IEEE Robotics and Automation Letters (RAL) paper presented at
2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Enhancing VIO Robustness Under Sudden Lighting Variation: A Learning-Based IMU Dead-Reckoning for UAV Localization

Daolong Yang, Haoyuan Liu, Xueying Jin, Jiawei Chen, Chengcai Wang, Xilun Ding, and Kun Xu, *Member, IEEE*

***Abstract*—Visual Inertial Odometry (VIO) is commonly used for real-time Unmanned Aerial Vehicle (UAV) localization. However, the performance of VIO significantly deteriorates when UAV encounters sudden lighting variation in the environment, which poses a significant risk during flight. To address this issue without introducing additional sensors, a learning-based dead-reckoning algorithm relying solely on inertial measurement, which shares the same source with VIO, is proposed. The core idea of our method tightly couples a model-based Left Invariant Extended Kalman Filter (LIEKF) with a statistical neural network, both driven by raw inertial measurement. We have validated our algorithm for comparable accuracy with commonly deployed VIO methods under favorable lighting conditions and outperforms other IMU dead-reckoning algorithms in open-source datasets and real-world scenarios. To further enhance localization robustness while UAV traverses environments with different lighting conditions, we introduce an approach that tightly integrates our algorithm with VIO, and validate its effectiveness in real-world scenarios. It is believed that our work presents a promising way for enhancing robustness in vision-based localization methods within the robotics society.**

***Index Terms*—Aerial systems: perception and autonomy, localization, deep learning methods.**

I. Introduction

UNMANNED Aerial Vehicles (UAVs) are commonly employed aerial autonomous vehicles prevalently used in industries, exploration, and even space mission [1]. Accurate UAV localization is essential for the control strategy, as inaccuracies in positioning during high-speed flights can result in severe collisions [2]. In the absence of external positioning systems, Visual-Inertial Odometry (VIO) is commonly used for state estimation on UAVs [3], [4]. However, VIO methods suffer from degradation of accuracy in low-texture environments, and even fail in adverse lighting conditions (e.g. dark environments or rapid changes in light), posing a significant challenge for UAV localization. As illustrated in Fig. 1. Inertial odometry is an alternative solution to the problem of state estimation for UAVs, which is also insensitive to lighting variation in the environment. The onboard Inertial Measurement Unit (IMU) can measure the three-axis acceleration and angular velocity, and then the trajectory of the UAV can be estimated through dead-reckoning, ensuring reliable moving and navigation. In practical applications, IMU is susceptible to scale factor errors, axis misalignment errors, and time-varying biases [5]. Relying on inertial measurement from the IMU for UAV localization is a challenging issue.

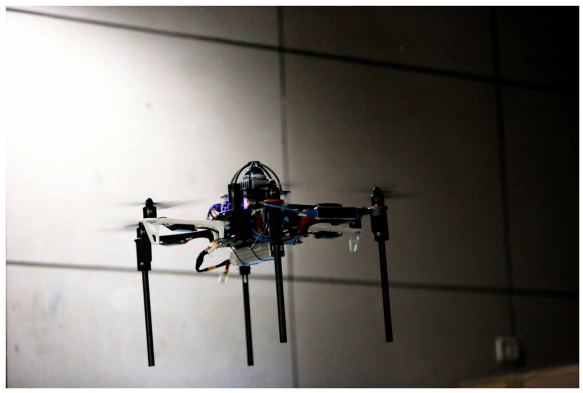

Fig. 1. Our hexacopter, as designed in [6], is transitioning from a well-lit environment to a dark area, where the accuracy of VIO degrades due to lack of visual features.

The most widely used filtering techniques in the field of IMU dead reckoning have long been the Bayesian filters, such as various extension of Kalman filters [7]. Similar to other filter methods, the Kalman filter relies on constant parameters that significantly affect filter performance and require careful calibration [8]. Recent advancements in data-driven approaches [9], [10], [11] have marked a significant breakthrough in IMU dead-reckoning, where IMU sensor data and ground-truth motion trajectories enables supervised learning of direct motion parameters. TLIO [12] integrates filtering techniques with neural networks, enhancing the overall algorithms accuracy and stability. The above-mentioned data-driven methods are all applied in pedestrian dead-reckoning. Unlike pedestrian, UAVs are more agile in space, resulting in rapid change in location and pose.

Manuscript received 29 October 2023; accepted 28 February 2024. Date of publication 18 March 2024; date of current version 4 April 2024. This letter was recommended for publication by Associate Editor K. Alexis and Editor G. Loianno upon evaluation of the reviewers' comments. This work was supported by the National Natural Science Foundation of China under Grant U22B2080 and Grant T2121003. *(Corresponding author: Kun Xu.)*

The authors are with the School of Mechanical Engineering and Automation, Beihang University, Beijing 102206, China (e-mail: YangDL@buaa.edu.cn; liuhaoyuan@buaa.edu.cn; xy_jin@buaa.edu.cn; chenjiawei@buaa.edu.cn; cc_wang@buaa.edu.cn; xlding@buaa.edu.cn; xk007@buaa.edu.cn).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2024.3377950, provided by the authors.

Digital Object Identifier 10.1109/LRA.2024.3377950

IMO [2] and Dido [13] combines IMU, quadrotor dynamics and deep learning methods to estimate the state of the quadrotor in both simulation and drone racing scenarios. Improved accuracy has been demonstrated over pedestrian dead-reckoning algorithms on their specific platform.

In this letter, we propose a novel IMU dead-reckoning algorithm inspired by TLIO [12] with specific modifications to adapt to UAV localization. We then take a further step by integrating the proposed algorithm with VIO system to enhance the robustness of localization under different lighting conditions.

The main contributions of this work are:

- An IMU dead-reckoning algorithm for UAV localization based on a Left Invariant Extended Kalman Filter (LIEKF) that is propagated by the inertial measurements and is updated by the position along with its corresponding uncertainty predicted by a Convolutional Neural Network (CNN). We name our algorithm *Learned Inertial Dead-Reckoning (LIDR)*.
- Validation of the proposed algorithm in micro aerial vehicle datasets EuRoc [14] and real-world scenarios. A comprehensive analysis of the proposed algorithm was conducted against VIO and state-of-the-art IMU dead-reckoning algorithms.
- An approach is introduced to tightly integrate our algorithm with the VIO system, and is validated in real-world scenarios to achieve robust localization when UAV encounters sudden lighting variation in the environment. We name our approach *Robust-VIO*.

