adaptation to changes in distance and incident angles in the environment. These limitations imply that in practical applications, especially those requiring high precision and environmental adaptability, the Lambertian model may not provide sufficient accuracy and reliability.

In this paper, our approach introduces angle and distance factors to compensate for intensity attenuation due to incident angle and distance, thereby correcting variations in LiDAR intensity values. This correction method is applied to a line feature matching module to further utilize the corrected intensity information for enhancing the localization accuracy of the LIO system. We propose a novel LIO framework that integrates geometric and intensity information for odometry estimation. Specifically, we analyze the physical model of intensity measurements and introduce angle and distance factors to correct distortions in LiDAR intensity values caused by distance and incident angle. Subsequently, additional intensity line features are incorporated into the odometry estimation formula to complement existing geometry-based surface constraints. Finally, we employ an error state Kalman filter to fuse IMU state predictions with LiDAR state observations to enhance accuracy. The main contributions of this study are summarized as follows:

- A novel tightly-coupled framework for LIO system, aiming to leverage both texture and geometric information for localization estimation. This framework integrates intensity-based line features for texture information and surface features for geometric information.
- Establish an intensity correction model to rectify the distortion of LiDAR intensity values caused by incident angle and measurement distance.

TABLE 1. Related works in LiDAR-based SLAM and odometry.

No.ref	Intensity used	Year	Work	Description
[11]	N	2017	LOAM	Real-time 3D SLAM algorithm, low drift
[12]	N	2018	LeGO-LOAM	Lightweight LiDAR odometry and mapping optimized for changing terrain
[13]	N	2021	F-LOAM	Fast LiDAR odometry and mapping
[14]	N	2020	Loam livox	Fast, robust, high-precision odometry and mapping for small field-of-view LiDAR
[15]	N	2020	LINS	LiDAR-Inertial State Estimator for Robust and Efficient Navigation
[16]	N	2021	FAST-LIO	Fast and robust LiDAR-inertial odometry via a tightly coupled iterative Kalman filter
[17]	N	2022	Fast-LIO2	Fast Direct LiDAR-Inertial Odometry
[18]	N	2022	Faster-LIO	Using Parallel Sparse Incremental Voxels
[19]	N	2020	LIO-SAM	Tightly coupled LiDAR-inertial odometry via smoothing and mapping
[20]	Y	2022	An intensity-augmented lidar-inertial slam for solid-state lidars in degenerated environments	Enhanced LiDAR-Inertial SLAM for Solid-State LiDAR in Degraded Environments
[21]	Y	2020	Intensity scan context	Intensity scan context encoding intensity and geometry for loop closure detection
[22]	Y	2023	Intensity-based identification of reflective surfaces for occupancy grid map modification	Indoor navigation by analyzing the intensity values of laser scans to detect reflective surfaces
[23]	Y	2019	Intensity-assisted icp for fast registration of 2d-lidar	Intensity-assisted ICP for fast 2D lidar registration
[24]	Y	2015	Towards intensity-augmented slam with lidar and tof sensors	Towards intensity strength-enhanced SLAM using LiDAR and ToF sensors
[25]	Y	2016	Modeling laser intensities for simultaneous localization and mapping	Modeling laser intensity for SLAM and test with hector SLAM

- Conduct a thorough evaluation of the proposed method in outdoor scenarios. The results demonstrate that our approach achieves more accurate localization and generates precise paths in outdoor environments, outperforming existing geometric-only methods.

In Section I we introduce the widespread applications of mobile robots across various industries and emphasize the importance of precise self-localization and mapping for the successful deployment of these robots. In Section II we discuss the challenges of existing robotic localization techniques and provide an overview of the advancements in Simultaneous Localization and Mapping (SLAM) systems. Section III provides an in-depth review and comparative analysis of LiDAR-based SLAM, LiDAR-inertial sensor fusion, and the utilization of LiDAR intensity values. Section IV introduces our proposed Light Detection and Ranging (LiDAR) inertial odometry (LIO) system, detailing its design and implementation from LiDAR data acquisition through to error state Kalman filter-based state estimation. Section V presents experimental evaluations conducted to validate our system's performance in integrating geometric and textural information, showcasing quantitative results in various testing environments. Finally, Section VI summarizes our contributions, discusses limitations, and outlines future research directions, underscoring the potential of integrating LiDAR intensity data for enhanced robotic navigation and mapping capabilities.

II. RELATED WORKS

The existing research on LiDAR-based SLAM is extensive. Here, we confine our review to the most relevant works as visual SLAM, involving solely LiDAR SLAM, LiDAR-inertial sensor fusion and the application of LiDAR intensity values.

With the widespread adoption of 3D laser scanners and the advancement of embedded processor computing capabilities, localization technology based on 3D LiDAR has rapidly developed. Despite the recent proliferation of works on visual odometry due to advantages such as small camera size, lightweight, non-contact, and low cost, such as the ORB [6], [7], [8] series, SVO [9], VINS-Mono [10], compared to visual SLAM, LiDAR-based SLAM demonstrates greater accuracy and robustness in environmental changes such as weather and lighting.

The summary of mentioned LiDAR SLAM methods is shown in Table 1. Most current research on LiDAR SLAM focuses on the geometric information of the environment. LOAM [11], as the first real-time 3D SLAM algorithm, achieves low drift and computational complexity using only a laser scanner. Many scholars have made improvements based on this foundation. LeGO-LOAM [12] optimizes ground segmentation and optimization steps to fully utilize ground information. F-LOAM [13] achieves distortion compensation through a non-iterative two-step method, reducing computational costs while balancing algorithm accuracy and efficiency. Livox-LOAM [14] addresses the issue of feature point selection in solid-state laser scanners with limited field of view. These methods are all extensions of LOAM, utilizing a simple feature extraction strategy based on analyzing local smoothness to extract geometric feature points and segmenting feature points into edge features and planar features based on local smoothness.

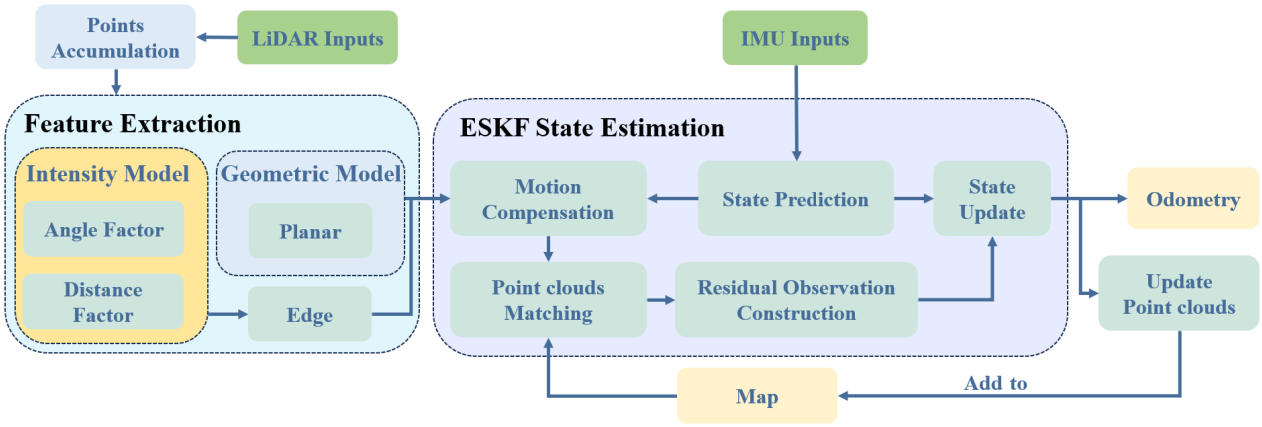


FIGURE 1. System Frameworks.

In recent years, many studies have proposed multi-sensor fusion solutions to address the challenges of real-time robustness and accuracy in SLAM algorithms. LINS [15] designed an Iterative Error State Kalman Filter (ESKF), which recursively corrects estimated states by generating new feature correspondences at each iteration, demonstrating outstanding robustness and accuracy while significantly improving real-time performance. FAST-LIO [16] utilizes a tightly-coupled Iterated Extended Kalman Filter to fuse LiDAR feature points with IMU data, while FAST-LIO2 [17] introduces an incremental k-d tree (ikd-Tree) on the basis of FAST-LIO, achieving superior real-time performance while downsampling on the naturally supporting tree. Faster-LIO [18] adopts incremental voxels (iVox) as the point cloud spatial data structure, improving search efficiency and achieving high-frequency point cloud output. LIO-SAM [19] integrates IMU pre-integration factors and GPS factors to achieve high-precision, real-time trajectory estimation, and map construction for mobile robots.

