IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 25 June 2024, accepted 4 August 2024, date of publication 7 August 2024, date of current version 19 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3440182

RESEARCH ARTICLE

Learning Visual-Inertial Odometry With Robocentric Iterated Extended Kalman Filter

KHAC DUY NGUYEN[1], DINH TUAN TRAN[2], (Member, IEEE), VAN QUYEN PHAM[3], DINH TUAN NGUYEN[3], KATSUMI INOUE[4], (Member, IEEE), JOO-HO LEE[2], (Senior Member, IEEE), AND ANH QUANG NGUYEN[3]

[1]School of Information and Communication Technology, Hanoi University of Science and Technology, Hanoi 100000, Vietnam
[2]College of Information Science and Engineering, Ritsumeikan University, Kyoto 603-8577, Japan
[3]School of Electrical and Electronic Engineering, Hanoi University of Science and Technology, Hanoi 100000, Vietnam
[4]National Institute of Informatics, Chiyoda 101-0003, Japan

Corresponding author: Anh Quang Nguyen (quang.nguyenanh@hust.edu.vn)

This research is supported by the Research Organization of Information and Systems (ROIS) National Institute of Informatics (NII) Open Collaborative Research 2023 under project number 23FP01 (Japan).

ABSTRACT In recent years, deep learning methodologies have been increasingly applied to the intricate challenges of visual-inertial odometry (VIO), especially in scenarios with rapid movements and scenes lacking clear structure. This paper introduces a novel hybrid approach that leverages the inherent strengths of traditional VIO techniques, while harnessing the potential of advanced machine learning technologies. By seamlessly integrating an iterated extended Kalman filter with deep learning techniques, our approach systematically takes into account uncertainties, thereby enhancing the overall reliability and robustness of the system. The proposed algorithm has been rigorously evaluated on the KITTI and EuroC datasets, outperforming other deep learning VIO methods. It achieved a translation error of 2.28% and a rotation error of 0.226 degrees per 100 meters on the KITTI odometry dataset.

INDEX TERMS Monocular camera, visual inertial odometry, iterated extended Kalman filter, deep learning.

I. INTRODUCTION

Estimating a robot's position, velocity, and orientation, referred to as odometry, poses significant challenges, especially in complex and dynamic environments. Visual-inertial odometry (VIO) is a commonly used approach that combines data from cameras and inertial measurement units (IMUs) to calculate the actual changes in position between consecutive camera frames [1], [2], [3], [4]. Nevertheless, VIO encounters difficulties like motion blur, changing lighting conditions, and drifting. Recent advancements in computer vision and machine learning have led to the emergence of innovative VIO techniques that integrate visual and inertial measurements for more precise and dependable positioning. However, traditional VIO methods still have shortcomings, such as the need for meticulous calibration and susceptibility to noise and errors.

To address these limitations, researchers have proposed employing deep learning methods to enhance the accuracy and efficiency of ego-motion estimation [5], [6], [7]. These techniques utilize Convolutional Neural Networks (CNNs) to extract image features and fully connected neural networks to process inertial data. The amalgamation of this information is then employed to compute relative ego-motion using a recurrent or attention-based regressor. These approaches have displayed promising outcomes in improving positioning accuracy and resilience in various environments, including those with challenging lighting conditions. Nonetheless, the estimated ego-motion remains vulnerable to abrupt changes in scene settings, such as velocity alterations. Furthermore, since IMU data is low-dimensional and adheres to a well-understood physics model, employing a deep network for processing IMU data may not always be necessary.

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

Previous research on hybrid deep learning-based VIO approaches has explored integrating a deep learning model with an Extended Kalman Filter (EKF) [8], [9]. These approaches employ the deep model to process image data, which is then fused with IMU data using the EKF. This approach has been shown to be effective with sufficient data quality. However, these methods typically focus on the EKF model and use relatively simple deep learning models. We argue that both components of a hybrid VIO system must be strong independently in order for the overall model to achieve high accuracy and minimize uncertainty.

Motivated by the limitations of existing hybrid deep learning-based VIO approaches, we introduce the Deep Iterated EKF VIO model (DI-EKF-VIO), a hybrid system that combines the strengths of classical and learning-based VIO systems. Our proposed method employs an Iterated Extended Kalman Filter (IEKF) to integrate ego-motion measurements derived from an optical flow network into a state estimator that is keenly aware of uncertainty. The state is propagated between camera frames by incorporating IMU measurements. In an extension of prior research, our predictions are imbued with a deep-learned heteroscedastic uncertainty model, enabling a systematic fusion of measurements within the filter. This innovative approach combines the robustness of classical VIO methods with the accuracy enhancements of learning-based systems, resulting in a more precise and dependable positioning system suitable for diverse environments. The architectural representation of our proposed method is illustrated in Figure 1.

The subsequent sections of this paper are organized as follows: Section II provides a concise overview of previous VIO methods, encompassing both conventional and learning-based approaches. Section III elaborates on our methodology, which integrates the Iterated Extended Kalman Filter and Deep Learning into VIO. Section IV assesses the performance of our approach using the KITTI and EuroC datasets. Finally, in Section V, we draw conclusions from our work.

II. RELATED WORK

A. CLASSICAL APPROACH

Traditional visual odometry methods used in Simultaneous Localization and Mapping (SLAM) algorithms follow a multi-step process. First, distinctive features are extracted from two consecutive images. These features are then described and matched in subsequent frames. The algorithm then associates the matched features with their corresponding positions in the map and updates the map accordingly. The feature extraction module is crucial in this process, as it is responsible for identifying unique features in the images.

These algorithms can be divided into two categories based on the number of feature points they use: sparse and dense. Sparse methods use a small number of feature points from an image, while dense methods use most or all of the feature points in an image. Dense methods require more computational resources than sparse methods, but they are more robust and accurate in situations with limited structural information (fewer feature points). In sparse methods, it is important to select the optimal subset of feature points in an image. These points should have distinctive characteristics and be easy to match in consecutive frames.

Loosely-coupled visual-inertial odometry methods, such as Multi-Sensor Fusion (MSF) [10], accept pose estimates from inertial sensors (such as IMUs) and combine them with pose estimates from visual odometry using an Extended Kalman Filter. MSF estimates pose, velocities, and biases, as well as a scaling factor to account for the scaling drift that can occur in monocular VO. Tightly-coupled VIO methods, such as Multi-State Constraint Kalman Filter (MSCKF) [1], OKVIS [4], and VINS-Mono [11], extract, track, and triangulate features from images and fuse them with IMU-propagated poses. MSCKF uses a least-squares optimization technique to triangulate features, while OKVIS and VINS-Mono use iterative non-linear least-squares optimizations to achieve fusion. Optimization methods are more computationally demanding than filter-based methods, but they tend to be more accurate.

While traditional VIO techniques offer a valuable tool for odometry, they encounter limitations in specific scenarios. Challenging lighting conditions, rapid camera motion, and occlusions can significantly hinder feature extraction and tracking, leading to inaccurate pose estimation. Modern SLAM methods often leverage Global optimization techniques, like bundle adjustment or least-squares minimization, which can be computationally expensive. This can be a significant disadvantage for real-time applications or those with limited resources. Processing large amounts of data and complex calculations can slow down the SLAM process. Moreover, As the size and complexity of the map grows, global optimization becomes even more computationally demanding. This can limit the scalability of SLAM systems for very large or dynamic environments. Additionally, depending on the initial conditions and the chosen optimization algorithm, there's a possibility of not reaching the optimal solution or getting stuck in local minima. This can lead to a sub-optimal map that doesn't accurately represent the environment.

There's a growing interest in deep learning-based SLAM. Deep learning eliminates the need for hand-crafted features. Convolutional Neural Networks can automatically learn robust features directly from image data, adapting to various lighting conditions, scales, and viewpoints. Additionally, deep learning models can be more robust to challenging environments with limited unique features or heavy occlusions. They can also be trained to handle dynamic scenes with motion blur or fast movements.