II. Related Work

Visual-Inertial Odometry (VIO) is widely employed on UAVs for autonomous navigation due to its cost-efficiency and lightweight requirements, and there is a variety of solutions to this problem [15]. VINS (Visual-Inertial Navigation System) [3] is a VIO method that integrates IMU data into the Visual Odometry framework, has been adopted in swarms of UAVs to navigate through dense forests, showcasing high-accuracy localization performance [16]. Commercial-grade VIO algorithms integrated in the Intel RealSense T265 camera have been deployed on UAVs in real-time, showcasing robust state estimation [4], [17]. These VIO methods perform well under favorable lighting conditions but experience a significant decrease in accuracy under sudden lighting variation, e.g., transitions from the well-lit area to dark area. To achieve robust localization in such scenarios, the state of UAV can be estimated using the lighting-invariant inertial measurement from the onboard IMU. Previous work from [18] has designed a novel filter to estimate the orientation and velocity of a quadrotor from inertial measurement. This approach was further refined in a subsequent study [19], where they integrated a first-order drag model to estimate velocity through the drag force acting on the quadrotor and decoupled the attitude into a yaw-tilt convention to achieve better results. These works have made a breakthrough in Inertial Odometry for UAVs, with a primary focus on velocity and attitude rather than position. With the rapid advancement of Artificial Intelligence (AI) technology, many researchers are leveraging learning-based methods to achieve precise localization relying solely on inertial measurement from IMU. These approaches were initially conducted on pedestrian dead-reckoning. IONet [10] utilizes a Long Short-Term Memory (LSTM) based network, where the inputs consist of accelerometer and gyroscope measurements within a time window in the world coordinate system. RoNIN [9] introduces three distinct neural network architectures to address the inertial navigation challenge: LSTM, Temporal Convolutional Network (TCN), and Residual Network (ResNet). These models perform regression tasks to estimate the pedestrian's velocity and displacement in two dimensions. Moreover, many researchers have combined deep neural networks with filtering frameworks to achieve better results. TLIO [12] employs a ResNet network, similar to RoNIN [9], to predict pedestrian displacements excluding yaw information. It combines the network's output values with raw IMU measurements using a Stochastic Cloning Extended Kalman Filter (SCEKF) framework to estimate position, rotation, and sensor bias in a closely integrated way. Some works have migrated these pedestrian dead-reckoning algorithms to UAVs. IMO [2] introduce an innovative method for state estimation in autonomous drone racing, integrating a TCN for translational motion prediction and an Extended Kalman Filter (EKF) to estimate state variables using IMU measurements. DIDO [13] present an inertial and quadrotor dynamical odometry system that introduces deep neural networks within a two-stage tightly-coupled Extended Kalman Filter (EKF) framework. Both DIDO and IMO relying on additional measurements beyond the IMU, e.g. tachometer, which are often unavailable in real-time onboard UAV platforms.

III. Methodology

Our dead-reckoning algorithm (LIDR) exclusively takes raw inertial measurement from IMU as input and comprises two major components: *Filter Module* and *Learning Module*.

In the *Filter Module* we adopt the Left Invariant Extended Kalman Filter (LIEKF) as the state observer, which is propagated based on a kinematic motion model of the IMU and updated by outputs from the *Learning Module*. LIEKF has been first proposed in [20] and deployed as a inertial state observer in previous works [21], [22], which exhibits more accurate and stable performance over standard EKF thanks to its powerful local convergence guarantees [20]. Measurements like contact sensors and pressure sensors are utilized for LIEKF updates in previous works, which typically have known noise level. However, it is challenging to determine the noise level of the network in our case, especially considering that input features can vary significantly due to the high agility of UAVs. Therefore, it is crucial to allow the network to learn the noise level by itself.

In the *Learning Module*, we adopt a statistical neural network trained for regression tasks, predicting both the position of UAV $\mathrm{P}_{\mathrm{pred}}$ and the associated uncertainty $\mathrm{N}_{\mathrm{pred}}$ as the noise level of the network. Different from TLIO, we incorporate the previous position $\mathrm{P}_{\mathrm{prev}}$ as an additional input to learn the spatial feature, and collect IMU data in the body frame rather than the *gravity-aligned frame* in TLIO to learn the temporal motion pattern which is independent of the attitude. These adaptations are made

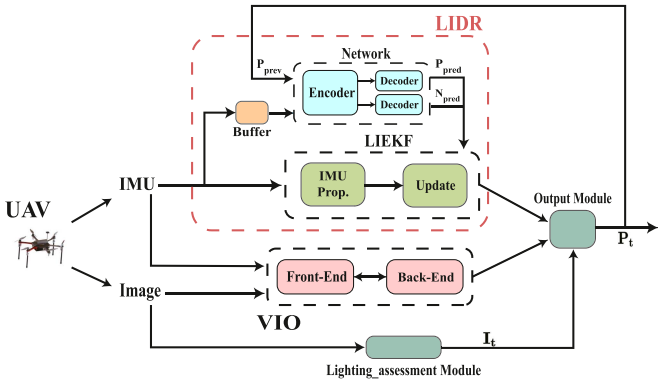

Fig. 2. Block diagram of our system. Our dead-reckoning algorithm (LIDR) is outlined by the red dashed line.

to accommodate the agile motion patterns of UAVs, which differ from pedestrians that typically have a gravity-aligned attitude and slower movement.

We further introduce an approach (Robust-VIO) to integrate our algorithm with VIO system. Our framework consists of two modules: the *Lighting_assessment Module* and the *Output Module*. In the *Lighting_assessment Module*, images from the camera are directly used to determine the current luminance level $\mathbf{I_t}$. In the *Output Module*, the accuracy of the VIO is evaluated based on $\mathbf{I_t}$ to determine whether to activate LIDR for localization.

The details of our work are presented below, and a block diagram of our system is shown in Fig. 2.

A. Learning Module

Our network employs a 1D version of the standard ResNet-18 architecture [23] as an encoder and adds two separate fully connected layers with 512 units at the end, serving as two separate decoders. We concatenate the IMU data within a time window of Δt and the previous UAV position to form input feature with dimensions of $(N \times 6 + 3)$, where $\mathrm{N} = \Delta \mathrm{t} \times IMU_{freq.}$. The output of the network contains two 3D vectors: the UAV position $\hat{\mathbf{p}}$ and the correspondence uncertainties $\hat{\mathbf{u}}$ which parametrize the diagonal entries of the measurement covariance matrix.

Two different loss functions are leveraged during training stage: the Mean Square Error (MSE) and the Gaussian Maximum Likelihood (ML).

The MSE loss is defined as:

$$\mathcal{L}_{MSE}(\mathbf{p}, \hat{\mathbf{p}}) = \frac{1}{n}\sum_{i=1}^{n} \left\|\mathbf{p_i} - \hat{\mathbf{p}}_\mathbf{i}\right\|^2, \tag{1}$$

where $\hat{\mathbf{p}} = \{\hat{\mathbf{p}}_\mathbf{i}\}_{i \leqslant n}$ are UAV position estimates output from the network and $\mathbf{p} = \{\mathbf{p_i}\}_{i \leqslant n}$ are the ground truth. n is the number of data in a single training batch.