To enhance the performance and reliability of SLAM systems across various environments, there has been a growing body of work focusing on LiDAR intensity values. These studies employ diverse methods and technologies to improve LiDAR intensity values, aiming to enhance system performance and reliability. Reference [20] proposed a novel feature extraction method based on both geometry and intensity to fully utilize the extracted features. Two multi-weighted functions were designed for plane points and edge points separately during the pose optimization process to maintain both time and space efficiency and reduce edge intensity outliers in the line map. Reference [21] discussed the intensity attributes from LiDAR scans and demonstrated its effectiveness in position recognition. Reference [22] detected reflective surfaces by analyzing the intensity values of laser scans to construct a reflection-aware map for safe indoor robot navigation. Reference [23] introduced an objective function to determine the optimal initial transformation guess based on statistical data of spatial distance and intensity residuals, which is considered to have better accuracy and robustness than traditional ICP methods. Reference [24] attempted to explicitly incorporate the intensity model as part of the sparse bundle adjustment (SBA) estimation problem by combining LiDAR and time-of-flight (ToF) sensors. Observability analysis showed the existence of such a solution. Reference [25] focused on modeling the influence of external parameters on intensity to obtain a pose-invariant surface reflectance metric. These studies rely on an intrinsic surface property, namely reflectance, and external parameters such as distance to the surface and incidence angle with the surface normal.

III. OVERVIEW OF PROPOSED SYSTEM

Our system introduces an innovative tightly-coupled LIO framework that surpasses traditional odometry methods by integrating both geometric and textural environmental information. The key innovation is the use of corrected LiDAR intensity values, which provide crucial textural cues, enhancing the robot's ability to perceive its surroundings. This adjustment for intensity value distortion caused by incidence angle and distance enables more accurate feature detection and environmental mapping.

A distinguishing feature of our system is the employment of an Error-State Kalman Filter (ESKF) [26], offering a significant evolution over the standard Kalman Filter. The ESKF addresses errors in the state estimation process more directly by operating on the error states themselves, leading to enhanced robustness and accuracy. This is particularly beneficial in the presence of non-linearities and disturbances inherent in outdoor operations. Furthermore, our system's feature extraction process is innovatively divided, focusing on both intensity-based and surface feature extraction, which together contribute to a more nuanced and reliable odometry solution. The system's architecture is designed to adapt to dynamic changes in the environment, ensuring continuous and precise state estimation and map construction.

IV. FEATURE EXTRACTION

Feature extraction shown in Fig.1 is primarily divided into two parts: intensity-based feature extraction and surface feature extraction based on geometric information of LiDAR point cloud. In the intensity-based feature extraction part, the system adopts an advanced intensity model that considers the angle and distance factors in the LiDAR wave reflection process. This modeling method can more accurately describe the relationship between LiDAR intensity values and the incident angle of the LiDAR beam and the distance to the target object, thereby improving the recognition and classification ability of object surfaces. Through this intensity model, the system can identify feature points with significant intensity changes, which typically correspond to important structural information such as edges or corners of object surfaces.

A. INTENSITY MODEL

The intensity values of LiDAR can be regarded as discrete integer values, returned after a series of integrated processes, such as photoelectric signal conversion and amplification, subsequent calibration processing, and digital quantization of the optical power received by LiDAR from target echoes. To determine the external parameters affecting intensity characteristics, the LiDAR equation commonly used in remote sensing must be considered. Assuming a Lambertian [27] reflector, the LiDAR equation defines the relationship between the received optical power and the external parameters as shown in equation (1).

$$I_{\text{rec}} \propto P_{\text{rec}} \propto \frac{\varrho \cos(\alpha)}{r^2} \tag{1}$$

where ϱ represents the surface reflectance, r denotes the distance to the surface (radial coordinate/distance), and α corresponds 1to the angle of incidence. The proportionality between the received optical power P_{rec} and external parameters is influenced by additional constant parameters such as the system transmission factor, aperture diameter, and so forth.

The derivation of equation (1) treats the target as a Lambertian body. However, real targets often deviate from the Lambertian body, and during the intensity correction process of scanning LiDAR, its distance does not necessarily strictly follow the inverse square change law [28]. So the LiDAR equation can be rewritten as

$$I_{\text{rec}} = l\left(P_{\text{rec}}\right) = g(\varrho, \alpha) \cdot f(r). \tag{2}$$

where I_{rec} is the intensity value actually returned by the LiDAR, l is the signal conversion function to characterize the subsequent signal processing process of the LiDAR, $g(\varrho, \alpha)$ and $f(r)$ are the estimated data-driven functions defining the effect on intensities.

1) ANGEL FACTOR

To succinctly and accurately describe the spatial distribution of the diffuse reflection light from targets, we design a semi-ellipsoid model to fit the LiDAR intensity values as an angular factor. Specifically, the model represents a semi-ellipsoid with a center located at robot body frame, characterized by a major axis length of 2b and two minor axis lengths of 2a. Once the reflectivity ratio ϱ is a constant coefficient, the length of any point on the semi-ellipsoid is considered as the reflectivity distribution as shown in equation (3).

$$g(\rho, \alpha) = \rho_{\text{fse}} \sqrt{\frac{\eta_{\text{fse}}^2}{\sin^2 \alpha + \eta_{\text{fse}}^2 \cos^2 \alpha}}. \tag{3}$$

where η_{fse} represents the ratio of the semi-ellipsoid's minor axis to its major axis, defined as $\eta_{\text{fse}} = \frac{\text{a}}{\text{b}}$. ρ_{fse} is the reflectivity estimated by fitting equation (3) to the measured data of $g(\alpha)$. A good fit validation has been expected.

2) DISTANCE FACTOR

The distance factor is used to describe how the reflection intensity received by the LiDAR changes with the target distance. During the measurement process of liDAR, the intensity of the reflected signal will attenuate as the distance increases. This attenuation is usually inversely proportional to the square of the distance, but in actual applications it may be affected by a variety of factors, such as atmospheric absorption, scattering wait. Negative exponents better describe the trend of target intensity values as a function of distance. The distance factor can be obtained as

$$f(r) = \exp(2\sigma r) \tag{4}$$

where σ is the distance attenuation factor.

3) SIGNAL CONVERSION FUNCTION

By substituting equation (3) and (4) into equation (2), a detailed LiDAR intensity value equation can be obtained as

$$I_{\text{rec}} = \rho_{\text{fse}} \sqrt{\frac{\eta_{\text{fse}}^2}{\sin^2 \alpha + \eta_{\text{fse}}^2 \cos^2 \alpha}} \exp(2\sigma r) \tag{5}$$

where ρ_{fse} is the only unknown parameter can reflect the reflective characteristics of objects, thereby distinguishing texture-changing feature points.

B. GEOMETRIC MODEL

Geometric model fitting is used for extracting plane features, following approach to feature selection in LOAM, where roughness represents the geometric characteristics of points. Points are sorted based on roughness, and those with the smallest roughness are chosen as plane feature points. Roughness defined as

$$c = \frac{1}{|S| \, \|P_i\|} \left\| \sum_{j \in S, j \neq i} \left(P_i - P_j\right) \right\| \tag{6}$$

where S is the number of the point's neighborhood point set, and c is the roughness of the point and neighborhood point

set. P_i and P_j respectively represent the coordinate value of the point i and point j of the unified ID in a scan.

V. ESKF STATE ESTIMATION

This chapter mainly focuses on the ESKF state estimation part shown in Fig.1. Upon receiving information from the IMU, the system first estimates the current attitude, velocity, and IMU biases. In the state prediction phase, the system first utilizes the IMU state information to estimate the robot's state in the global map [29]. Then, the motion compensation is applied to the point cloud coordinates during the state prediction using the motion attitude information, to complete the feature extraction step. Subsequently, the motion state is applied to each feature point's timestamp to reproject the feature point cloud to the same moment, eliminating the distortion caused by the robot's motion in the LiDAR point clouds. Following this, the residual equation for point cloud registration is built through matching the feature point cloud, and the Kalman gain is computed to fuse the state prediction information from the IMU with the LiDAR observation residuals, completing the state estimation. Finally, the state and covariance matrix are updated to finish the state update.