B. LEARNING-BASED APPROACH

Deep learning has made significant progress in solving classification problems in computer vision and are becoming a major area of research. It has played a vital role in addressing these challenges, and is now the primary focus

of research in this field. Deep learning techniques have been integrated into various aspects of conventional odometry methods, such as feature extraction, feature matching and pose estimation. The use of Convolutional Neural Networks has been instrumental in improving the performance of these deep learning approaches in addressing these challenges. CNNs are well-suited for learning to extract diverse features and fusing them to describe either the entire input image or specific regions. This ability to learn meaningful representations for perceptual understanding by combining abstract features makes the resulting descriptions resilient to noise and well-equipped to handle specific challenges. These sophisticated feature representations, robustly acquired through deep CNNs, have been used to address a wide range of computer vision problems.

Aside from earlier explorations of learning-based visual odometry models [12], [13], there has been a growing emphasis on enhancing the accuracy and robustness of end-to-end learning-based VO models. To enhance their capabilities, end-to-end VO models often incorporate auxiliary outputs related to camera movements, including depth and optical flow. These models predict depth by maintaining depth consistency between consecutive images, which provides supervisory signals for training the model [7], [14]. A similar temporal matching effect can be achieved by simultaneously predicting optical flow [15], [16], which encompasses joint predictions of depth, optical flow, and camera motion.

While deep learning offers significant potential, it also comes with challenges of Computational Demands: Training and running deep learning models can be computationally expensive, potentially limiting their application on resource-constrained platforms. Another concern is that Deep learning models can be complex ''black boxes,'' making it difficult to understand how they arrive at their results. This can be a concern for safety-critical applications where understanding errors or biases is crucial.

The future of SLAM likely lies in hybrid approaches that combine the strengths of traditional methods with deep learning. Previous studies on hybrid deep learning-based VIO approaches have integrated a deep learning model with an Extended Kalman Filter [8], [9]. These approaches use the deep model to process image data, which is then fused with IMU data using the EKF. This method has been shown to be effective with sufficient high-quality data. However, it primarily focuses on the EKF model and uses relatively simple deep learning models. We argue that both components of a hybrid VIO system must excel independently in order for the overall model to achieve robust performance and accuracy. Our goal is to improve both the EKF model and the deep network within the existing framework while maintaining the overall structure.

C. OTHER APPROACH

Alternative methodologies for Odometry incorporate LiDAR input rather than IMU, yielding notable outcomes as demonstrated by [17], [18], [19]. LiDAR input offers the benefit of precise depth information in contrast to both visual and IMU data, thereby harboring the potential to surpass visual-inertial techniques. Nevertheless, we have chosen not to explore LiDAR-based approaches for a multitude of reasons. First, LiDAR sensors are typically more expensive and bulkier than visual and inertial sensors, which can limit their accessibility, especially in scenarios with budgetary and size constraints. We opted to focus on visual and inertial sensors to ensure our system remains cost-effective and accessible, particularly for researchers and practitioners operating within resource-limited environments. Moreover, although LiDAR sensors provide precise depth information, modern deep learning algorithms can approximate similar results. In our proposed method, depth information is estimated and extracted from temporal visual data, potentially making LiDAR information redundant if the deep learning algorithm performs adequately. And lastly, by concentrating exclusively on visual and inertial sensor fusion, we were able to dedicate our efforts to refining deep learning algorithms tailored specifically to these sensor modalities. This focused approach allowed us to explore the potential of deep learning techniques within the context of visual-inertial odometry without introducing the added complexity of integrating LiDAR data.

Aside from sensor fusion method that bring more than 1 input to the problem, there are also approach that utilize only 1 input, notably Visual data only [6], [20], [21], Inertial only [22], [23], LiDAR data only [24], [25], [26]. While single-sensor methods have been explored in prior research, we have chosen not to pursue them for several compelling reasons. In general, single sensor methods are often susceptible to environmental conditions and sensor-specific limitations. For example, camera-only methods might struggle in low-light conditions or with featureless environments, while IMU-only methods can suffer from drift over time. Every sensor has its own advantage and limitation. Camera-only methods might not capture depth information accurately at long distances, while LiDAR-only methods might struggle with occlusions or reflective surfaces. Most of the single-sensor method tries to exploit the advantage of the utilized sensor while trying to cover its incompletion. Sensor fusion methods, on the other hand, leverage the complementary strengths of multiple sensors to improve overall performance. By integrating data from multiple sources, these methods can provide more robust and accurate odometry estimates compared to single sensor methods.

III. DI-EKF-VIO: DEEP ITERATED EXTENDED KALMAN FILTER

In our hybrid approach to Visual-Inertial Odometry, we enhance the ego-motion estimation system by incorporating a robocentric Iterated Extended Kalman Filter backend. This section provides an overview of our architecture, as described in Section III-A, then an introduction to the notation used in this work in Section III-B. Subsequently, we dig further into the formulation of the robocentric IEKF in

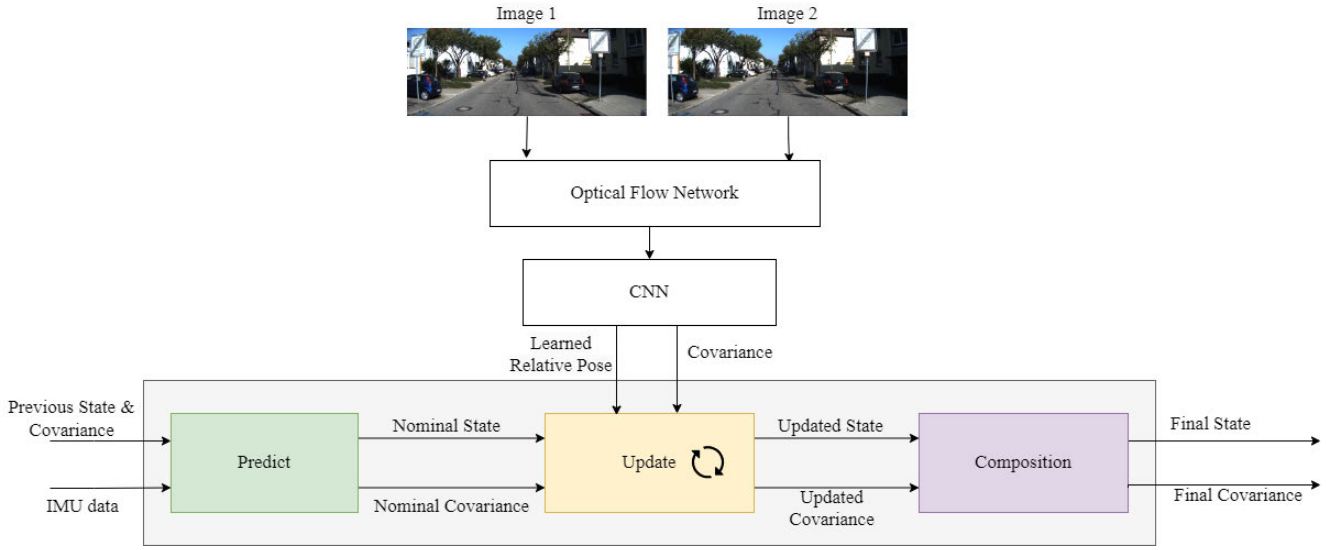

FIGURE 1. Overall proposed method architecture.

Section III-C. The fusion of our approach with deep learning is discussed in Section III-D.

A. OVERALL ARCHITECTURE

Depicted in figure 1, the proposed system takes advantage of both image data from a camera and IMU data to estimate the robot's pose over time. The input to the system consists of these two data sources. The image data is first processed by a deep learning model, which is tasked with extracting informative features and estimating the relative pose changes between consecutive images. This relative pose information describes the robot's movement in terms of 6 DoF.

The deep learning model's predictions for relative pose are then fed into an iterative Extended Kalman Filter alongside the IMU data. The IEKF acts as a fusion module, combining the strengths of both data sources. By incorporating both sources, the IEKF refines the pose estimations and generates a more robust and accurate representation of the robot's absolute pose in the environment. This absolute pose information is represented as a 4×4 transformation matrix, which can be directly used for various robotic tasks such as navigation and manipulation.

In essence, the system leverages the deep learning model's ability to extract meaningful features from images for precise relative pose estimation, while the EKF incorporates the continuous and complementary information from the IMU to achieve a robust and accurate estimation of the robot's absolute pose over time.