The ML loss is defined based on the assumption that the probability of a label $y \in \mathbb{R}^k$ given a model input $x \in \mathcal{X}$ can be approximated by a multivariate Gaussian distribution [24]. In our training stage, we try to minimize the negative log-likelihood of the position estimates, so the ML loss is defined as:

$$\mathcal{L}_{ML}(\mathbf{p}, \hat{\mathbf{\Sigma}}, \hat{\mathbf{p}}) = \frac{1}{n}\sum_{i=1}^{n} -log\left(\frac{1}{\sqrt{(2\pi)^3 det(\hat{\mathbf{\Sigma}}_\mathbf{i})}} e^{-\frac{1}{2}\|\mathbf{p_i}-\hat{\mathbf{p}}_\mathbf{i}\|^2_{\hat{\mathbf{\Sigma}}_\mathbf{i}}}\right), \tag{2}$$

where $\hat{\mathbf{\Sigma}} = \{\hat{\mathbf{\Sigma}}_\mathbf{i}\}_{i \leqslant n}$ are the 3 covariance matrices for ith data. $\hat{\mathbf{\Sigma}}_\mathbf{i}$ has 6 degrees of freedom, and adopt the same strategy as [12], simply assuming a diagonal covariance output. Thus the equation from network uncertainty output vector $\hat{\mathbf{u}}$ to $\hat{\mathbf{\Sigma}}_\mathbf{i}$ can be written as:

$$\hat{\Sigma}_i(\hat{\mathbf{u}}_\mathbf{i}) = diag\left(e^{2\hat{u}_{ix}}, e^{2\hat{u}_{iy}}, e^{2\hat{u}_{iz}}\right). \tag{3}$$

B. Filter Module

1) Theoretical Background: A matrix Lie group [21] denoted $\mathcal{G}$ and its associated Lie Algebra denoted $\mathfrak{g}$. Define

$$(\cdot)^\wedge : \mathbb{R}^{dim\mathfrak{g}} \to \mathfrak{g}, \tag{4}$$

be the linear map that takes elements of the tangent space of $\mathcal{G}$ at the identity to the corresponding matrix representation, so we can write the exponential map

$$\exp(\xi) = \exp_m(\xi^\wedge), \tag{5}$$

where $\exp_m(\cdot)$ is the standard matrix exponential.

A process dynamics evolving on Lie group with state at time t, $X_t \in \mathcal{G}$ is denoted by

$$\frac{d}{dt}\mathbf{X_t} = f_{\mathbf{ut}}(\mathbf{X_t}), \tag{6}$$

and $\bar{\mathbf{X}}_\mathbf{t}$ is used to denote an estimate of the state. The state estimation error is defined using left multiplication of $\mathbf{X_t^{-1}}$ as follows.

Definition. (Left Invariant Error): The error between two trajectories $\bar{\mathbf{X}}_\mathbf{t}$ and $\mathbf{X_t}$ is

$$\eta^\mathbf{l}_\mathbf{t} = \mathbf{X_t^{-1}}\bar{\mathbf{X}}_\mathbf{t} = (\mathbf{L}\bar{\mathbf{X}}_\mathbf{t})^{-1}(\mathbf{L}\mathbf{X_t}), \quad \mathbf{L} \in \mathcal{G}. \tag{7}$$

The following theorems represent fundamental results for deriving an LIEKF.

Theorem 1: (Autonomous Error Dynamics [20]) A system is group affine if the dynamics $f_{\mathbf{ut}}(\cdot)$ satisfies: for all $t > 0$ and $\mathbf{X_1}, \mathbf{X_2} \in \mathcal{G}$

$$f_{\mathbf{ut}}(\mathbf{X_1X_2}) = f_{\mathbf{ut}}(\mathbf{X_1})\mathbf{X_2} + \mathbf{X_1}f_{\mathbf{ut}}(\mathbf{X_1}) - \mathbf{X_1}f_{\mathbf{ut}}(\mathbf{I_d})\mathbf{X_2}, \tag{8}$$

where $\mathbf{I_d}$ denotes the identity matrix. Furthermore, if this condition is satisfied, the left-invariant error dynamics are trajectory independent and satisfy

$$\frac{d}{dt}\eta^\mathbf{l}_\mathbf{t} = g^l_{\mathbf{ut}}(\eta^\mathbf{l}_\mathbf{t}), \text{where} \quad g^l_{\mathbf{ut}}(\boldsymbol{\eta}) = f_{\mathbf{ut}}(\eta) - f_{\mathbf{ut}}(\mathbf{I_d})\eta. \tag{9}$$

Theorem 2: (Log-linear Property of the Error [20]) Consider the left-invariant error η^l_t as defined by (7) between two trajectories. Let $\xi_0 \in \mathbb{R}^{dim\mathfrak{g}}$ be such that initially $\eta^l_0 = \exp(\xi_0)$. Define $\mathbf{A_t}$ to be a $dim\mathfrak{g} \times dim\mathfrak{g}$ matrix satisfying $g_{\mathbf{ut}}(\exp(\xi)) = (\mathbf{A_t}\xi)^\wedge + \mathcal{O}(\|\xi\|^2)$. For all $t \geqslant 0$, let ξ_t be the solution of the

linear differential equation

$$\frac{d}{dt}\xi_t = \mathbf{A_t}\xi_t. \tag{10}$$

Then, we have for the true nonlinear error η_t, the correspondence at all times and for arbitrarily large errors

$$\forall t \geq 0 \quad \eta_t^l = \exp(\xi_t). \tag{11}$$

This result shows a wide range of nonlinear errors can be exactly recovered from the time-varying linear error equations.

Theorem 3: (Stability Properties [20]) The LIEKF estimate $\bar{\mathbf{X}}_\mathbf{t}$ is an asymptotically stable observer of $\mathbf{X_t}$, and the convergence is valid over the whole trajectory.

The last theorem establishes that the LIEKF possesses local stability properties (i.e., capability to recover from perturbations or erroneous initializations), relying on error equation characteristics that the EKF does not possess.

2) State Representation: We construct the state in LIEKF based on the IMU model. The IMU measurement mainly consists of accelerometer and gyroscope where the accelerometer measures the three-axis accelerations $\tilde{a}_t$ and the gyroscope measures the three-axis angular velocities $\tilde{\omega}_t$. The measurements are modeled as being corrupted by additive white noises $\mathbf{w_t} \sim \mathcal{GP}(\mathbf{0}_{3,1}, \sum \delta(t-t'))$, where $\mathcal{GP}$ denotes a Gaussian process and $\delta(t-t')$ denotes the Dirac delta function [21].

Although the implementation of an IMU-based state estimator typically involves modeling IMU biases, which are slowly varying signals that corrupt the measurements in an additive manner. Unfortunately, there is no Lie group that includes the bias terms while also satisfying the dynamics that meet the group affine property outlined in Theorem 1, which affecting the local stability of LIEKF as discussed in Theorem 3. Additionally, given that MEMS IMUs on UAV often exhibit low in-run bias stability compared to high-grade IMUs, making it challenging to estimate precise biases at each step [25]. To address this issue, we treat the IMU biases as part of the Gaussian noise $\mathbf{w_t}$ on each propagation step and leverage LIEKF to converge the errors. In the experiment section, we will demonstrate that this modification does not lead to degradation of the overall algorithm.