A. STATE PREDICTION

This section focuses on predicting the motion state of the robot using the accelerometer and gyroscope data from the IMU. The symbols listed in Table 2 will be used throughout this paper.

1) CONTINUOUS MODEL

In continuous time, we represent the IMU readings as $\tilde{\omega}, \tilde{\boldsymbol{a}}$. Therefore, we can write the relationship between the derivative of the state variable and the IMU predicted value as

$$x = \begin{bmatrix} \dot{\boldsymbol{p}} \\ \dot{\boldsymbol{v}} \\ \dot{\boldsymbol{R}} \\ \dot{\omega}_b \\ \dot{\boldsymbol{a}}_b \\ \dot{\boldsymbol{g}} \end{bmatrix} = \begin{bmatrix} \hat{v} \\ \hat{R}\left(\tilde{\boldsymbol{a}} - \hat{a}_b\right) + g \\ \hat{R}\left[\tilde{\omega} - \hat{\omega}_b\right]_\wedge \\ \boldsymbol{b}_\omega \\ \boldsymbol{b}_a \\ \boldsymbol{0} \end{bmatrix} \tag{7}$$

The time derivative expressions of the translation, zero-bias, and gravity formulas of error variables can be easily obtained by taking the derivatives of both sides of the equations with respect to time. As for the velocity and rotation equations a continuous model of the error state can be obtained by utilizing the properties of matrix operations and neglecting the small quantities. After rearrangement, the kinematic equations of the error variables can be organized according to equation (8).

$$\delta x = \begin{bmatrix} \delta\dot{\boldsymbol{p}} \\ \delta\dot{\boldsymbol{v}} \\ \delta\dot{\theta} \\ \delta\dot{\omega}_b \\ \delta\dot{\boldsymbol{a}}_b \\ \delta\dot{\boldsymbol{g}} \end{bmatrix} = \begin{bmatrix} \delta v \\ -\boldsymbol{R}\left(\tilde{\boldsymbol{a}} - \boldsymbol{b}_a\right)_\wedge \delta\theta - \boldsymbol{R}\delta\boldsymbol{b}_a - \eta_a + \delta\boldsymbol{g} \\ -\left(\tilde{\omega} - \boldsymbol{b}_\omega\right)_\wedge \delta\theta - \delta\boldsymbol{b}_\omega - \eta_g \\ \eta_{b\omega} \\ \eta_{ba} \\ \boldsymbol{0} \end{bmatrix} \tag{8}$$

2) DISCRETE MODEL

Base on the continuous model, we can discrete the continuous model in equation (8) at IMU sample period Δt_i as shown in equation (9).

$$\begin{aligned} &\begin{bmatrix} \boldsymbol{p}(t+\Delta t_i) \\ \boldsymbol{v}(t+\Delta t_i) \\ \boldsymbol{R}(t+\Delta t_i) \\ \boldsymbol{b}_\omega(t+\Delta t_i) \\ \boldsymbol{b}_a(t+\Delta t_i) \\ \boldsymbol{g}(t+\Delta t_i) \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{p}(t) + \boldsymbol{v}\Delta t_i + \frac{1}{2}\left(\boldsymbol{R}\left(\tilde{\boldsymbol{a}} - \boldsymbol{b}_a\right)\right)\Delta t_i^2 + \frac{1}{2}\boldsymbol{g}\Delta t_i^2 \\ \boldsymbol{v}(t) + \boldsymbol{R}\left(\tilde{\boldsymbol{a}} - \boldsymbol{b}_a\right)\Delta t_i + \boldsymbol{g}\Delta t_i \\ \boldsymbol{R}(t)\exp\left(\left(\tilde{\omega} - \boldsymbol{b}_\omega\right)\Delta t_i\right) \\ \boldsymbol{b}_\omega \\ \boldsymbol{b}_a \\ \boldsymbol{g} \end{bmatrix} \end{aligned} \tag{9}$$

The discrete form of the error state equation only requires the addition of bias terms and gravity terms to the continuous form. Additionally, due to the special nature of the rotation matrix, the discrete form of the error state equation requires computation of the rotation part in the continuous form using the Rodrigues formula. The error state equation after rearrangement is obtained in Equation (10).

$$\begin{cases} \begin{bmatrix} \delta\boldsymbol{p}(t+\Delta t_i) \\ \delta\boldsymbol{v}(t+\Delta t_i) \\ \delta\theta(t+\Delta t_i) \\ \delta\boldsymbol{b}_\omega(t+\Delta t_i) \\ \delta\boldsymbol{b}_a(t+\Delta t_i) \\ \delta\boldsymbol{g}(t+\Delta t_i) \end{bmatrix} = F_x \begin{bmatrix} \delta\boldsymbol{p}(t) \\ \delta\boldsymbol{v}(t) \\ \delta\theta(t) \\ \delta\boldsymbol{b}_\omega \\ \delta\boldsymbol{b}_a \\ \delta\boldsymbol{g} \end{bmatrix} + F_n \\ F_x = \begin{bmatrix} I & \Delta t_i I & 0 & 0 & 0 & 0 \\ 0 & I & -\Delta t_i R\left(\tilde{a} - b_a\right)_\wedge & -R\Delta t_i & 0 & \Delta t_i I \\ 0 & 0 & exp\left(-\Delta t_i\left(\tilde{\omega} - b_g\right)\right) & 0 & -\Delta t_i I & 0 \\ 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & 0 & I \end{bmatrix} \\ F_n = \begin{bmatrix} 0 \\ \eta_v \\ -\eta_\theta \\ \eta_\omega \\ \eta_a \\ 0 \end{bmatrix}, \begin{array}{l} \sigma\left(\eta_v\right) = \Delta t_i \sigma_a, \\ \sigma\left(\eta_\theta\right) = \Delta t_i \sigma_\omega, \\ \sigma\left(\eta_\omega\right) = \sqrt{\Delta t_i}\sigma_{b\omega} \\ \sigma\left(\eta_a\right) = \sqrt{\Delta t_i}\sigma_{ba} \end{array} \end{cases} \tag{10}$$

In order to maintain the symbolic unity with ESKF, we express the motion equation in a linearized form and then perform the prediction process of ESKF. The prediction process includes the prediction of the nominal state (IMU integral) and the prediction of the error state covariance as shown in equation (11).

$$\begin{aligned} \delta\boldsymbol{x} &= \boldsymbol{F}(\delta\boldsymbol{x}) + \omega, \quad \omega \sim \mathcal{N}(0, \boldsymbol{Q}) \\ \delta\boldsymbol{x}_{\text{pred}} &= \boldsymbol{F}\delta\boldsymbol{x} \\ \boldsymbol{P}_{\text{pred}} &= \boldsymbol{F}\boldsymbol{P}\boldsymbol{F}^{\mathrm{T}} + \boldsymbol{Q} \\ \boldsymbol{Q} &= \text{diag}\left(\boldsymbol{0}_3, \text{Cov}\left(\eta_v\right), \text{Cov}\left(\eta_\theta\right), \text{Cov}\left(\eta_\omega\right),\right. \end{aligned}$$

TABLE 2. Variable and symbol definitions.

Symbol	Meaning
x	the true state of the ESKF
p, v	position, velocity vector
R	rotation matrix
$\boldsymbol{b}_a, \boldsymbol{b}_\omega$	accelerometer, gyroscope bias vector
g_t	gravity vector
δ	the error state of the parameter
$\wedge$	vector cross product
ω	gyroscope noise term
Q	Noise covariance matrix
F	Linearized Jacobian matrix
H	Jacobian matrix of the observation equation with respect to the error state
K	kalman gain
$\boldsymbol{v}$	observation noise
$\boldsymbol{P}_{\text{pred}}$	predicted covariance matrix
$\boldsymbol{P}$	updated covariance matrix

$$\text{Cov}\left(\eta_a\right), \mathbf{0}_3) \tag{11}$$

where noise terms do not participate in propagation and are grouped separately into the noise part. Continuous-time noise terms can be viewed as the energy spectral density of a random process, while discrete-time noise variables are random variables. The standard deviation of these noisy random variables follows a normal distribution.