B. NOTATION

Our work employs three distinct frames: The inertial frame, denoted as $\underrightarrow{\mathcal{F}}_i$. The reference frame at time k is denoted as $\underrightarrow{\mathcal{F}}_{r_k}$, where k and $k+1$ correspond to points in time when an image is received. The vehicle frame at time τ is denoted as $\underrightarrow{\mathcal{F}}_{v_\tau}$, where τ and $\tau+1$ correspond to points in time when an IMU measurement is received between time k and $k+1$.

We use the symbol δ to represent a perturbation to subsequent quantities. Subscripts and superscripts are employed to track the coordinate frames of physical quantities. For instance, $\mathbf{v}_a^{bc}$ represents the velocity of $\underrightarrow{\mathcal{F}}_b$ relative to $\underrightarrow{\mathcal{F}}_c$ expressed in $\underrightarrow{\mathcal{F}}_a$.

A rigid transformation from frame $\underrightarrow{\mathcal{F}}_b$ to frame $\underrightarrow{\mathcal{F}}_a$ is denoted as $\mathbf{T}_{ab} \in \mathbb{SE}(3)$, which consists of two components: the translation $\mathbf{r}_{ab} \in \mathbb{R}^3$ and the rotation $\mathbf{C}_{ab} \in \mathbb{SO}(3)$.

The gravity expressed in $\underrightarrow{\mathcal{F}}_a$ is denoted as $\mathbf{g}_a \in \mathbb{R}^3$. The velocity of $\underrightarrow{\mathcal{F}}_a$ with respect to $\underrightarrow{\mathcal{F}}_b$ expressed in $\underrightarrow{\mathcal{F}}_c$ is represented as $\mathbf{v}_c^{ab} \in \mathbb{R}^3$.

The gyroscope and accelerometer measurements at time τ are denoted as $\mathbf{b}_{\omega_\tau}$ and $\mathbf{b}_{a_\tau}$, both belonging to $\mathbb{R}^3$.

We use $\bar{(\cdot)}$ to indicate noisy quantities, $\check{(\cdot)}$ for quantities propagated throughout the Extended Kalman Filter prediction step, and $\hat{(\cdot)}$ for quantities corrected after the EKF update step.

C. ROBOCENTRIC IEKF

We based our IEKF formulation on the approach of [8] and [9]. The states at the latest IMU measurement time step, τ, is comprise of 2 elements: the inertial state $\mathbf{x}_{r_k i}$ and the vehicle state $\mathbf{x}_{r_k v_\tau}$ which is formulated by the following component:

$$\begin{aligned}\mathbf{x}_\tau &= \left[\, \mathbf{x}_{r_k i} \mid \mathbf{x}_{r_k v_\tau} \,\right] \\ &= \Big[\, \mathbf{C}_{r_k i}\ \mathbf{r}_{r_k}^{i r_k}\ \mathbf{g}_{r_k} \mid \\ &\qquad \mathbf{C}_{r_k v_\tau}\ \mathbf{r}_{r_k}^{v_\tau r_k}\ \mathbf{v}_{v_\tau}^{v_\tau i}\ \mathbf{b}_{\omega_\tau}\ \mathbf{b}_{a_\tau} \,\Big]\end{aligned} \tag{1}$$

The error states is then defined as:

$$\begin{aligned}\delta\mathbf{x}_\tau = \Big[\, &\delta\boldsymbol{\phi}_{r_k i}^\top\ \delta\mathbf{r}_{r_k}^{i r_k \top}\ \delta\mathbf{g}_{r_k}^\top \mid \\ &\delta\phi_{r_k v_\tau}^\top\ \delta\mathbf{r}_{r_k}^{v_\tau r_k \top}\ \delta\mathbf{v}_{v_\tau}^{v_\tau i \top}\ \delta\mathbf{b}_{\omega_\tau}^\top\ \delta\mathbf{b}_{a_\tau}^\top \,\Big]^\top\end{aligned} \tag{2}$$

The error states are defined within the vector space $\mathbb{R}^3$, representing perturbations to the states defined in equation 1. These perturbations are handled using simple addition, except for the rotational quantities, which are defined within the space $\mathfrak{so}(3)$. Specifically, the formulation for error in rotational quantities is as presented in equation 3, where $\mathbf{C} \in \mathbb{SO}(3)$ and $\phi \in \mathbb{R}^3$, with $(\cdot)^\wedge$ denoting the skew-symmetric operator.

$$\mathbf{C} = \overline{\mathbf{C}} \exp\left(\phi^\wedge\right) \tag{3}$$

Next, we calculate the time derivatives of the states and their constituent elements.

$$\begin{aligned}
\dot{\mathbf{C}}_{r_k v_\tau} &= \mathbf{C}_{r_k v_\tau} \left[\boldsymbol{\omega}_{v_\tau}^{v_\tau i}\right]^\wedge \\
\dot{\mathbf{r}}_{r_k}^{v_\tau r_k} &= \mathbf{C}_{r_k v_\tau} \mathbf{v}_{v_\tau}^{v_\tau i} \\
\dot{\mathbf{v}}_{v_\tau}^{v_\tau i} &= \mathbf{a}_{v_\tau}^{v_\tau i} - \left[\boldsymbol{\omega}_{v_\tau}^{v_\tau i}\right]^\wedge \mathbf{v}_{v_\tau}^{v_\tau i} \\
\dot{\mathbf{b}}_{\omega_\tau} &= \mathbf{n}_{b_\omega} \\
\dot{\mathbf{b}}_{a_\tau} &= \mathbf{n}_{b_a}
\end{aligned} \tag{4}$$

Subsequently, we formulate the measurement model for the IMU states and apply perturbations to these states. Through conjunction with equation 4, this procedural step culminates in the derivation of equation 5 in a matrix representation. This equation encompasses the linearized system matrix denoted as $\mathbf{F}$, the linearized error matrix symbolized by $\mathbf{G}$, and the noise term $\mathbf{n}$.

$$\delta\dot{\mathbf{x}}_\tau = \mathbf{F}\delta\mathbf{x}_\tau + \mathbf{G}\mathbf{n} \tag{5}$$

where $\mathbf{n} = \begin{bmatrix} \mathbf{n}_\omega^\top & \mathbf{n}_{b_\omega}^\top & \mathbf{n}_a^\top & \mathbf{n}_{b_a}^\top \end{bmatrix}^\top$

The process model for the IMU states is employed during the prediction phase to advance the state estimate, denoted as $\hat{\mathbf{x}}k$, relative to the most recent robocentric frame $\overrightarrow{\mathcal{F}} r_k$. This advancement occurs from time step k to time step τ utilizing the IMU measurements. The outcome of this operation is the predicted IMU state, represented as $\check{\mathbf{x}}_{rkv_\tau}$. Subsequently, this predicted state undergoes further processing using Euler's method, resulting in the outcomes denoted in equation 6:

$$\begin{aligned}
\check{\mathbf{C}}_{r_k v_{\tau+1}} &\approx \check{\mathbf{C}}_{r_k v_\tau} \exp\left(\left(\boldsymbol{\omega}_m - \check{\mathbf{b}}_{\omega_\tau}\right)^\wedge \delta t\right) \\
\check{\mathbf{r}}_{r_k}^{v_{\tau+1} r_k} &\approx \check{\mathbf{r}}_{r_k}^{v_\tau r_k} + \check{\mathbf{v}}_{r_k}^{v_\tau i}\delta t + \frac{1}{2}\check{\mathbf{C}}_{r_k v_\tau}\left(\mathbf{a}_m - \check{\mathbf{b}}_{a_\tau}\right)\delta t^2 \\
\check{\mathbf{v}}_{r_k}^{v_{\tau+1} i} &\approx \check{\mathbf{v}}_{r_k}^{v_\tau i} + \check{\mathbf{C}}_{r_k v_\tau}\left(\mathbf{a}_m - \check{\mathbf{b}}_{a_\tau}\right)\delta t
\end{aligned} \tag{6}$$

To advance the state covariance forward in time, we need the transition matrix for the error state between IMU time steps. This matrix can be efficiently approximated using a first-order approximation method, as indicated by equation 7, where $\mathbf{1}$ represents the identity matrix and $\delta t = t_{\tau+1} - t_\tau$.

$$\Phi_{\tau+1,\tau} = \exp\left(\int_{t_\tau}^{t_{\tau+1}} \mathbf{F}(s)ds\right) \approx \mathbf{1} + \mathbf{F}\delta t \tag{7}$$