Given the IMU model dynamics, we try to integrate the orientation, velocity, and position of the IMU in the world frame, denoted as $\mathbf{R_t}, \mathbf{v_t}, \mathbf{p_t}$, respectively. Together, these state variables form the group of double direct isometries, the matrix Lie Group $SE_2(3)$ [26]. An element $\mathbf{X_t} \in SE_2(3)$ is a 5×5 matrix represented by

$$\mathbf{X_t} \triangleq \begin{bmatrix} \mathbf{R_t} & \mathbf{v_t} & \mathbf{p_t} \\ \mathbf{0}_{1\times 3} & 1 & 0 \\ \mathbf{0}_{1\times 3} & 0 & 1 \end{bmatrix} \qquad \mathbf{u_t} \triangleq \begin{bmatrix} \tilde{\omega}_\mathbf{t} \\ \tilde{\mathbf{a}}_\mathbf{t} \end{bmatrix}, \tag{12}$$

where the input $\mathbf{u_t}$ is composed of measurements acquired from the IMU in the body (or IMU) frame. The Lie algebra of $SE_2(3)$, denoted by $\mathfrak{se}_2(3)$ is a 5 dimensional square matrix with 9 degrees of freedom. We use $(\cdot)^\wedge : \mathbb{R}^9 \to \mathfrak{g}$ to map a vector to the corresponding element of the Lie algebra, defined as follows.

Given $\xi \in \mathbb{R}^9$

$$\xi^\wedge = \begin{bmatrix} \xi_\mathbf{R} \\ \xi_\mathbf{v} \\ \xi_\mathbf{p} \end{bmatrix}^\wedge = \begin{bmatrix} (\xi_\mathbf{R})_\times & \xi_\mathbf{v} & \xi_\mathbf{p} \\ \mathbf{0}_{1\times 3} & 0 & 0 \\ \mathbf{0}_{1\times 3} & 0 & 0 \end{bmatrix}, \tag{13}$$

where $(\cdot)_\times$ denotes a 3×3 skew-symmetric matrix.

3) IMU Propagation: Expanding upon the previously introduced state representation, the IMU dynamics can be formally expressed in matrix form as follows:

$$\begin{aligned} \frac{d}{dt}\mathbf{X_t} = & \begin{bmatrix} \mathbf{R_t}(\tilde{\omega}_t)_\times & \mathbf{R_t}\tilde{\mathbf{a}}_\mathbf{t} + \mathbf{g} & \mathbf{v_t} \\ \mathbf{0_{1\times 3}} & 0 & 0 \\ \mathbf{0_{1\times 3}} & 0 & 0 \end{bmatrix} \\ & - \begin{bmatrix} \mathbf{R_t} & \mathbf{v_t} & \mathbf{p_t} \\ \mathbf{0_{1\times 3}} & 0 & 0 \\ \mathbf{0_{1\times 3}} & 0 & 0 \end{bmatrix} \begin{bmatrix} (\mathbf{w_t^g})_\times & \mathbf{w_t^a} & \mathbf{0_{3\times 1}} \\ \mathbf{0_{1\times 3}} & 0 & 0 \\ \mathbf{0_{1\times 3}} & 0 & 0 \end{bmatrix} \\ \triangleq & f_{\mathbf{u_t}}(\mathbf{X_t}) - \mathbf{X_t}\mathbf{w_t}^\wedge, \end{aligned} \tag{14}$$

where $\mathbf{w_t} = [(\mathbf{w_t^g})^T, (\mathbf{w_t^a})^T, \mathbf{0_{1\times 3}}]$. The deterministic system dynamics, $f_{\mathbf{ut}}(\cdot)$ can be shown to satisfy the group affine property (8). Therefore, in accordance with Theorem 1, the error dynamics will evolve independently of the system's state. Using (9), the left-invariant error dynamics is

$$\frac{d}{dt}\eta_\mathbf{t}^\mathbf{l} = g_{\mathbf{u_t}}^l(\eta_\mathbf{t}^\mathbf{l}) + \mathbf{w_t}^\wedge \eta_\mathbf{t}^\mathbf{l}, \tag{15}$$

where the second term arises from additive noise. The derivation follows the results in [20].

Additionally, Theorem 2 specifies that the invariant error adheres to a log-linear property. Namely, if $\mathbf{A_t}$ is defined, then the log of invariant error, $\xi \in \mathbb{R}^{dim\mathfrak{g}}$, approximately satisfies the linear system

$$\frac{d}{dt}\xi_t = \mathbf{A_t}\xi_t + \mathbf{w_t}. \tag{16}$$

To compute $\mathbf{A_t}$, we linearize the invariant error dynamics, $g_{\mathbf{ut}}^l(\cdot)$, using the first order approximation to yield

$$\begin{aligned} g_{\mathbf{u_t}}^l(\eta_\mathbf{t}^\mathbf{l}) &= g_{\mathbf{u_t}}^l\left(\mathbf{I_d} + \xi_t^\wedge\right) \\ &= f_{\mathbf{u_t}}\left(\mathbf{I_d} + \xi_t^\wedge\right) - f_{\mathbf{u_t}}(\mathbf{I_d})\left(\mathbf{I_d} + \xi_t^\wedge\right), \end{aligned} \tag{17}$$

$g_{\mathbf{u_t}}^l(\cdot)$ can be further written as:

$$g_{\mathbf{u_t}}^l(\eta_\mathbf{t}^\mathbf{l}) = \begin{bmatrix} -(\tilde{\omega}_\mathbf{t})_\times \xi_\mathbf{t}^\mathbf{R} \\ -(\tilde{\mathbf{a}}_\mathbf{t})_\times \xi_\mathbf{t}^\mathbf{R} - (\tilde{\omega}_\mathbf{t})_\times \xi_\mathbf{t}^\mathbf{v} \\ \xi_\mathbf{t}^\mathbf{v} - (\tilde{\omega}_\mathbf{t})_\times \xi_\mathbf{t}^\mathbf{p} \end{bmatrix}^\wedge. \tag{18}$$

With the above, the prediction step of LIEKF can be written as follow. The state estimate, $\bar{X}_t$, is propagated though the deterministic system dynamics, while the covariance matrix, P_t, is computed using Riccati equation [27], namely,

$$\frac{d}{dt}\bar{\mathbf{X}}_\mathbf{t} = f_{\mathbf{u_t}}(\bar{\mathbf{X}}_\mathbf{t}) \text{and} \frac{d}{dt}\mathbf{P_t} = \mathbf{A_t}\mathbf{P_t} + \mathbf{P_t}\mathbf{A_t}^T + \bar{\mathbf{Q}}_\mathbf{t}, \tag{19}$$

where the matrices $\mathbf{A_t}$ and $\bar{\mathbf{Q}}_\mathbf{t}$ are obtained from (18) and (16)

$$\mathbf{A_t} = \begin{bmatrix} -(\tilde{\omega}_\mathbf{t})_\times & \mathbf{0_{3\times3}} & \mathbf{0_{3\times3}} \\ -(\tilde{\mathbf{a}}_\mathbf{t})_\times & -(\tilde{\omega}_\mathbf{t})_\times & \mathbf{0_{3\times3}} \\ \mathbf{0_{3\times3}} & \mathbf{I_{3\times3}} & -(\tilde{\omega}_\mathbf{t})_\times \end{bmatrix}, \bar{\mathbf{Q}}_\mathbf{t} = \mathrm{Cov}[\mathbf{w_t}]. \quad (20)$$