So far, we have derived the process of IMU state prediction in ESKF, which corresponds to the state prediction equation in the Kalman filter. In order to ensure the convergence of the filter, LiDAR observations are introduced in the construction part of the residual equation to correct the Kalman filter.

B. MOTION COMPENSATION

During the process of a laser rangefinder following the movement of a robot, the robot's position varies at different times due to its own motion. Consequently, each laser point is generated at a different reference pose, which is the root cause of motion distortion. Laser scanning occurs concurrently with the robot's movement, and the laser data at each angle is not acquired instantaneously. The motion error of the laser frame induced by the robot's motion cannot be disregarded [30].

The IMU operates at a different frequency from the LiDAR, resulting in LiDAR points being processed collectively at fixed time intervals known as LiDAR frames. During the state prediction phase, the IMU effectively estimates the state, providing the nominal state at the beginning and end of each LiDAR frame. Timestamps for LiDAR points received in the current frame are recorded upon IMU data reception. Utilizing these timestamps, nominal states for individual LiDAR points are interpolated to establish homogeneous coordinate transformations relative to the frame's end [31]. Consequently, all points within the frame are projected to the frame's end moment based on their respective state transformations, compensating for structural distortions induced by robot motion.

C. POINT CLOUD MATCHING

In this section, nearest neighbor points cloud matching [32] is performed based on the previously extracted features. Edge points are matched with line features, while surface points are matched with surface features. Fig.2 illustrates the situation of point cloud matching, where different materials exist on the same plane. Different colors are used to distinguish different materials. During the environment scanning process, changes in reflectivity can provide information about the edges or contours of target objects. When reflectivity undergoes significant changes, it usually indicates that the LiDAR has scanned the edges or contours of objects. Therefore, by detecting points of reflectivity change, we can identify line features in the environment.

Traditional ICP algorithms rely solely on spatial positional information and may fail to recognize material changes and thus fail to identify line features resulting from material transitions. However, our method can extract implicit line constraints from surface features and extract effective edge and surface feature point information from complex point clouds. For each non-intensity mutation point, it can be considered to have the same point-to-surface matching relationship as geometric feature extraction. In subsequent sections, the distances from points to lines and from points to surfaces are used to construct error functions to solve the non-linear optimization problem of pose. The introduction of additional line feature constraints makes the system more accurate and robust.

The nearest neighbor matching for feature points is conducted through the ikd-tree algorithm [33]. The kd-tree algorithm is employed to search for nearest neighbors in the new point cloud, where a point will not be added if the Euclidean distance between it and its nearest neighbor is less than a set threshold. To enhance the algorithm's efficiency, we have replaced the traditional kd-tree with the ikd-tree algorithm. The ikd-tree algorithm offers a rapid approach for adding and removing points to the kd-tree, thus significantly improving the system's runtime speed.

Through this approach, we identify corresponding points in the source point cloud that share similar texture properties with points in the target point cloud, thereby aiding in establishing more accurate match pairs and reducing the number of iterations in subsequent residual calculations for nearest neighbor matching. Our method takes into account the texture attributes of points, implicitly providing richer information for aligning subsequent point clouds through intensity discontinuities. Consequently, we enhance the efficiency of feature point cloud matching in complex environments, particularly when dealing with targets with different materials and reflection properties.

D. RESIDUAL OBSERVATION EQUATION CONSTRUCTION

With the motion compensation in Section IV-B, we can view the scan of feature points p_k in the local frame, all sampled in the current timestamp, and its intensity feature k in the current

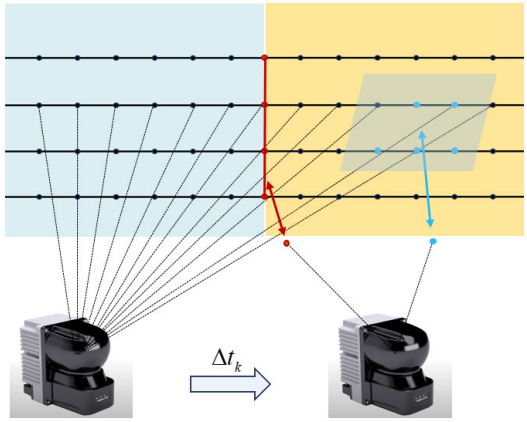


FIGURE 2. Point cloud matching between inter-frame.

timestamp. Since we used a LiDAR with built-in IMU, we need not consider the coordinate transformation between the LiDAR and IMU. Then, the points can be transformed to the global frame.

$$^{G}\mathbf{p}_j = {}^{G}\widehat{\mathbf{T}}_{L_k}{}^{L_k}\mathbf{p}_j; j = 1, \cdots, n. \tag{12}$$

where j means point clouds number in the current scan. L_k means in LiDAR frame and the time stamp is k.

For each LiDAR point in world frame, the closest point which has same intensity feature and the closest plane made by the map point clouds should be considered for construct the residual computation equation.

Due to the LiDAR measurement noise,each measured point ${}^{L}\mathbf{p}_j$ is typically contaminated by a measure-caused noise ${}^{L}\mathbf{n}_j$ considering the ranging and beam-directing noise. So considering this noise leads to the true point location in the local LiDAR coordinate frame ${}^{L}\mathbf{p}_j^{\text{gt}}$ as

$${}^{L}\mathbf{p}_j^{\text{gt}} = {}^{L}\mathbf{p}_j - {}^{L}\mathbf{n}_j. \tag{13}$$

The residual equation is defined as the point-line residual equation at the intensity value mutation point and the point-surface residual equation at the non-mutation point can be defined as

$$h(\boldsymbol{x}) = \begin{cases} \mathbf{u}_j^T \left({}^{G}p_{\text{plane}_j} - {}^{G}q_{\text{plane}_j} \right) \\ \mathbf{u}_j^{\wedge} ({}^{G}p_{\text{edge}_j} - {}^{G}q_{\text{edge}_j}) \end{cases} \tag{14}$$

where the distance from a point to a line is computed by the cross product of the normal vector $\mathbf{u}_j^{\wedge}$, which is an antisymmetric matrix, and the difference of the three-dimensional points. Leveraging the properties of vector cross product, the distance from a point to a line can be calculated using vector cross product. $\mathbf{u}_j^T$ is the form of residual measures the inner product between the point's distance from the plane and the point's projection vector onto the plane's normal vector. This inner product represents the difference between the projection of the point in the direction of the normal vector of the plane and the projection of the point in the direction of the normal vector of the plane, thereby measuring the distance from the point to the plane.

After projecting the real point into the world coordinate system, it should accurately lie on the plane (or edge) in the map, thus the residual equation should be zero. In other words, substituting equation (13) into equation (12), and then substituting into equation (14) should yield zero, shown in equation (15).

$$0 = h(\boldsymbol{x}) = \begin{cases} \mathbf{u}_j^T \left({}^{G}\widehat{\mathbf{T}}_{L_k} \left({}^{L_k}\mathbf{p}_{\text{plane}_j} - {}^{L}n_j \right) - {}^{G}q_j \right) \\ \mathbf{u}_j^{\wedge} \left({}^{G}\widehat{\mathbf{T}}_{L_k} \left({}^{L_k}\mathbf{p}_{\text{edge}_j} - {}^{L}n_j \right) - {}^{G}q_j \right) \end{cases} \tag{15}$$

E. ESKF STATE UPDATE

Based on equation (11), we perform the prediction process of ESKF. The prediction process includes prediction of the nominal state (IMU integration) and prediction of the error state. According to the IMU state transfer equation, the equations are organized into matrix form for ESKF state update.On this basis, perform the prediction process of ESKF LiDAR can observe state variables.The prediction process includes prediction of the nominal state (IMU integration) and prediction of the error state shown in equation (16).

$$\begin{aligned} z &= h(\boldsymbol{x}) + \boldsymbol{v}, \boldsymbol{v} \sim \mathcal{N}(0, \boldsymbol{V}) \\ \delta \boldsymbol{x}_{\text{pred}} &= \boldsymbol{F} \delta \boldsymbol{x} \\ \boldsymbol{P}_{\text{pred}} &= \boldsymbol{F} \boldsymbol{P} \boldsymbol{F}^{\mathrm{T}} + \boldsymbol{Q} \end{aligned} \tag{16}$$