The predicted state uncertainty can then be expressed as:

$$\begin{aligned}
\mathbf{Q} &= \operatorname{diag}\left(\sigma_\omega^2\mathbf{1}, \sigma_a^2\mathbf{1}, \sigma_{b_\omega}^2\mathbf{1}, \sigma_{b_a}^2\mathbf{1}\right), \\
\check{\mathbf{P}}_{\tau+1} &= \boldsymbol{\Phi}_{\tau+1,\tau}\check{\mathbf{P}}_\tau\boldsymbol{\Phi}_{\tau+1,\tau}^\top + \mathbf{G}\mathbf{Q}\mathbf{G}^\top\delta t
\end{aligned} \tag{8}$$

The measurement model in our IEKF, which is the deep learning network, outputs relative pose measurements $\tilde{\mathbf{z}}_k$, which is composed of $\begin{bmatrix} \tilde{\boldsymbol{\phi}}_{r_k v_{k+1}}^\top & \tilde{\mathbf{r}}_{r_k}^{v_{k+1} r_k \top} \end{bmatrix}^\top$ with corresponding covariances $\mathbf{R}_k$ (which will be discussed further in Section III-D). We can begin to use the relative pose measurement $\tilde{\mathbf{z}}_k$ for linearizing measurement model, and use covariance $\mathbf{R}_k$ in the updating step.

The measurement residual $\boldsymbol{\epsilon}_{k+1} = \begin{bmatrix} \boldsymbol{\epsilon}_\theta^\top & \boldsymbol{\epsilon}_r^\top \end{bmatrix}^\top$ can be approximated as the subtraction of two rotational vector using the first-order Baker-Campbell-Hausdorff formula as shown in equation 9

$$\begin{aligned}
\boldsymbol{\epsilon}_\theta &= \ln\left(\exp\left(\tilde{\boldsymbol{\phi}}_{r_k v_{k+1}}^\wedge\right)\exp\left(\boldsymbol{\phi}_{r_k v_{k+1}}^\wedge\right)^\top\right)^\vee \\
&\approx \tilde{\boldsymbol{\phi}}_{r_k v_{k+1}} - \boldsymbol{\phi}_{r_k v_{k+1}}
\end{aligned} \tag{9}$$

Subsequently, we differentiate relative pose measurement equation with respect to the error states to derive the measurement Jacobian $\mathbf{H}_{k+1} = \frac{\partial \boldsymbol{\epsilon}_{k+1}}{\partial \delta \mathbf{x}_{k+1}}$. The derivations are presented as follows:

$$\begin{aligned}
\boldsymbol{\epsilon}_\theta &\approx \underbrace{\tilde{\boldsymbol{\phi}}_{r_k v_{k+1}} - \hat{\boldsymbol{\phi}}_{r_k v_{k+1}}}_{\bar{\epsilon}_\theta} - \mathbf{J}_r\left(\phi_{r_k v_{k+1}}\right)^{-1}\delta\phi_{r_k v_{k+1}} \\
\boldsymbol{\epsilon}_r &= \underbrace{\tilde{\mathbf{r}}_{r_k}^{v_{k+1} r_k} - \hat{\mathbf{r}}_{r_k}^{v_{k+1} r_k}}_{\bar{\epsilon}_r} - \delta\mathbf{r}_{r_k}^{v_{k+1} r_k}
\end{aligned} \tag{10}$$

The final expression for $\mathbf{H}_{k+1}$ is given in equation 11, where $\mathbf{J}$ represents the right Jacobian of $\mathbb{SO}(3)$.

$$\mathbf{H}_{k+1} = \begin{bmatrix} \mathbf{0}_{9\times3} & -\mathbf{J}\left(-\check{\boldsymbol{\phi}}_{r_k v_{k+1}}\right)^{-1} & \mathbf{0} & \mathbf{0}_{9\times3} \\ \mathbf{0}_{9\times3} & \mathbf{0} & -\mathbf{I} & \mathbf{0}_{9\times3} \end{bmatrix} \tag{11}$$

To perform the Extended Kalman Filter (EKF) iteratively, we initiate by updating $\mathbf{H}_{k+1,l+1}$, where l denotes the iteration. Upon updating $\mathbf{H}_{k+1,l+1}$, we subsequently update $\mathbf{K}_{k+1,l+1}$ and $\mathbf{x}_{k+1,l+1}$.

$$\mathbf{H}_{k+1,l+1} = \frac{\partial h\left(\mathbf{x}_{k+1,l}\right)}{\partial \delta \mathbf{x}_{k+1,l}} \tag{12}$$

$$\begin{aligned}
\mathbf{K}_{k+1,l+1} &= \check{\mathbf{P}}_{k+1}\mathbf{H}_{k+1,l}^T\left(\mathbf{H}_{k+1,l}\check{\mathbf{P}}_{k+1}\mathbf{H}_{k+1,l}^T + \mathbf{R}_{k+1}\right)^{-1} \\
\delta\hat{\mathbf{x}}_{k+1,l+1} &= \mathbf{K}_{k+1,l+1}\bar{\boldsymbol{\epsilon}}_{k+1}
\end{aligned} \tag{13}$$

At the end of the iteration, $\mathbf{P}_{k+1}$ undergoes an update. Following this, $\delta\hat{\mathbf{x}}_{k+1}$ is introduced into the predicted nominal states in accordance with their defined perturbations as outlined in Section III-A. This operation results in $\hat{\mathbf{x}}_{k+1}$. Importantly, it should be noted that, up to this point, the estimates for rotation and position in the reference frame, namely $\mathbf{C}_{r_k v_\tau}$ and $\mathbf{r}_{rk}^{v_\tau r_k}$, have not changed.

$$\hat{\mathbf{P}}_{k+1} = \left(\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_{k+1}\right)\check{\mathbf{P}}_{k+1} \tag{14}$$

Finally, all states' reference frames are shifted from $\underrightarrow{\mathcal{F}}_k$ to $\underrightarrow{\mathcal{F}}_{k+1}$ to obtain $\hat{\mathbf{x}}_{k+1,rk+1}$. For the next EKF iteration, the local vehicle pose is reset after being combined with the inertial pose. It's important to note that the velocity and biases remain unchanged since they are already represented in the $\underrightarrow{\mathcal{F}}_{r_{k+1}}$ frame.

$$\begin{aligned}\hat{\mathbf{C}}_{r_{k+1}i} &= \hat{\mathbf{C}}^T_{r_k v_{k+1}} \hat{\mathbf{C}}_{r_k i}, \quad \hat{\mathbf{g}}_{r_{k+1}} = \hat{\mathbf{C}}^T_{r_k v_{k+1}} \hat{\mathbf{g}}_{r_k}, \\ \hat{\mathbf{r}}^{ir_{k+1}}_{r_{k+1}} &= \check{\mathbf{C}}^T_{r_k v_{k+1}} \left(\hat{\mathbf{r}}^{ir_k}_{r_k} - \hat{\mathbf{r}}^{v_k r_k}_{r_k} \right), \\ \check{\mathbf{C}}_{r_{k+1} v_{k+1}} &= \mathbf{I}, \quad \check{\mathbf{r}}^{v_{k+1} r_{k+1}}_{r_{k+1}} = \mathbf{0} \end{aligned} \tag{15}$$

The state covariances must also be propagated to account for the operation described in equation 16.

$$\begin{aligned}\hat{\mathbf{P}}_{k+1,r_{k+1}} &= \mathbf{U}_{k+1} \hat{\mathbf{P}}_{k+1} \mathbf{U}^T_{k+1}, \\ \mathbf{U}_{k+1} &= \frac{\partial \delta \hat{\mathbf{x}}_{k+1,r_{k+1}}}{\partial \delta \hat{\mathbf{x}}_{k+1}} \end{aligned} \tag{16}$$

D. DI-EKF-VIO: DEEP ITERATED EXTENDED KALMAN FILTER

With deep learning, DI-EKF-VIO incorporates the architecture of an optical flow estimator into a part of the overall structure. Given two consecutive images in time as input, this network's task is to produce an optical flow estimation corresponding to the two initial images. This estimation is then fed into a regressor network, which is a convolutional network to compute the relative poses $\mathbf{z}_k$ and the uncertainty values $\mathbf{w}_k$. The architecture of the network is generally depicted as shown in Figure 2 below.