The continuous dynamics can be discretized by assuming a zero-order hold on the inertial measurement and performing analytical integration from t_{k-1} to t_k [21]. The discrete dynamics for the individual state elements becomes

$$\bar{\mathbf{R}}_{\mathbf{t_k}} = \mathbf{R}_{\mathbf{t_{k-1}}} \exp(\tilde{\omega}_\mathbf{t}\Delta t), \quad (21)$$

$$\bar{\mathbf{v}}_{\mathbf{t_k}} = \mathbf{v}_{\mathbf{t_{k-1}}} + (\mathbf{R}_{\mathbf{t_{k-1}}}\tilde{\mathbf{a}}_\mathbf{t} + \mathbf{g})\Delta t, \quad (22)$$

$$\bar{\mathbf{p}}_{\mathbf{t_k}} = \mathbf{p}_{\mathbf{t_{k-1}}} + \mathbf{v}_{\mathbf{t_{k-1}}}\Delta t + \frac{1}{2}(\mathbf{R}_{\mathbf{t_{k-1}}}\tilde{\mathbf{a}}_\mathbf{t} + \mathbf{g})\Delta t^2, \quad (23)$$

where $\Delta t = t_k - t_{k-1}$. The analytical solution to the continuous-time Riccati equation (19) is given by

$$\mathbf{P_k} = \mathbf{\Phi}\mathbf{P_{k-1}}\mathbf{\Phi}^T + \bar{\mathbf{Q}}_\mathbf{k-1}, \quad (24)$$

where

$$\mathbf{\Phi} = \exp_m(\mathbf{A_t}\Delta t) \quad \bar{\mathbf{Q}}_\mathbf{k-1} \approx \mathbf{\Phi}\bar{\mathbf{Q}}_\mathbf{t}\mathbf{\Phi}^T\Delta t$$

ensuring covariance propagation. The explicit proof can be found in [21].

4) Left-Invariant Measurement Model: The Network measurement model corresponds to the left-invariant observation form: $\mathbf{Y_t} = \bar{\mathbf{X}}_\mathbf{t}\mathbf{b} + \mathbf{V_t}$

$$\begin{bmatrix} \mathbf{p} \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{R}}_\mathbf{t} & \bar{\mathbf{v}}_\mathbf{t} & \bar{\mathbf{p}}_\mathbf{t} \\ \mathbf{0_{1\times3}} & 1 & 0 \\ \mathbf{0_{1\times3}} & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{0_{3\times1}} \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} \mathbf{u} \\ 0 \\ 0 \end{bmatrix}, \quad (25)$$

where $\mathbf{p}$ is the position and $\mathbf{u}$ is the associated uncertainty, both of which are outputs predicted by the network. Therefore, the innovation depends solely on the invariant error and the update equations take the form

$$\eta_\mathbf{t}^{\mathbf{l}+} = \eta_\mathbf{t}^\mathbf{l} \exp\left(\mathbf{L_t}\left(\left(\eta_\mathbf{t}^\mathbf{l}\right)^{-1}\mathbf{b} - \mathbf{b} + \bar{\mathbf{X}}_\mathbf{t}^{-1}\mathbf{V_t}\right)\right), \quad (26)$$

where $\mathbf{L_t}$ is a gain matrix. Linearizing the left-invariant error $\eta_\mathbf{t}^\mathbf{l}$ and neglecting the higher order terms, we get

$$\xi_\mathbf{t}^{\mathbf{l}+} = \xi_\mathbf{t}^\mathbf{l} + \mathbf{L_t}\left(-(\xi_\mathbf{t}^\mathbf{l})^\wedge\mathbf{b} + \bar{\mathbf{X}}_\mathbf{t}^{-1}\mathbf{V_t}\right). \quad (27)$$

Define the measurement Jacobian $\mathbf{H}\xi = \xi^\wedge\mathbf{b}$, we can further write (27) as

$$\xi_\mathbf{t}^{\mathbf{l}+} = (\mathbf{I} - \mathbf{L_t}\mathbf{H})\,\xi_\mathbf{t}^\mathbf{l} + \mathbf{L_t}\bar{\mathbf{X}_\mathbf{t}}^{-1}\mathbf{V_t}. \quad (28)$$

Therefore, the full state and covariance update equation of the LIEKF can be written using the derived linear update equation and the theory of Kalman filtering as

$$\bar{\mathbf{X}}_\mathbf{t}^+ = \bar{\mathbf{X}}_\mathbf{t}\exp\left(\mathbf{L_t}\left(\bar{\mathbf{X}}_\mathbf{t}^{-1}\mathbf{Y_t} - \mathbf{b}\right)\right), \quad (29)$$

$$\mathbf{P}_\mathbf{t}^+ = (\mathbf{I} - \mathbf{L_t}\mathbf{H})\mathbf{P_t}(\mathbf{I} - \mathbf{L_t}\mathbf{H})^T + \mathbf{L_t}\bar{\mathbf{N}}_\mathbf{t}\mathbf{L_t}^T, \quad (30)$$

where

$$\bar{\mathbf{N}}_\mathbf{t} = \bar{\mathbf{X}}_\mathbf{t}^{-1}\mathrm{Cov}[\mathbf{V_t}]\bar{\mathbf{X}}_\mathbf{t}^{-T} \qquad \mathbf{L_t} = \mathbf{P_t}\mathbf{H}^T\mathbf{S}^{-1}$$

$$\mathbf{S} = \mathbf{H}\mathbf{P_t}\mathbf{H}^T + \bar{\mathbf{N}}_\mathbf{t} \qquad \mathbf{H} = \begin{bmatrix} \mathbf{0_{3\times3}} & \mathbf{0_{3\times3}} & \mathbf{I_{3\times3}} \end{bmatrix},$$

and $\mathrm{Cov}[\mathbf{V_t}]$ derived directly from (3).

C. Modules in Robust-VIO

In the *Lighting_assessment Module*, the current image is initially converted to grayscale. Subsequently, we calculate the average values of all pixels in adjacent M frames to obtain the current luminance level $\mathbf{I_t}$ of the environment, which is then input into the *Output Module*. Within the *Output Module*, a comparison is made between $\mathbf{I_t}$ and a predefined *threshold*. When $\mathbf{I_t}$ exceeds the *threshold*, we consider the lighting envionment to be favorable and output the position estimated by VIO. Conversely, if $\mathbf{I_t}$ falls below the *threshold*, we consider the lighting environment is adverse, then we utilize previous UAV state to initialize the LIDR and output the position estimated by LIDR from then on.