In the ESKF framework, we currently possess an estimation of the nominal state alongside an estimation of the error state. To update the error state, it is imperative to compute the Jacobian matrix of the observation equation with respect to the error state, as shown in equation (17).

$$\boldsymbol{H} = \frac{\partial h}{\partial \boldsymbol{x}} \frac{\partial \boldsymbol{x}}{\partial \delta \boldsymbol{x}} \tag{17}$$

where base on the difference of residual type, $\frac{\partial \mathbf{h}}{\partial x}$ is different. When residual type is point-plane residual, $\frac{\partial \mathbf{h}}{\partial x} = u_j^T$ and if point-edge residual, $\frac{\partial \mathbf{h}}{\partial x} = \mathbf{u}_j^{\wedge}$. $\frac{\partial \boldsymbol{x}}{\partial \delta \boldsymbol{x}}$ is only related to the state equation's rotation and movement. According to the definition of the state variables established earlier, we can obtain $\frac{\partial \boldsymbol{x}}{\partial \delta \boldsymbol{x}}$ as

$$\frac{\partial \boldsymbol{x}}{\partial \delta \boldsymbol{x}} = \operatorname{diag}\left(\boldsymbol{I}_3, \boldsymbol{I}_3, \frac{\partial \log(\boldsymbol{R}(\exp(\delta\theta)))}{\partial \delta\theta}, \boldsymbol{I}_3, \boldsymbol{I}_3, \boldsymbol{I}_3 \right) \tag{18}$$

So by equation (16), (17) and (18), we can calculate the Kalman gain and update the error statement according to equation (19).

$$\begin{aligned} \boldsymbol{K} &= \boldsymbol{P}_{\text{pred}} \boldsymbol{H}^{\mathrm{T}} \left(\boldsymbol{H} \boldsymbol{P}_{\text{pred}} \boldsymbol{H}^{\mathrm{T}} + \boldsymbol{V} \right)^{-1} \\ \delta \boldsymbol{x} &= \boldsymbol{K} (z - h(\boldsymbol{x})) \\ \boldsymbol{P} &= (\boldsymbol{I} - \boldsymbol{K} \boldsymbol{H}) \boldsymbol{P}_{\text{pred}} \end{aligned} \tag{19}$$

After the prediction and update processes, we corrected the estimation of error states. Next, it is only necessary to incorporate the error states into the nominal states and then reset the ESKF.

VI. EVALUATION AND RESULTS

Firstly, we provide a detailed introduction to the robot platform, including its hardware specifications and technical configuration. We describe the hardware components such as sensors, actuators, and computing units mounted on the robot platform, as well as its application scenarios and performance characteristics in different environments.

Subsequently, to evaluate our enhanced mobile robot odometry system, we conducted tests in outdoor scenarios. The campus environment, with its buildings of various textured materials and unknown dynamic obstacles, provided a suitable experimental setting. These tests were performed while the robot was in motion to ensure that the system's functionality was thoroughly evaluated.

Finally, LOAM and FAST-LIO were chosen as comparison benchmarks because they represent classic and modern technologies in the field of LiDAR odometry respectively. LOAM is known for its stability and accuracy in a variety of environments, while FAST-LIO features its efficient computational performance and tightly coupled sensor fusion strategy. Both algorithms are frequently used as benchmarks for evaluating new methods, having been tested on multiple public datasets and proven to be stable and reliable, providing an impartial basis for comparison. By comparing with these advanced algorithms, our study aims to demonstrate the significant improvements of the proposed method in terms of accuracy, efficiency, and environmental adaptability.

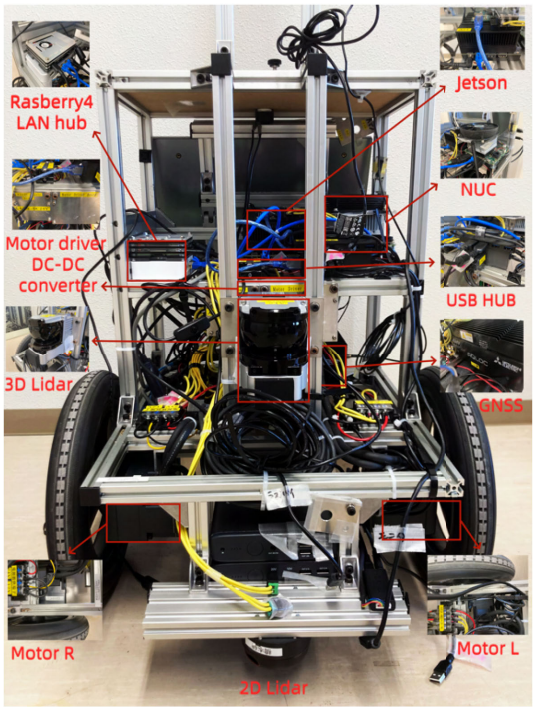


FIGURE 3. Robot Platform.

A. ROBOT PLATFORM

To validate the feasibility of the proposed algorithm, a robotic platform was constructed, contains all fundamental functions for ROS system development. The robot platform is equipped with advanced hardware components to achieve efficient autonomous navigation and environmental perception functions shown in Fig.3. The core computing units include NVIDIA Jetson series embedded boards and Raspberry Pi 4 single-board computers, responsible for processing machine learning and artificial intelligence algorithms. Intel NUC serves as another powerful computing resource, handling complex data in ROS [34] system processing tasks. Network communication is managed by a LAN hub, ensuring stable connections between all devices. Motor drivers control the robot's left and right motors (Motor R and Motor L), providing appropriate voltage supply through DC-DC converters. Additionally, a USB hub expands the connectivity ports, allowing access to more external devices. Environmental perception is achieved by 3D LiDAR, providing accurate distance measurements and environmental modeling, enabling the robot to effectively navigate obstacles and plan paths indoors and outdoors. The entire system is designed to enable the robot to maintain efficiency and accuracy while performing complex tasks.The specific hardware models used are shown in Table 3.

TABLE 3. Robot hardware list.

Platform	i-Cart Middle	T-frog Project
Motor	TF-M30-24	T-frog Project
Motor driver	TF-2MD3-R6	T-frog Project
Computer	NUC7	Intel
GPU	Jetson Xavier	NVIDIA
3D LiDAR	YVT-35LX-FK	Hokuyo3d
2D LiDAR	A2M8	RPLIDAR
AHRS	MTi-30-2A8G4	Xsens
Camera	C922	Logi
RasPi	4	RasPi Foundation
Arduino	Uno 3	Arduino

B. LIDAR FEATURE COMPENSATION

In the campus environment scene, common elements such as walls, ground, trees, and marble surfaces represent the diversity and complexity of outdoor environments. Their selection helps ensure the developed system's applicability and generalization ability in real-world conditions. Additionally, the reflectivity of these targets varies significantly, making them suitable for distinguishing intensity values.

The reliability of the feature extraction model was validated through feature value testing on objects of different distances and angles. The experiment consisted of two parts. The first part analyzed the reflection intensity values of targets at fixed distances and materials to control variables. The second part tested the reflection of the same target at different distances to verify if the model eliminated intensity value attenuation caused by distance.

The LiDAR incident angle is defined as 0 degrees when the LiDAR frontally enters, while the left and right scanning range is defined from -90 degrees to +90 degrees. In Fig.4 a), it can be observed that different materials may exhibit the

same LiDAR intensity at different incident angles, making it difficult to extract edge information when selecting line features. In Fig.4 b), it can be seen that LiDAR intensity values decay at different distances, and the intensity values of different materials may be the same at different distances.

After compensation by angle and distance factors, the distribution of intensity values is shown in Fig.5. Through this process, the LiDAR intensity values under different materials are clearly distinguished, thereby effectively capturing texture information from the environment.