Through the partitioning of the network architecture into two distinct sub-networks, each serving unique roles and generating separate outputs, the training process gains robustness and flexibility. This segregated training approach enables independent enhancement of each network component, thereby contributing to an overall improvement in performance. Further elaboration on this methodology is provided in Section IV.

In the context of optical flow estimation, we utilized a well-established deep learning framework previously documented in scholarly literature, along with its pretrained parameters. Considering the delineation of optical flow estimation as a distinct problem domain, our principal objective is dedicated to augmenting visual-inertial odometry. In pursuit of this objective, we embraced the RAFT [27], recognized for its resilience and extensive adoption within the research community. Section IV provides a comprehensive examination of the influence exerted by various optical flow networks on ultimate results.

The architecture of the regressor network employed for output estimation resembles that of the ResNet network, akin to the approach outlined in [28]. Illustrated in Figure 3, each ResNet block maintains a consistent structure. However, within the DI-EKF-VIO framework, this network integrates spatial and channel attention blocks, as proposed in [29], alongside the ResNet architecture. The incorporation of

TABLE 1. Specification of ResNet utilized in the Regression network.

Layer	Output channel	Num. blocks	Stride	Padding
Conv1	64	3	2	1
Conv2	128	4	2	1
Conv3	128	6	2	1
Conv4	256	7	2	1
Conv5	256	3	2	1

spatial and channel attention mechanisms, following the methodology of [29], serves to augment the ResNet's output with focused attention. Our observations indicate that this amalgamation has contributed to enhanced regression outcomes. Furthermore, we provide detailed specifications of the ResNet network in Table 1.

The output of ResNet is then flattened and fed into a sequential of linear layer for regression purpose. The network's output consists of 12 elements, corresponding to the relative pose $\mathbf{z}_k = \begin{bmatrix} \tilde{\boldsymbol{\phi}}^T_{r_k v_{k+1}} \tilde{\mathbf{r}}^{v_{k+1} r_k T}_{rk} \end{bmatrix}^T \in \mathbb{R}^6$ and the uncertainty $\mathbf{w}_k = \begin{bmatrix} \mathbf{w}^T_{\mathbf{r}_k} & \mathbf{w}^T_{\phi_k} \end{bmatrix}^T \in \mathbb{R}^6$, where $\mathbf{w}_{\mathbf{r}_k}$ and $\mathbf{w}_{\phi_k}$ are the uncertainty values for the translation and rotation respectively.

To ensure positive definiteness for the output covariance matrix, we assume the uncertainties of the non-correlated measurements for each motion dimension. We apply Formula 17 element-wise to $\mathbf{w}_k$, resulting in the diagonal covariance matrix $\mathbf{R}_k$ as in equation 18. Here, w_i represents each element in the vector $\mathbf{w}_k$, σ_0 is the initial estimate of the noise standard deviation bias, and $\beta \in \mathbb{R}_{>0}$ is an adjusting parameter. The value of the tanh function is bounded between -1 and 1, and by varying β, we can control the degree of deviation of σ^2 from σ_0^2, allowing us to set reasonable lower and upper bounds for σ^2 values. Another characteristic is that this formula encourages small values for $\mathbf{w}_k$ because the derivative magnitude of the tanh function rapidly approaches zero as w_i moves far away from zero.

$$\sigma^2 = \sigma_0^2 10^{\beta \tanh(w_i)} \tag{17}$$

$$\mathbf{R}_k = diag(\sigma_0^2 10^{\beta \tanh(w_k)}) \tag{18}$$

To train the model, DI-EKF-VIO employs a combination of two different loss functions, L_r and L_a. L_r is a loss function directly applied to the output of the deep learning network, as depicted in equation 19. On the other hand, L_a is a loss function applied to the final output of the model (after passing through the deep learning network and EKF). L_r is used to guide the model to directly learn the relative poses of the object, while the L_a loss takes into account both the measured data and the associated uncertainties to guide the learning process of the network. By incorporating the measurement covariance matrix $\mathbf{R}_k$, the network can better understand the reliability of the measurements and adjust its predictions accordingly. This helps generate more accurate and stable absolute pose estimates, as it accounts for the uncertainty in

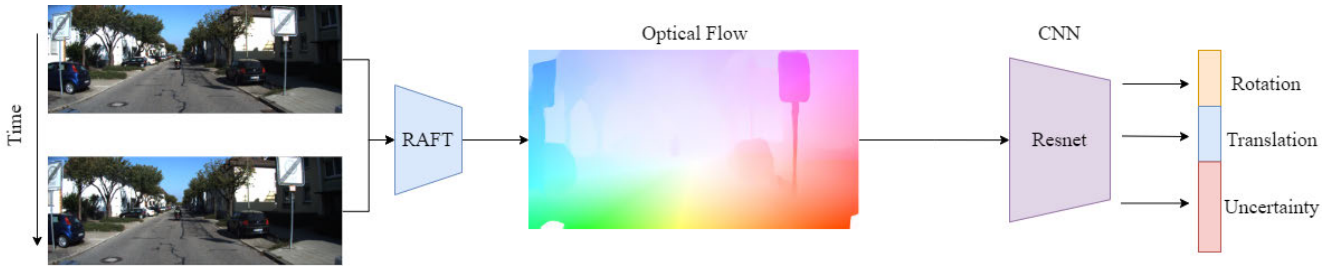

FIGURE 2. Measured optical flow is used to estimate relative poses $\mathbf{z}_k$ and the uncertainty value $\mathbf{w}_k$.

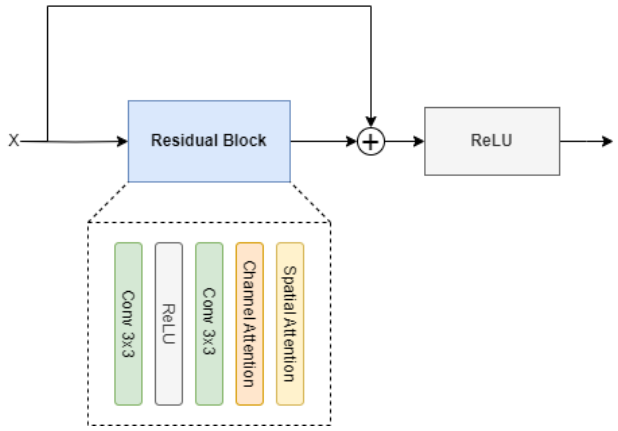

FIGURE 3. ResNet block used in the Network. The architecture follows the conventional ResNet18 Network, but adding the Spatial & Channel Attention Layer to each block to further improve results.

the measurements.

$$L_r = \sum_{k=1}^{N} \kappa_1 \mathbf{e}_{\boldsymbol{\phi}}^T \mathbf{e}_{\boldsymbol{\phi}} + \mathbf{e}_{\mathbf{r}}^T \mathbf{e}_{\mathbf{r}}$$
$$\mathbf{e}_{\boldsymbol{\phi}} = \tilde{\boldsymbol{\phi}}_{r_k v_{k+1}} - \boldsymbol{\phi}_{r_k v_{k+1}} \quad \mathbf{e}_{\mathbf{r}} = \tilde{\mathbf{r}}_{r_k}^{v_{k+1} r_k} - \mathbf{r}_{r_k}^{v_{k+1} r_k} \tag{19}$$

We utilize the loss function introduced in [8] for L_a. In contrast to the commonly used Mean Square Error (MSE) loss in [30] and [31], this particular loss function offers the advantage of preventing quaternion flips during the initial stages of training, where errors tend to be significant. Additionally, the expression $\mathbf{I} - \hat{\mathbf{C}}_{v_k i}^T \mathbf{C}_{v_k i}$ serves as an approximation of $\ln\left(\hat{\mathbf{C}}_{v_k i}^T \mathbf{C}_{v_k i}\right)$ when $\hat{\mathbf{C}}_{v_k i}$ closely matches $\mathbf{C}_{v_k i}$, providing a valid distance measure while bypassing the issue of differentiability near π.

$$L_a = \sum_{k=1}^{N} \left\| \hat{\mathbf{r}}_{v_k}^{i v_k} - \mathbf{r}_{v_k}^{i v_k} \right\|_2^2 + \kappa_2 \left\| \mathbf{I} - \hat{\mathbf{C}}_{v_k i}^T \mathbf{C}_{v_k i} \right\|_F^2 \tag{20}$$

Here, κ_2 is a tuning coefficient used to balance the role of components in the loss function. We use the squared Euclidean norm to measure the error in absolute translation estimates $\hat{\mathbf{r}}_{v_k}^{i v_k}$ compared to the actual translation $\mathbf{r}_{v_k}^{i v_k}$. Simultaneously, we employ the Frobenius norm to measure the rotation error between $\hat{\mathbf{C}}_{v_k i}^T \mathbf{C}_{v_k i}$ and the identity matrix $\mathbf{I}$.