D. Implementation Details

To train the network in LIDR, an overlapping sliding window is adopted to extract IMU input samples. Each window contains N IMU samples of total size $N \times 6$. We set $N = 100$ in deloyment, which corresponds to a 0.5 s window of data for a 200 Hz IMU. To simulate scenarios in which the network may receive inaccurate UAV position inputs, we introduce random perturbations to the input positions using zero-mean Gaussian noise, thereby reducing the network's sensitivity to these input errors. Adam optimizer is employed for optimization with an initial learning rate of 0.0001. Furthermore, we applied zero weight decay and incorporated dropout with a probability of 0.5 into the two separate fully connected layers. The network is first trained with $\mathcal{L}_{MSE}$ to achieve numerical stability and subsequently transitions to $\mathcal{L}_{ML}$ to ensure the statistical stability. The noise values in LIEKF are: $\sigma_a = 0.5$, $\sigma_g = 0.1$. In practice, we scale the covariance from (3) by 10 to compensate for the temporal correlation of measurements. Note that this factor is a hyperparameter correlated with the accuracy of network inference, and smaller values will assign more confidence to the network outputs, leading to an increase in the Kalman gain during the update step. In the Robust-VIO, we consider $M = 4$ and set the *threshold* to 50 in real-world scenarios.

IV. Experiments

In the experiment section, we firstly validated the proposed LIDR performance by comparing it with state-of-the-art IMU dead-reckoning algorithms and VIO under favorable lighting conditions, both in the EuRoc dataset and real-world scenarios. Secondly, we conducted experiments where the UAV encounters a sudden lighting change from bright to dark. The proposed Robust-VIO is evaluated against pure VIO to demonstrate its enhancement of robustness in the environment with sudden lighting variation.

A. Performance Analysis in EuRoc Dataset

Experiment Setup: EuRoc dataset [14] is collected on-board a Micro Aerial Vehicle (MAV) that includes stereo images, synchronized IMU measurements, and accurate ground-truth data for motion and structure. The dataset comprises two scenarios: the Machine Hall (MH) and the Vicon Room (V1 and V2). We adopted a similar strategy as [2] to seperate EuRoc dataset. For

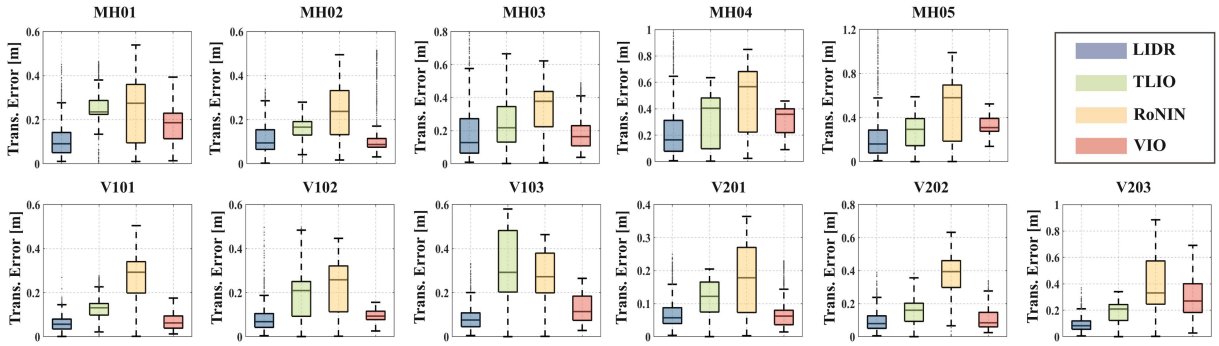

Fig. 3. EuRoc dataset evaluation. Translation errors achieved by RoNIN [9], VIO [3], TLIO [12], and LIDR(ours).

TABLE I
EuRoC Dataset Average Translation Error (m)

Trajectory	Algorithm			
	RoNIN [9]	TLIO [12]	VIO [3]	LIDR(ours)
MH_01_easy	0.25	0.25	0.17	**0.10**
MH_02_easy	0.23	0.16	0.12	**0.11**
MH_03_medium	0.33	0.25	**0.18**	**0.18**
MH_04_difficult	0.47	0.32	0.31	**0.24**
MH_05_difficult	0.49	0.27	0.32	**0.22**
V1_01_easy	0.26	0.12	**0.06**	**0.06**
V1_02_medium	0.23	0.19	0.09	**0.08**
V1_03_difficult	0.26	0.31	0.13	**0.08**
V2_01_easy	0.17	0.11	**0.07**	**0.07**
V2_02_medium	0.36	0.15	0.11	**0.09**
V2_03_difficult	0.40	0.18	0.30	**0.09**

The best value is in bold and the second-best is in underlined.

each trajectory sequence, 60% of the data is used for training, 15% of the data is used for validation, and 15% of the data is used for testing. In total, the training, validation, and test datasets contain approx. 14, 3.5 and 3.5 min of flight data. Note that during deployment, the position fed to the network is estimated from the previous timestamp, and the inertial measurement is sampled using an overlapping window approach. This confirms a different input feature to the network from the training phase. We choose the state-of-the-art IMU dead reckoning algorithms RoNIN [9], an estimator concatenating displacements from the trained network, and TLIO [12], a tightly-coupled EKF with displacement updates from the trained network, as baselines. To adapt RoNIN for UAV localization, we extend its original form to 3D as [12]. Both our network and the networks of TLIO and RoNIN are trained on the same dataset. We also incorporate VINS-Mono [3], a robust monocular visual-inertial state estimator, as a baseline. In the experiment, we deactivate its loop closing module while maintaining default parameters, denoted as 'VIO'.

Evaluation: Average translation error and error distribution for each sequence of EuRoc dataset are presented in Table I and Fig. 3 respectively. In Table I, the best value is in bold and the second-best is in underlined. LIDR outperform RoNIN and TLIO in all the sequences. The average improvements over RoNIN and TLIO are equal to 61% and 42%, respectively. LIDR also achieves comparable or slightly superior performance to VIO. Especially, as dataset complexity escalates from '*_easy*' to '*_medium*' to '*_difficult*', VIO exhibits a significant decline in accuracy, whereas LIDR demonstrate a more robust performance. The largest improvement over VIO is equal to 68% in *V2_03_difficult*. This is due to the motion pattern estimated by VIO is highly influenced by environmental factors, e.g., luminance and texture, while the motion pattern learned by LIDR remains unaffected. This characteristic enhances the robustness of LIDR in scenarios challenging for visual localization.

B. Performance Analysis in Real-World Scenarios

Experiment Setup: To validate the performance of our proposed algorithm in real-world scenarios, which is characterized by higher noise levels in IMU measurements compared to the EuRoc dataset, we conducted experiments using a custom-made hexacopter platform. Our hexacopter is equipped with an Intel RealSense T265 camera. T265 has a fisheye lens, capturing grayscale images at a resolution of 848 $\times$ 800 pixels, and is equipped with an integrated IMU. VIO from T265 has been employed in numerous onboard UAV systems, showcasing consistent and accurate localization performance [4], [17]. To prevent potential failures stemming from false calibration and parameter tuning, we directly adopt VIO from T265 as the baseline in the real-world scenarios, denoted as 'VIO'. In the Euroc dataset, TLIO has demonstrated clear superiority over RoNIN, primarily attributed to the tight coupling of the EKF with the network. Therefore, in this section we exclusively consider TLIO as the baseline for clarity. We also include IMO [2], a learned inertial odometry designed for drone racing that takes the collective thrust of the quadrotor into consideration in network design, as an additional baseline. The raw data is directly extracted from the T265 integrated IMU and used as input for considered methods. For the LIDR, TLIO, and IMO networks, we employed training, validation, and test datasets, each containing approx. 4.5, 1.5, and 2 minutes of flight data. Additionally, we calculated the thrust for IMO input using PWM signals derived in [28]. In the whole experiment sequence, the hexacopter is flown in a motion capture system and controlled by human practitioner.