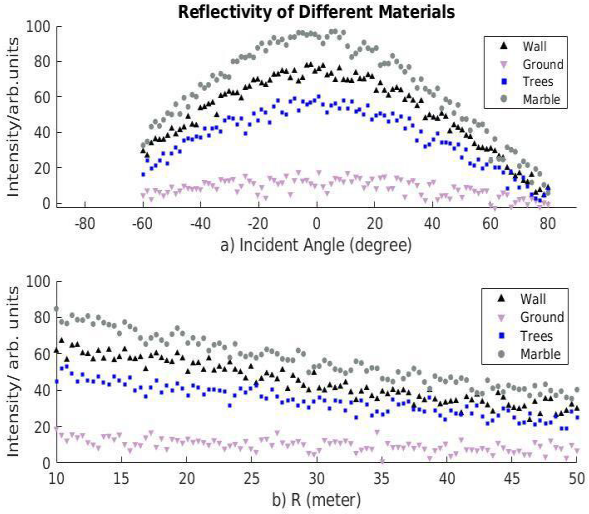


FIGURE 4. Intensity value distribution before compensation a) Distribution of LiDAR point cloud intensity values under different incident angles. b) Distribution of LiDAR point cloud intensity values at different distances.

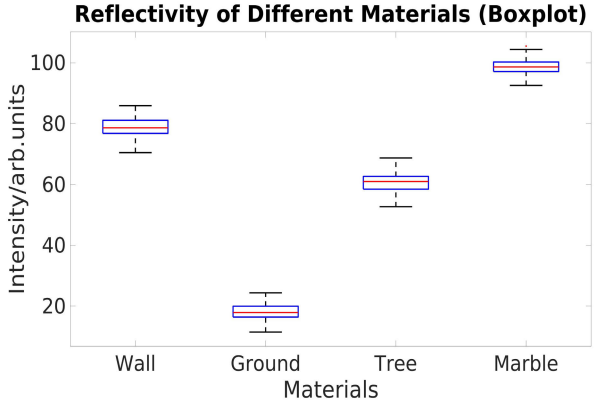


FIGURE 5. Intensity value distribution after compensation.

C. MAPPING RESULTS

The mapping results after intensity compensation correction are shown in Fig.6. The results show that the intensity-compensated mapped LiDAR point cloud can clearly distinguish the intensity values of different material surfaces. The processed intensity values accurately capture

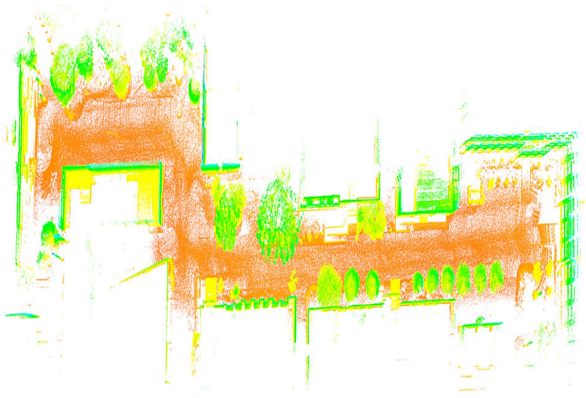


FIGURE 6. Mapping Results Colored with intensity.

and present the unique reflective properties of different materials. The mapping clearly reveals the changes in material and texture features in the environment and can reflect the reality in the actual scene.

By extracting the intensity change points from Fig.6, these feature points are shown in Fig.7. The points were strategically selected based on significant intensity changes. Unlike traditional line features that correspond to structural edges or corners in the environment, the distribution and selection of these points emphasize the algorithm's ability to discern important environmental cues. Specifically, areas with noticeable texture changes were used as part of the feature matching process, which avoided the instability caused by losing critical features in traditional feature extraction. By focusing on these feature points, the richness of map features is enhanced, providing a basis for subsequent positioning and trajectory tasks.

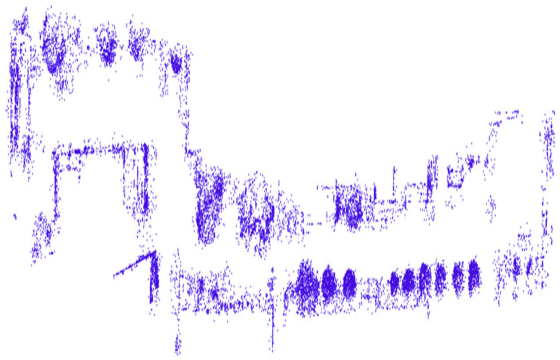


FIGURE 7. Intensity Feature Points.

D. MAPPING COMPARISON EXPERIMENTS

1) COMPARISON BEFORE AND AFTER INTENSITY VALUE CORRECTION

The comparison between the map results of using intensity change points before compensation and using corrected intensity change points for line feature matching is shown in Fig.8. It can be observed that the uncorrected line features

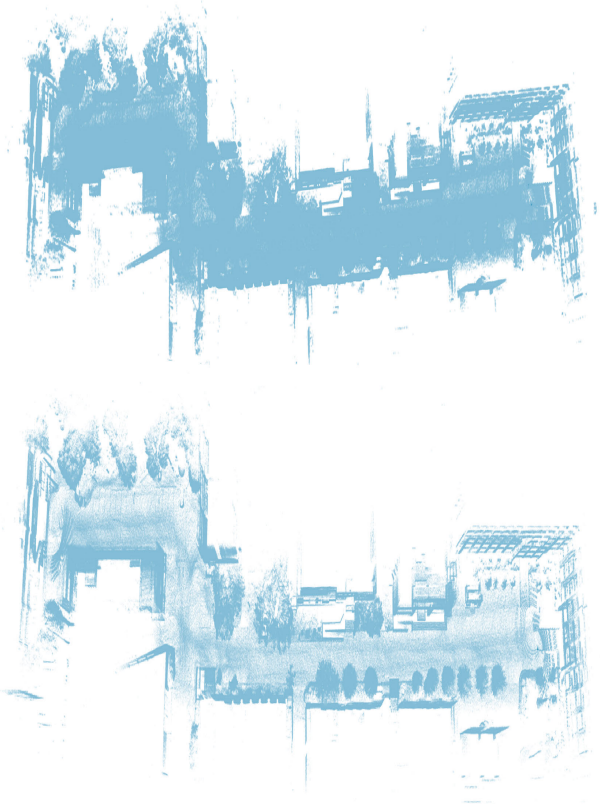


FIGURE 8. Mapping Comparison Result Before and After Intensity.

appear cluttered, resulting in inferior map outcomes with less distinct structural characteristics. The map results after correction exhibit clear structural features.

TABLE 4. Comparison results when output in 10HZ.

Method	Per LiDAR Scan		Total time	
	Feature points Num.	Running time (ms)	Avg.tra. (m)	Avg.rot. (m/deg)
LOAM	536	64	3.622	4.268
LOAM+IMU	536	55	2.784	3.541
FAST-LIO	671	**18**	0.365	0.433
Ours	**814**	21	**0.136**	**0.143**

2) COMPARISON WITH DIFFERENT ALGORITHM

The RTK GNSS trajectory is used as the ground truth reference. Specifically, we used identical raw sensor data (recorded in a rosbag) for testing to ensure each algorithm operated under consistent environmental conditions, enables a fair comparison of the performance differences among different algorithms when processing identical data. The recorded running times represent the average time each algorithm takes per frame. We recorded and computed these times while considering variations in the density of moving objects in the environment. Specifically, we logged the

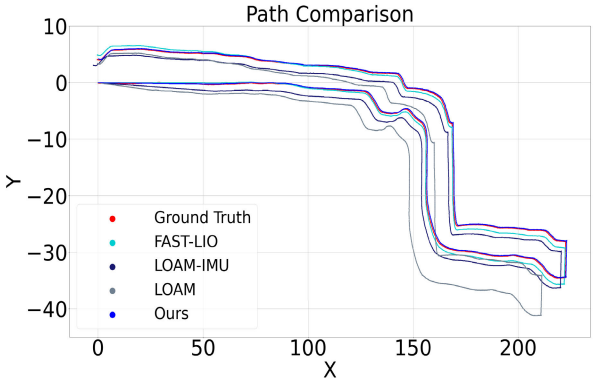


FIGURE 9. Path Comparison.

running time for each frame and calculated the average across all frames to reflect the overall performance of each algorithm.