To enable a metrically scaled pose initialization, a scale parameter λ is augmented to the end of the state vector (similar to [9]). The continuous time dynamics model for the scale is $\dot{\lambda} = 0$ and error state is $\delta\dot{\lambda} = 0$. The scale factor is applied to the IMU translation state, through $\lambda \mathbf{r}_{r_k}^{v_{k+1} r_k}$, prior to computing $\boldsymbol{\epsilon}_r$ within the measurement model. The rotation measurement is unchanged. The measurement Jacobian $\mathbf{H}_{k+1}$ becomes

$$\begin{bmatrix} \mathbf{0}_{9\times3} & -\mathbf{J}\left(-\check{\boldsymbol{\phi}}_{r_k v_{k+1}}\right)^{-1} & \mathbf{0} & \mathbf{0}_{9\times2} & \mathbf{0}_{9\times1} \\ \mathbf{0}_{9\times3} & \mathbf{0} & -\lambda\mathbf{I} & \mathbf{0}_{9\times2} & r_{r_k}^{v_{k+1} r_k} \end{bmatrix} \tag{21}$$

IV. EXPERIMENTS

We begin experimenting DI-EKF-VIO on 2 different datasets with different settings: KITTI [32] and EuroC [33].

DI-EKF-VIO is implemented using PyTorch. The model utilizes the Adam optimizer with coefficients $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a learning rate of 10^{-4}. In our investigation of DI-EKF-VIO, we undertook a methodical examination of the Adam optimizer's hyperparameters to elucidate their influence on the system's efficacy. However, given the considerable scale of the dataset and constraints pertaining to our training resources, we opted not to extensively explore alternative optimizers. Rather, we selected the Adam optimizer, recognized for its versatility in model training. Throughout our experimentation, we systematically adjusted the values of β_1, β_2, and the learning rate while maintaining consistency in other model parameter. We observed that setting β_1 to values lower than 0.9 resulted in slower convergence during training, leading to prolonged optimization times and potentially suboptimal performance. Conversely, increasing β_1 beyond 0.9 accelerated convergence initially but often led to overshooting or instability issues in later epochs. Similarly, variations in β_2 impacted the smoothness of optimization trajectories, with excessively high values causing erratic behavior.

Moreover, our experiments highlighted the sensitivity of the model to the learning rate. While a learning rate of 10^{-4} provided satisfactory results in most cases, we found that adjusting this parameter could fine-tune the balance between convergence speed and final accuracy. Higher learning rates facilitated faster initial progress but risked

overshooting optimal solutions, whereas lower rates required more iterations to converge but yielded more precise results.

For the optical flow estimator network, DI-EKF-VIO employs RAFT [27] along with its pre-trained weights. Additionally, the project implements other optical flow estimation networks for comparison, aiming to understand the importance of optical flow quality in relation to pose estimation accuracy.

During training, the project divides large image sequences into smaller subsequences of 6 frames. This allows for faster training because a larger batch size can be used. However, it also results in fewer updates to the Extended Kalman Filter compared to using longer subsequences of 32 frames as seen in Deep-EKF-VIO [8]. Nonetheless, since DI-EKF-VIO uses an iterative EKF filter instead of the traditional EKF in Deep-EKF-VIO, a sufficient number of EKF updates is still ensured. The IEKF is iterated 6 times in our settings.

Data augmentation techniques are also used to prevent overfitting, such as varying the starting point of the subsequences. In addition to cutting sequences of length 6 starting from frame 0, the project also cuts sequences of the same length starting from frame 3. This effectively doubles the input data while ensuring diversity due to the different starting points of the subsequences.

Furthermore, data augmentation includes techniques such as introducing random noise to brightness, contrast, saturation, and color levels. Data is also augmented by horizontally flipping frames and temporally reversing the data. The left-right flipping generates examples with turning scenarios, while temporal reversing prevents the network from having a forward-moving bias since vehicles often move forward. These data augmentation techniques overall help to improve the model's generalization capabilities and reduce the risk of overfitting.

A. KITTI DATASET

The KITTI odometry dataset [32] comprises outdoor driving environments on urban streets primarily involving 3-DoF planar motions, with speeds ranging from complete standstill to 90 km/h. Some scenes taken from the dataset is shown in figure 4. The groundtruth data is provided by the output of the GPS/IMU localization unit, which is then projected onto the coordinate system of the left camera post-rectification. Sequences 00, 02, and 05 are excluded from the evaluation process due to missing IMU data for several seconds at various timestamps. However, valid portions of these sequences are retained for training data. The evaluation metric used is the KITTI standard evaluation metric, calculating errors in translation and rotation per unit of distance traveled. These errors are evaluated at distances of 100, 200, 300, 400, 500, 600, 700, and 800 meters. The unit for translation error is percentage, and the unit for rotation error is degrees per 100 meters ($_o$/100m). The average error for these data sequences is computed by taking the average of all segments from 100 to 800 meters across all test sequences.

FIGURE 4. Images taken from KITTI Dataset.

Longer sequences have a more significant impact on the average error calculation.

Although evaluating VIO methods on the KITTI dataset is less common due to incomplete inertial data, we prioritized comparisons with methods similar to ours, those considered state-of-the-art at the time of our research, and those with publicly available implementations for validation.

Therefore, we selected ORB-MSF [34], DeepVIO [6], Deep-EKF-VIO [8], TartanVO [28],for comparison. ORB+MSF is a traditional sensor fusion method serving as a benchmark for classical approaches. It utilizes ORB for visual feature extraction and the popular MSF for sensor fusion with inertial data via EKF. DeepVIO is a state-of-the-art learning-based VIO method demonstrating exceptional performance on the KITTI dataset. It leverages CNNs and LSTM networks for sensor data processing, along with a self-supervised training method. Deep-EKF-VIO is a hybrid method combining a DeepVO model for image feature extraction and LSTM for IMU data, with EKF for sensor fusion. TartanVO stands out for utilizing only a monocular camera. We compared against TartanVO because our feature extraction deep network is based on it. This comparison allows us to assess the performance gains achieved through sensor fusion with IMU data using our proposed IEKF method. Results are shown in table 2

Deep-EKF-VIO [8] achieves high accuracy results in terms of r_{err} on several sequences, but it significantly struggles on sequence 01. This can be attributed to the fact that in Deep-EKF-VIO, the authors apply LSTM [35] similar to [30] to estimate relative poses and propagate this information throughout the estimation of different data sequences. This allows the model to retain velocity information across sequences, which leads to better pose estimation results, given that the robot's velocities are similar across those sequences. However, in sequence 01, the robot moves at a high speed of 90 km/h, which is vastly different from the velocities in the other sequences. As a result, Deep-EKF-VIO incorrectly estimates the relative pose of the robot in sequence 01, causing the results to deviate significantly from the ground truth. On the other hand, DI-EKF-VIO does not employ LSTM or any other temporal information but directly estimates relative poses based on two consecutive input images. Although this approach ignores the time constraints between poses, it can lead to better results for sequence 01 compared to Deep-EKF-VIO.

Table 2 also clearly shows that combining both IMU and camera information yields better results than using

TABLE 2. Experimental results on KITTI dataset. Unit of t_{err} and r_{err} are [%] and [°/100m] correspondingly. "-" indicate unavailable or unrecored results. Best results are written bold in the table.