Evaluation: The performance of baselines is summarized in Table II, and a visualization of the estimated trajectory is presented in Fig. 4. In Table II, the best value is in bold and the second-best is in underlined. The IMO accumulates significant drift in our experiment. We assume this is because the quadrotor is controlled in a similar way during each run by the human

TABLE II
PERFORMANCE EVALUATION IN REAL-WORLD SCENARIOS

Algorithm	Average Translation Error (m)
TLIO [12]	0.79
IMO [2]	1.39
LIDR(ours)	<u>0.17</u>
VIO	**0.14**

The best value is in bold and the second-best is in underlined.

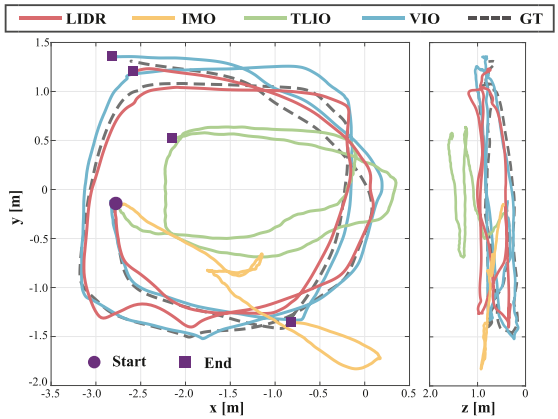

Fig. 4. Evaluation in real-world scenarios. Trajectoy estimated by VIO, TLIO [12], IMO [2] and LIDR(ours) after hexacopter takes off.

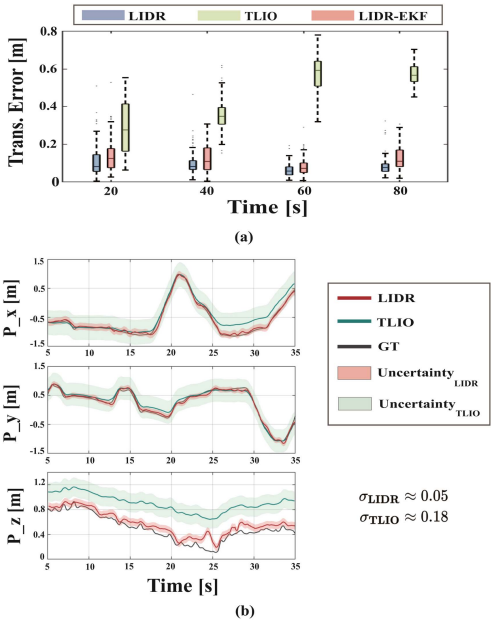

Fig. 5. Ablation study. (a) is the accumulative translation error over time. In LIDR-EKF, we adopt the same *Learning Module* in LIDR but a standard EKF for state propagation. (b) is the estimated x, y, z position as well as associated uncertainties ($\pm 3\sigma$) of TLIO [12], and our algorithm. The standard deviation (σ) of considered methods in the sequence is annotated on the right side.

pilot in drone racing, which makes thrust a prominent feature for network learning. Our algorithm achieves the second-best performance in real-world scenarios, demonstrating comparable results with VIO despite the noisy inertial measurements from T265. Performance of TLIO exhibits a notable decline in real-world scenarios than in EuRoc dataset. The accuracy achieved by LIDR outperformed TLIO by 78%, confirming that our algorithm is more robust with noisy input.

Ablation Study: Since our work share the same concept with TLIO, we additionally conducted an ablation study to further evaluate the proposed algorithm. We use the network trained on real-world scenarios in the learning module and evaluate the performance on different trajectories. To directly evaluate the effectiveness of our modifications in *learning module* and *filter module*, we combine the *learning module* in LIDR with a standard EKF as an additional baseline, denoted as 'LIDR-EKF'. Note that the stochastic cloning EKF adopted in TLIO is a derivative of the EKF, primarily designed for processing relative state measurements, and it shares the same convergence properties as the EKF. Accumulative translation error over time of the considered methods is shown in Fig. 5(a). The accumulative error of TLIO increased significantly with time, whereas LIDR-EKF, despite adopting EKF similar with TLIO, exhibits better performance. This suggests that the *learning module* in our algorithm produces more accurate result for the filter updates. LIDR outperforms LIDR-EKF by 25% in accuracy and exhibits a more concentrated error distribution, demonstrating a more consistent performance over time. This result suggests that the adopted LIEKF is a more stable state observer than EKF on the noisy inertial measurement and further proves that dropping off the bias term in state propagation does not lead to the degradation of the overall algorithm. We further evaluate the stability properties of the proposed algorithm. To this end, we adopt the UAV state at slightly different timestamps for initialization, and then calculate the estimated trajectory with the associated uncertainty in the same sequence, as shown in Fig. 5(b). The uncertainty of LIDR is only 27% of TLIO, and the trajectory estimated by LIDR is more consistent with the ground truth. This result further validates the robustness of our algorithm.

C. Validation of Robust-VIO

In this section, our proposed Robust-VIO is validated to achieve robust localization in the environment with sudden lighting variation. For a fair comparison, we adopt the VIO from T265 in Robust-VIO, the same as our experimental setup in the previous section. In the experiment, we controlled the hexacopter in an indoor environment and abruptly turned off all lights to simulate the lighting variation, the video of our experiment is shown in the supplementary material. A visualization of the trajectory estimated by VIO and our approach as well as the cumulative error over time is shown in Fig. 6. VIO exhibits significant drift in trajectory and accumulates a large error after the lighting variation. In contrast, our framework actively detects the current luminance level of the environment and activates LIDR for localization when the lighting conditions getting adverse. Runtime of different modules in our framework are listed in Table III. Our algorithm achieves an approximate runtime of 23 Hz on Nvidia Xavier NX, a commonly deployed onboard laptop on UAVs, which satisfying real-time requirements. It's worth noting that our algorithm is now built on Python, and migrating it to C++ could further enhance its runtime performance.

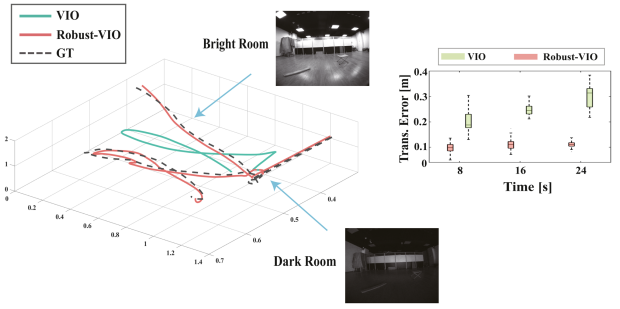

Fig. 6. (Left)Trajectory estimated by VIO and our approach. The image is captured by the onboard T265 camera, and the current luminance level $\mathbf{I_t}$ is displayed above the image. (Right)Accumulative translation error over time after the hexacopter enters a dark room. VIO demonstrates accurate localization accuracy in a bright room, but after the lighting variation, VIO exhibits significant drift. In the Robust-VIO, the *Lighting_assessment* module activates LIDR for localization when the current luminance level $\mathbf{I_t}$ is below the *threshold*, achieving robust localization when VIO fails.