After experimental evaluation, the quantitative results of performance indicators directly reflect the advantages of our algorithm in various key performance parameters shown in Table 4. In terms of average trajectory accuracy (Avg. tra.), our algorithm achieves an average trajectory error of 0.136 meters, demonstrating significantly lower error and highlighting its high accuracy. The average rotation error (Avg. rot.) of our algorithm is 0.143 degrees, indicating both high accuracy and stability in direction estimation. The number of feature points (Feature Points Num.) is 814, which reflects the efficiency of our algorithm in extracting environmental features and demonstrates its ability to accurately capture the richness of the environment. The running time is 21 milliseconds, slightly behind FAST-LIO which is renowned for its high processing speed.

The trajectories of all compared algorithms and our proposed algorithm are shown in Fig.9. The trajectory estimated by "Ours" is roughly consistent with the RTK GNSS trajectory and demonstrates the highest precision. "FAST-LIO" follows, while "LOAM" and "LOAM-IMU" exhibit noticeable deviations due to the loosely coupled structure and the lack of effective means to eliminate cumulative errors, resulting in erroneous final estimated trajectories. Therefore, the overall comparison demonstrates that our proposed algorithm achieves higher accuracy compared to classical open-source Lidar SLAM algorithms, validating the reliability of our method in terms of accuracy.

3) COMPARISON UNDER THE KITTI DATASET

Our algorithm was tested on the KITTI dataset sequence 05, an authoritative benchmark for autonomous driving and SLAM, as shown in Figure 10. Compared with LOAM and FAST-LIO, our algorithm has achieved significant improvements in trajectory accuracy, and its path is very close to the ground truth trajectory. The results of LOAM and FAST-LIO have some fluctuations or deviations, indicating that there may be positioning errors in some areas, resulting in large trajectory deviations.

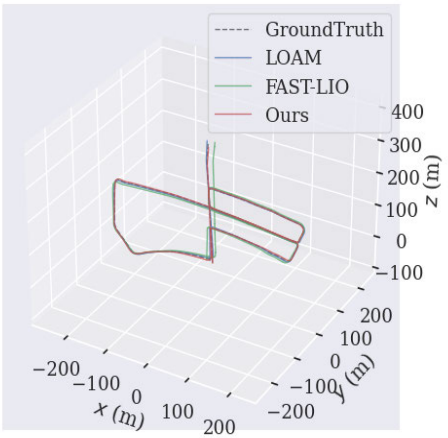


FIGURE 10. Trajectory Comparison.

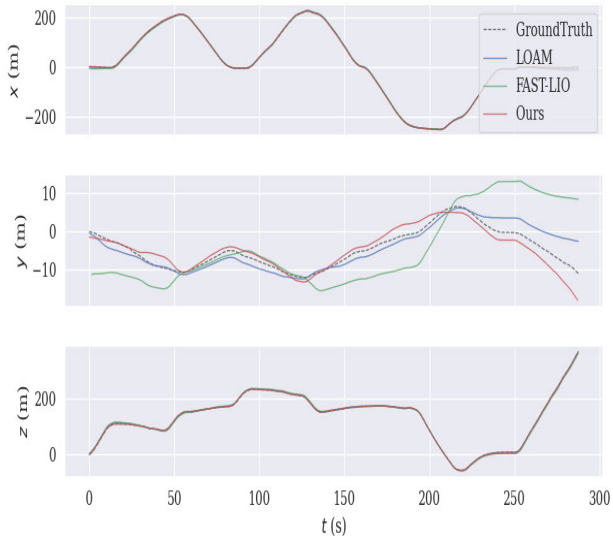


FIGURE 11. Trajectory comparison of xyz axis.

Fig.11 shows the stability of our algorithm in time series, maintaining consistent performance even in dynamic and unpredictable environmental changes. As time progresses, The trajectories of different algorithms on the y-axis begin to diverge. The trajectories of LOAM and FAST-LIO may exhibit significant deviations at certain time points, indicating potential accumulated errors in long-term localization or complex environments. The trajectory of our algorithm appears very close to the Ground Truth, demonstrating its high accuracy and reliability. Within the presented time frame, our algorithm shows better trajectory tracking performance compared to LOAM and FAST-LIO, maintaining closer alignment with the Ground Truth.

VII. CONCLUSION

This study proposed an innovative tightly-coupled LiDAR-inertial odometry (LIO) system, enhancing autonomous navigation capabilities by integrating geometric and textural environmental information. The proposed system addressed odometry drift in unknown outdoor environments, a common issue when relying solely on geometric cues. Key contributions included: Development of a novel LIO framework leveraging both geometric and intensity information from LiDAR data, improving localization accuracy and path generation. Establishment of an intensity correction model to rectify distortions in LiDAR intensity values caused by incidence angle and measurement distance, enhancing texture change recognition. Extensive experimental evaluations demonstrated superior performance over methods utilizing only geometric information. Benchmarking against state-of-the-art techniques, including LOAM, LOAM+IMU, and FAST-LIO, showing superior trajectory and rotation accuracy with competitive processing times. The results demonstrate that our system improves trajectory accuracy by approximately 16.1% with LOAM, 15.8% with LOAM+IMU and 11.2% with FAST LIO, rotation accuracy error precision by approximately 15.9% with LOAM, 15.3% with LOAM+IMU and 8.2% with FAST-LIO, demonstrate that our system significantly outperforms these methods in terms of localization accuracy and path generation precision.

However, limitations arise from using intensity discontinuities as line features, primarily influenced by the robot's speed and the density of dynamic obstacles in its vicinity. At high speeds, higher data update rate received by sensors can make the detection and tracking of intensity discontinuity features difficult. In environments characterized by dense and dynamic obstacles, such as areas with frequent movements of people or vehicles, the continuous scene changes can undermine the reliability of intensity-based features, thereby affecting feature alignment and odometry computation accuracy.

Future work will focus on improving dynamic obstacle modeling and handling techniques, especially in dense and highly mobile environments, is also essential. Exploring multi-sensor fusion strategies, such as integrating vision and GPS, to enhance the accuracy of position and orientation estimation in dynamic environments, helps overcome limitations posed by intensity discontinuities and enhances system robustness and reliability.

REFERENCES

[1] D. Di Paola, A. Milella, G. Cicirelli, and A. Distante, "An autonomous mobile robotic system for surveillance of indoor environments," *Int. J. Adv. Robotic Syst.*, vol. 7, no. 1, p. 8, Mar. 2010.
[2] I. Parra Alonso, R. Izquierdo Gonzalo, J. Alonso, Á. García-Morcillo, D. Fernández-Llorca, and M. Á. Sotelo, "The experience of DRIVERTIVE-DRIVERless cooperaTIve VEhicle-team in the 2016 GCDC," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 4, pp. 1322–1334, Apr. 2018.
[3] I. T. Lakmal, K. L. A. Nuwan Perera, H. A. Harindu Y. Sarathchandra, and C. Premachandra, "SLAM-based autonomous indoor navigation system for electric wheelchairs," in *Proc. Int. Conf. Image Process. Robot. (ICIP)*, Mar. 2020, pp. 1–6.
[4] J. Zhang, M. Kaess, and S. Singh, "On degeneracy of optimization-based state estimation problems," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 809–816.
[5] H. Wang, C. Wang, and L. Xie, "Intensity-SLAM: Intensity assisted localization and mapping for large scale environment," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 1715–1721, Apr. 2021.

[6] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[7] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[8] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021.

[9] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "SVO: Semidirect visual odometry for monocular and multicamera systems," *IEEE Trans. Robot.*, vol. 33, no. 2, pp. 249–265, Apr. 2017.

[10] T. Qin, P. Li, and S. Shen, "VINS-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

[11] J. Zhang and S. Singh, "Low-drift and real-time LiDAR odometry and mapping," *Auton. Robots*, vol. 41, no. 2, pp. 401–416, Feb. 2017.

[12] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and ground-optimized LiDAR odometry and mapping on variable terrain," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 4758–4765.

[13] H. Wang, C. Wang, C.-L. Chen, and L. Xie, "F-LOAM : Fast LiDAR odometry and mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 4390–4396.

[14] J. Lin and F. Zhang, "Loam livox: A fast, robust, high-precision LiDAR odometry and mapping package for LiDARs of small FoV," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 3126–3131.

[15] C. Qin, H. Ye, C. E. Pranata, J. Han, S. Zhang, and M. Liu, "LINS: A LiDAR-inertial state estimator for robust and efficient navigation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 8899–8906.