Sequence	ORB+MSF		DeepVIO		Deep-EKF-VIO		TartanVO		Ours (Visual Only)		Ours (Visual + IMU)	
	t_{err}	r_{err}	t_{err}	r_{err}	t_{err}	r_{err}	t_{err}	r_{err}	t_{err}	r_{err}	t_{err}	r_{err}
01	2.03	0.199	4.52	1.44	25.24	0.159	-	-	**0.94**	0.347	0.94	**0.127**
04	0.90	0.094	-	-	0.34	**0.007**	-	-	1.23	0.599	**0.23**	0.010
06	1.51	0.269	-	-	2.14	**0.156**	4.72	2.95	2.20	0.699	**0.75**	0.220
07	1.73	0.478	2.71	1.66	**0.93**	**0.305**	4.32	3.41	2.75	0.961	1.86	0.330
08	1.30	0.319	2.13	1.02	**1.23**	0.255	-	-	3.44	0.897	1.41	**0.202**
09	1.38	0.276	1.38	1.12	**1.09**	0.253	6.0	3.11	2.41	0.581	4.05	**0.250**
10	1.40	0.316	**0.85**	1.03	1.17	**0.248**	6.89	2.73	3.38	1.12	1.03	0.261
Average	**1.37**	0.313	3.72	1.58	-	-	5.48	3.05	2.83	0.781	2.28	**0.226**

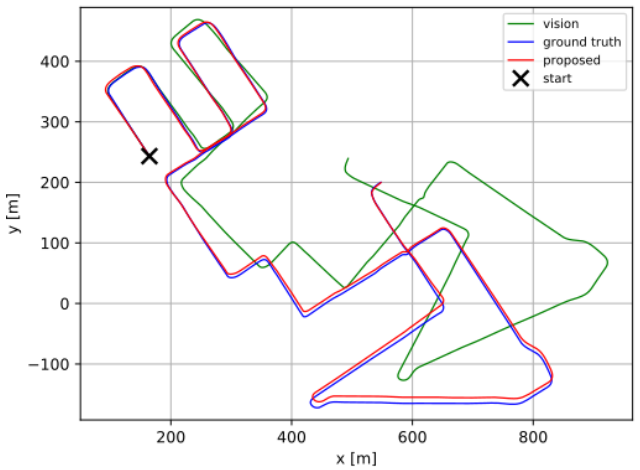

FIGURE 5. Trajectory result on sequence KITTI 08.

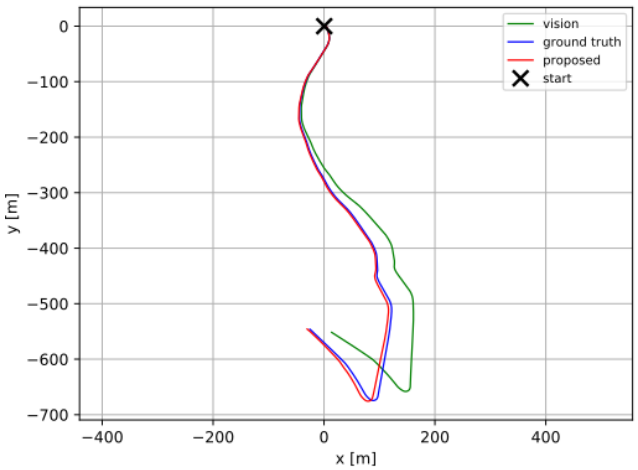

FIGURE 6. Trajectory result on sequence KITTI 10.

only the camera. Given the localized nature of the problem settings, wherein no overarching global reference, such as GPS, is available, the deep learning model's reliance solely on camera input results in a cumulative error in rotational estimation, consequently leading to a substantial deviation from ground truth over time. The inclusion of IMU data in the fusion process facilitates precise angular rotation estimation, thereby mitigating the progressive drift inherent in the estimation process. Figures 5 and 6 illustrate estimation results for some sequences, comparing the use of EKF versus not using EKF.

It is evident that without using EKF, the results are prone to cumulative drift due to local errors. This cumulative drift becomes more significant as the trajectory progresses. Drifted results can lead to substantial deviations, as seen in Figures 5 and 6, where the trajectory is initially correct but deviates as it moves away from the starting point.

B. EUROC DATASET

The EuRoC MAV dataset [33] comprises 11 indoor sequences captured using a Skybotix stereo VI sensor mounted on a small Unmanned Aerial Vehicle (UAV) commonly known as a Micro Aerial Vehicle (MAV). Some images taken from the datasets is displayed in figure 7. The available data includes grayscale images, accelerometer, gyroscope readings, and ground truth measurements obtained from the Vicon motion capture system and Leica MS50 laser tracker. One significant challenge in this dataset, crucial for our project, is synchronizing the sensor data with the motion ground truth, as they are recorded using different systems. This means that the input images and the corresponding camera ego-motion are not perfectly aligned. For instance, while the images are captured by the onboard system, the position data is gathered by the Vicon system. Despite both sources having global timestamps, there is inconsistency in aligning the image frames with the positional data. Unlike the KITTI datasets, where the camera motion is consistently forward with a fixed speed, the motion of the MAV (Micro Aerial Vehicle) can vary in direction with minimal displacement per frame. In such scenarios, the actual motion may be smaller than the inherent noise in the data. It is organized into three distinct scenes: Machine Hall (MH) containing 5 sequences, Vicon Room 1 (V1) consisting of 3 sequences, and Vicon Room 2 (V2) also comprising 3 sequences. Sequence V2 03 has been excluded from the dataset due to issues related to the acquisition of image data at the anticipated rate.

Unlike the KITTI dataset, the EuroC dataset presents a more challenging scenario due to the presence of significant

FIGURE 7. Images taken from EuroC Dataset.

relative errors in the IMU data. This is because the EuroC dataset contains grayscale single-channel images. To use the same pre-trained optical flow estimation network weights as in KITTI, the grayscale images are duplicated three times to create RGB-like images. This is necessary to match the input format of the optical flow estimation network used in KITTI.

Except for sequence V2 03, which was omitted due to issues with image acquisition rate, all remaining sequences from the EuroC dataset are used. Sequences MH 05 and V1 02 are utilized for validation data, while all other sequences are used for training data. The challenges posed by the EuroC dataset, including its large IMU error, make it a more demanding dataset compared to KITTI.

As a result, unlike the results obtained from the KITTI dataset, the EuroC dataset yields considerably worse results compared to traditional methods. The experimental results are compared with results proposed in [8], [9], [11], and [28]. It is important to note that the EuroC dataset presents greater challenges due to its significant IMU errors and other complexities, making it harder for the DI-EKF-VIO approach to outperform traditional methods in this particular context. Vins-Mono [11] is a state-of-the-art classical VIO model, employs a tightly coupled optimization approach, jointly optimizing the system state and 3D positions of tracked features. Deep-EKF-VIO [8] while performs well on KITTI, its performance on EuroC is lower. This raises questions about the effectiveness of the hybrid EKF approach in more challenging datasets. Self-EKF-VIO [9] builds upon Deep-EKF-VIO by incorporating self-supervised training. The comparison is demonstrated in Table 3.

The predicted trajectories of several sequences in the EuroC dataset are shown in Figures 8 and 9. The integration of deep learning with the Extended Kalman Filter presents a notable observation: the incorporation of EKF tends to yield less accurate predicted trajectories when compared to instances where EKF is not utilized. A plausible explanation for this phenomenon lies in the inferior quality of IMU data within the EuroC dataset as opposed to the IMU data contained within the KITTI dataset. The EuroC dataset suffers from substantial IMU errors. The high noise levels in EuroC's IMU measurements lead to filter divergence. The EKF tend to underestimates the noise in the IMU data, causing it to place undue trust in these erroneous measurements. This results in the EKF propagating noise and generating inaccurate state estimates, ultimately reflected in the degraded trajectory predictions seen in Figures 8 and 9.

One potential limitation of our approach on the EuroC dataset is the use of grayscale images. The pre-trained optical flow estimation network was originally designed for RGB images. While we convert the grayscale EuroC images to pseudo-RGB by replicating channels, this process might introduce artifacts that could negatively affect network performance. Future work should investigate the extent of this impact and potentially explore techniques for handling grayscale data within the deep learning framework.

Furthermore, the EuroC dataset presents distinct challenges compared to the KITTI dataset. KITTI captures images from a car driving in an urban environment, offering a relatively stable viewpoint. In contrast, the EuroC dataset captures indoor environments from a flying UAV, which experiences significantly greater freedom of movement. This can lead to sequences with rapid rotations, a known difficulty for visual odometry methods. The limited field of view of the camera in these scenarios may result in insufficient unique visual features for accurate tracking. The noisy IMU data in EuroC further amplifies this issue, as the EKF struggles to compensate for the rapid changes.