TABLE III
RUNTIME ANALYSIS ON ONBOARD COMPUTER NVIDIA XAVIER NX

Stage	CPU/GPU	Time (ms)
Filter Module (propogation)	CPU	3.6
Filter Module (update)	CPU	22.2
Learning Module	GPU	14.4
Robust-VIO Framework	CPU	2.3
Total		42.5

V. CONCLUSION

In this letter, we propose a learning-based IMU dead-reckoning algorithm for UAV localization. The proposed algorithm incorporates a statistical neural network to predict the position of UAV along with the corresponding uncertainty, which is then used to update the LIEKF. The LIEKF is propagated using the inertial measurement. The performance of our algorithm is validated in both the EuRoc dataset and real-world scenarios with state-of-the-art IMU dead-reckoning algorithms and VIO methods. To achieve robust localization in different lighting conditions, we further introduce an approach to integrate our algorithm with VIO system. This approach have been validated in real-world scenarios to achieve consistent localization when UAV encounters sudden lighting variation in the environment, where VIO fails. While our algorithm are specifically constructed and validated for UAVs, they can be generalized to other mobile robots to address the impact of lighting variation on VIO-based localization.

Similar to other learning-based methods, our proposed algorithm has limitations associated with the scope of training data. When encountering an unseen trajectory, the network may output inaccurate estimates, leading the filter module to converge to an incorrect position. A promising solution is to design a more sophisticated network structure that can extract features in both spatial and temporal sequences, enhancing its generalization over unseen trajectories.

REFERENCES

[1] B. Ma, Z. Jiang, Y. Liu, and Z. Xie, "Advances in space robots for on-orbit servicing: A comprehensive review," *Adv. Intell. Syst.*, vol. 8, no. 3, 2023, Art. no. 2200397.

[2] G. Cioffi, L. Bauersfeld, E. Kaufmann, and D. Scaramuzza, "Learned inertial odometry for autonomous drone racing," *IEEE Robot. Automat. Lett.*, vol. 8, no. 5, pp. 2684–2691, May 2023.

[3] 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.

[4] A. Agarwal, J. R. Crouse, and E. N. Johnson, "Evaluation of a commercially available autonomous visual inertial odometry solution for indoor navigation," in *Proc. Int. Conf. Unmanned Aircr. Syst.*, 2020, pp. 372–381.

[5] Y. Yang, P. Geneva, X. Zuo, and G. Huang, "Online IMU intrinsic calibration: Is it necessary?," in *Proc. Robot.: Sci. Syst.*, 2020.

[6] H. Liu et al., "Collaborative robots sim: A simulation environment of air-ground robots with strong physical interactivity," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, 2021, pp. 1841–1847.

[7] F. Guo, H. Yang, X. Wu, H. Dong, Q. Wu, and Z. Li, "Model-based deep learning for low-cost IMU dead reckoning of wheeled mobile robot," *IEEE Trans. Ind. Electron.*, Aug. 15, 2023, early access, doi: 10.1109/TIE.2023.3301531.

[8] M. Brossard, S. Bonnabel, and A. Barrau, "Denoising IMU gyroscopes with deep learning for open-loop attitude estimation," *IEEE Robot. Automat. Lett.*, vol. 5, no. 3, pp. 4796–4803, Jul. 2020.

[9] S. Herath, H. Yan, and Y. Furukawa, "Ronin: Robust neural inertial navigation in the wild: Benchmark, evaluations, & new methods," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 3146–3152.

[10] C. Chen, X. Lu, A. Markham, and N. Trigoni, "IONet: Learning to cure the curse of drift in inertial odometry," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 6468–6476.

[11] H. Yan, Q. Shan, and Y. Furukawa, "RIDI: Robust IMU double integration," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 621–636.

[12] W. Liu et al., "TLIO: Tight learned inertial odometry," *IEEE Robot. Automat. Lett.*, vol. 5, no. 4, pp. 5653–5660, Oct. 2020.

[13] K. Zhang et al., "DIDO: Deep inertial quadrotor dynamical odometry," *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 9083–9090, Oct. 2022.

[14] M. Burri et al., "The euroc micro aerial vehicle datasets," *Int. J. Robot. Res.*, vol. 35, no. 10, pp. 1157–1163, 2016.

[15] W. Liu, K. Mohta, G. Loianno, K. Daniilidis, and V. Kumar, "Semi-dense visual-inertial odometry and mapping for computationally constrained platforms," *Auton. Robots*, vol. 45, no. 6, pp. 773–787, 2021.

[16] X. Zhou et al., "Swarm of micro flying robots in the wild," *Sci. Robot.*, vol. 7, no. 66, 2022, Art. no. eabm5954.

[17] R. Bonatti, R. Madaan, V. Vineet, S. Scherer, and A. Kapoor, "Learning visuomotor policies for aerial navigation using cross-modal representations," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 1637–1644.

[18] J. Svacha, K. Mohta, M. Watterson, G. Loianno, and V. Kumar, "Inertial velocity and attitude estimation for quadrotors," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 1–9.

[19] J. Svacha, G. Loianno, and V. Kumar, "Inertial yaw-independent velocity and attitude estimation for high-speed quadrotor flight," *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 1109–1116, Apr. 2019.

[20] A. Barrau and S. Bonnabel, "The invariant extended Kalman filter as a stable observer," *IEEE Trans. Autom. Control*, vol. 62, no. 4, pp. 1797–1812, Apr. 2017.

[21] R. Hartley, M. Ghaffari, R. M. Eustice, and J. W. Grizzle, "Contact-aided invariant extended kalman filtering for robot state estimation," *Int. J. Robot. Res.*, vol. 39, no. 4, pp. 402–430, 2020.

[22] E. R. Potokar, K. Norman, and J. G. Mangelson, "Invariant extended kalman filtering for underwater navigation," *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 5792–5799, Jul. 2021.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[24] R. L. Russell and C. Reale, "Multivariate uncertainty in deep learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 12, pp. 7937–7943, 2021.

[25] P. Zhang, X. Zhan, X. Zhang, and L. Zheng, "Error characteristics analysis and calibration testing for MEMS IMU gyroscope," *Aerosp. Syst.*, vol. 2, pp. 97–104, 2019.

[26] A. Barrau, "Non-linear state error-based extended Kalman filters with applications to navigation," Ph.D. dissertation, Mines Paristech, Paris, France, 2015.

[27] P. S. Maybeck, *Stochastic Models, Estimation, and Control*. Cambridge, MA, USA: Academic Press, 1982.

[28] Y. Yu and X. Ding, "A quadrotor test bench for six degree of freedom flight," *J. Intell. Robot. Syst.*, vol. 68, pp. 323–338, 2012.