[16] W. Xu and F. Zhang, "FAST-LIO: A fast, robust LiDAR-inertial odometry package by tightly-coupled iterated Kalman filter," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 3317–3324, Apr. 2021.

[17] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "FAST-LIO2: Fast direct LiDAR-inertial odometry," *IEEE Trans. Robot.*, vol. 38, no. 4, pp. 2053–2073, Aug. 2022.

[18] C. Bai, T. Xiao, Y. Chen, H. Wang, F. Zhang, and X. Gao, "Faster-LIO: Lightweight tightly coupled LiDAR-inertial odometry using parallel sparse incremental voxels," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 4861–4868, Apr. 2022.

[19] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "LIO-SAM: Tightly-coupled LiDAR inertial odometry via smoothing and mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 5135–5142.

[20] H. Li, B. Tian, H. Shen, and J. Lu, "An intensity-augmented LiDAR-inertial SLAM for solid-state LiDARs in degenerated environments," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–10, 2022.

[21] H. Wang, C. Wang, and L. Xie, "Intensity scan context: Coding intensity and geometry relations for loop closure detection," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 2095–2101.

[22] A. Mora, A. Prados, P. González, L. Moreno, and R. Barber, "Intensity-based identification of reflective surfaces for occupancy grid map modification," *IEEE Access*, vol. 11, pp. 23517–23530, 2023.

[23] Y. Tian, X. Liu, L. Li, and W. Wang, "Intensity-assisted ICP for fast registration of 2D-LIDAR," *Sensors*, vol. 19, no. 9, p. 2124, May 2019.

[24] R. A. Hewitt and J. A. Marshall, "Towards intensity-augmented SLAM with LiDAR and ToF sensors," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 1956–1961.

[25] S. Khan, D. Wollherr, and M. Buss, "Modeling laser intensities for simultaneous localization and mapping," *IEEE Robot. Autom. Lett.*, vol. 1, no. 2, pp. 692–699, Jul. 2016.

[26] J. Solà, "Quaternion kinematics for the error-state Kalman filter," 2017, *arXiv:1711.02508*.

[27] X. Li, Y. Liang, and L. Xu, "Bidirectional reflectance distribution function based surface modeling of non-Lambertian using intensity data of light detection and ranging," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 31, no. 9, pp. 2055–2063, Aug. 2014.

[28] W. Zhen and S. Scherer, "Estimating the localizability in tunnel-like environments using LiDAR and UWB," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 4903–4908.

[29] G. P. C. Júnior, A. M. C. Rezende, V. R. F. Miranda, R. Fernandes, H. Azpúrua, A. A. Neto, G. Pessin, and G. M. Freitas, "EKF-LOAM: An adaptive fusion of LiDAR SLAM with wheel odometry and inertial data for confined spaces with few geometric features," *IEEE Trans. Automat. Sci. Eng.*, vol. 19, no. 3, pp. 1458–1471, Apr. 2022.

[30] K. Meng, H. Sun, J. Qi, and H. Wang, "Section-LIO: A high accuracy LiDAR-inertial odometry using undistorted sectional point," *IEEE Access*, vol. 11, pp. 144918–144927, 2023.

[31] L. Zhang, Q. Wang, S. Gu, T. Jiang, S. Jiang, J. Liu, S. Luo, and G. Yan, "Tightly-coupled SLAM integrating LiDAR and INS for unmanned vehicle navigation in campus environments," *IEEE Access*, vol. 12, pp. 26441–26456, 2024.

[32] A. V. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," in *Robotics: Sci. Syst.*, vol. 2, no. 4. Seattle, WA, USA, 2009, p. 435.

[33] Y. Cai, W. Xu, and F. Zhang, "ikd-Tree: An incremental K-D tree for robotic applications," 2021, *arXiv:2102.10808*.

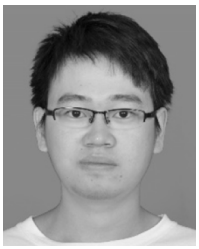
[34] M. Quigley. (2009). *ROS: An Open-Source Robot Operating System*. [Online]. Available: https://api.semanticscholar.org/CorpusID:6324125

XIONGWEN JIANG received the B.Sc. degree in electrical engineering from the Beifang University of Nationalities, Ningxia, China, in 2018, and the M.Sc. degree from Shenyang University of Chemical Technology, in 2021. He is currently pursuing the Ph.D. degree with Gunma University, Kiryu, Japan. His current research interests include SLAM, autonomous vehicle localization and mapping, indoor/outdoor seamless localization, and multi-sensor fusion localization technology.

TAIGA KUROIWA received the B.Sc. degree in electronic information mathematics from Gunma University, Kiryu, Gunma, Japan, where he is currently pursuing the M.Sc. degree. He is actively researching autonomous robots. His primary focus lies in the application of AI to enhance autonomy in robotics.

HAOLAN ZHANG received the B.Sc. degree in automation engineering from Dalian Polytechnic University, Dalian, China, in 2021. He is currently pursuing the M.Sc. degree with Gunma University, Kiryu, Japan. His current research interests include SLAM, computer vision, and 3D reconstruction. He is a Student Member of IEEJ.

TAKATO YOSHIDA received the B.Sc. degree in electronic information mathematics from Gunma University, Kiryu, Japan, in 2023, where he is currently pursuing the M.Sc. degree. His current research interest includes self-localization for mobile vehicles using nonlinear observers. He is a Student Member of IEEJ.



LINFENG SUN received the B.Sc. and M.Sc. degrees in electrical engineering from Yangzhou University, Yangzhou, China, in 2018 and 2021, respectively. He is currently pursuing the Ph.D. degree with Gunma University, Kiryu, Japan. His current research interests include digital control of power electronics, microgrids, PMSM motor drives, and AI-based control. He is a Student Member of IEEJ. He was honored with the Best Paper Award from the International Conference I-SEEC and ICTSS, in 2022, and the Outstanding Academic Master's Thesis Award from Yangzhou University, in 2022.

YU CAO was born in Yangzhou, China, in 1993. He received the B.S. degree in electrical engineering from Yangzhou University, Yangzhou, in 2017, and the M.S. and Ph.D. degrees from the Division of Electronics and Informatics, College of Science and Technology, Gunma University, Gunma, Japan, in 2020 and 2024, respectively. He is currently a Postdoctoral Fellow with the Faculty of Informatics, Gunma University.

HAOHAO ZHANG is currently an Associate Professor with the Division of Electronics and Informatics, Gunma University. Before joining Gunma University, he was an Application Engineer specializing in turbochargers with Honeywell Aerospace, from April 2016 to May 2019. He was the Robotics Software Engineer Leader with Ryomo Systems Company Ltd., from April 2022 to September 2023. His research primarily focuses on the development of driving methods for autonomous mobile robots using edge AI, and their implications for factory automation and social issues in Japan. He is a member of JIEP, RSJ, IEEA, and GTSJ.

TAKAHIRO KAWAGUCHI (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in engineering from Keio University, Tokyo, Japan, in 2011, 2013, and 2017, respectively. From 2013 to 2015, he was with the Toshiba Research and Development Center. From 2017 to 2019, he was a Researcher with the Department of Systems and Control Engineering, School of Engineering, Tokyo Institute of Technology, Tokyo. From 2019 to 2020, he was a specially appointed Assistant Professor with the Department of Systems and Control Engineering, Tokyo Institute of Technology. He is currently an Assistant Professor with the Division of Electronics and Informatics, School of Science and Technology, Gunma University, Gunma, Japan. His research interests include system identification theory and the application of machine learning techniques. He is a member of the Society of Instrument and Control Engineers and the Institute of System, Control, and Information Engineers.

SEIJI HASHIMOTO (Member, IEEE) received the M.E. and Ph.D. degrees in electrical and electronic engineering from Utsunomiya University, Japan, in 1996 and 1999, respectively. He joined the Department of Mechanical Engineering, Oyama National College of Technology. Since 2002, he has been a Research Associate with the Department of Electronic Engineering, Gunma University, where he is currently a Professor with the Program of Intelligence and Control. He has consulted for companies in control and energy applications and has been a Visiting Professor in China. He has more than 100 refereed articles and more than ten granted and pending patents. His research interests include system identification, motion control, AI-based control and diagnosis, and energy regeneration and its application to industrial fields. He is a member of IEEJ and SICE.