Additionally, the EuroC dataset may contain sequences with varying or low illumination conditions. Such conditions can negatively impact feature extraction and matching within the visual odometry component. When combined with the noisy IMU data, these scenarios can lead to significant errors in the estimated trajectories.

Finally, environments with repetitive textures pose another challenge. These environments offer minimal unique visual features for the network to track, potentially causing the visual odometry component to drift. The noisy IMU data in EuroC exacerbates this issue in DI-EKF-VIO.

Nevertheless, when we examine Table 3, we can see that even without utilizing EKF, the combination of deep learning and camera data still achieves better outcomes than networks using IMU information and other visual odometry

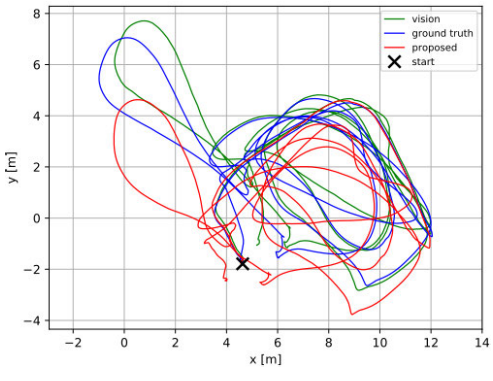

FIGURE 8. Trajectory result on sequence EuroC MH03.

TABLE 3. Experimental results on EuroC dataset. Unit of results is meter, with bold black number are best results, and red numbers indicate best deep learning results.

Sequence	VINS-Mono	Self-EKF-VIO	Deep-EKF-VIO	TartanVO	Ours (Vision Only)	Ours (Vision + IMU)
MH 01	0.27	0.51	1.17	-	0.66	0.96
MH 02	**0.12**	0.78	1.56	-	0.73	0.81
MH 03	**0.13**	0.69	1.89	-	0.46	1.16
MH 04	**0.23**	1.00	2.12	0.74	0.52	1.13
MH 05	**0.35**	0.80	1.96	0.68	0.39	1.10
V1 01	**0.07**	0.43	2.07	-	0.15	0.96
V1 02	**0.10**	0.61	2.20	0.45	0.26	1.41
V1 03	**0.13**	0.72	2.83	0.64	0.35	0.94
V2 01	**0.08**	0.20	1.49	0.67	0.15	0.47
V2 02	**0.08**	0.81	2.22	1.04	0.45	0.70

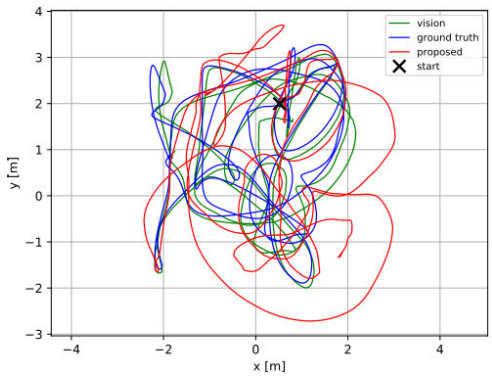

FIGURE 9. Trajectory result on sequence EuroC V102.

networks. This highlights the superior capabilities of the deep learning network in DI-EKF-VIO when compared to other approaches.

C. OPTICAL FLOW PRETRAIN NETWORK

DI-EKF-VIO employs the architecture and weights of the RAFT optical flow estimation network RAFT [27] as a pretrained network for training. RAFT was chosen as the pretrained network due to its accurate optical flow estimation results and robustness on unobserved data. Hereafter, we will modify the pretrained network architecture to assess the significance of accurate optical flow estimation in relation to relative pose estimation. We will solely compare the estimation results of the deep learning network without using EKF for this experiment. However, it can be implied that a more accurate relative pose estimation from the deep learning network will yield even more precise pose estimation results across the system.

We conduct a comparison between results obtained using the RAFT optical flow estimation network and results from using the RAFT-S [27] and GMFlow [36] optical flow estimation networks. This comparison is performed on sequences KITTI 01, 04, 06, and 10, as shown in table 4.

It's evident that DI-EKF-VIO, utilizing the pretrained network RAFT-L as introduced in [27], achieves the most promising results among the three compared networks. This observation can be attributed to the optical flow estimation

TABLE 4. Compared KITTI result using different optical flow pretrain network.

Sequence	RAFT-S		GMFlow		RAFT-L (Proposed)	
	t_{err}	r_{err}	t_{err}	r_{err}	t_{err}	r_{err}
01	1.39	0.619	1.69	0.128	**1.03**	**0.128**
04	0.31	0.011	0.87	0.012	**0.20**	**0.011**
06	0.83	0.217	0.98	0.273	**0.83**	**0.217**
07	1.70	0.332	**1.61**	0.360	1.64	**0.330**
08	2.78	0.266	3.71	0.349	**1.38**	**0.221**
09	**4.11**	0.254	4.47	0.258	4.21	**0.251**
10	1.03	0.267	1.07	0.274	**1.01**	**0.255**
Average	2.85	0.233	2.73	0.259	**2.27**	**0.226**

performance that the authors of the paper reported using RAFT-L, which surpasses both of the other networks. From this, it can be deduced that even though the differences might not be significant, a better-performing pretrained optical flow estimation network can lead to improved relative pose estimation results.

D. RUNTIME COMPARISON

Intuitively, increasing the number of iterations has a positive impact on the system's accuracy because it allows for improved estimation with each iteration of IEKF. However, it's important to note that as the number of iterations goes up, the runtime of the IEKF also increases significantly. Thus, there exists a trade-off between accuracy and runtime.

Nevertheless, our testing has led us to a crucial observation: increasing the number of iterations does not always result in a corresponding increase in accuracy, as demonstrated in table 5. We posit that this discrepancy occurs because as the number of iterations increases, the IEKF's linearization process becomes more susceptible to bias and errors. Consequently, the overestimation caused by bias in the IEKF aggravate, leading to a decline in accuracy. To address this issue, we conducted experiments with varying numbers of iterations to strike a balance between maintaining accuracy and achieving a sufficiently low runtime for future real-time implementation.

As in table 5, these results illustrate the trade-off between runtime and accuracy as the number of iterations increases. With fewer iterations, the runtime is low, but accuracy is

TABLE 5. Compared KITTI result using different number of iteration settings.

Num. Iterations	Trans. Err (%)	Runtime/frame (s)
1	2.52	**0.064**
6	2.28	0.084
12	**2.27**	0.146

only moderate. As the number of iterations grows to 6, accuracy significantly improves, although at the cost of a moderate increase in runtime. However, with 12 iterations, while accuracy continues to increase, the runtime becomes high, and there is a noticeable decrease in accuracy due to issues related to overestimation caused by bias. Therefore, choosing the appropriate number of iterations depends on striking a balance between runtime constraints and achieving the desired level of accuracy for your specific application.

E. PRACTICAL DISCUSSION

Analysis of data from the KITTI and EuRoC datasets underscores the potential of DI-EKF-VIO for autonomous navigation tasks in vehicles and UAVs. This approach offers a compelling solution for enhancing navigation precision and efficiency by fusing monocular camera data with inertial measurements from an IMU. Compared to prevalent LiDAR-based systems in the autonomous vehicle industry, DI-EKF-VIO presents several advantages. By leveraging readily available cameras and IMU, which are generally cheap and easy to acquire, DI-EKF-VIO is a more cost-effective solution. Additionally, cameras and IMU are generally smaller and lighter than LiDAR sensors, leading to a more compact system design - particularly beneficial for UAVs with limited payload capacity.

Real-time operation is also critical. The use of iterated EKFs enhances performance but introduces a significant computational burden. In our work, we have explored balancing iterations with accuracy by modifying the number of iteration in the IEKF. As shown in table 5, utilizing iteration of 6 can achieve real time computation, with the FPS of 12.

Another crucial factor for deploying DI-EKF-VIO in real-world scenarios with diverse environments is its generalizability. The current evaluation utilizes the KITTI and EuRoC datasets, which offer a variety of conditions. However, to solidify the claims of DI-EKF-VIO's effectiveness across a wider range of situations, further validation with datasets encompassing extreme weather conditions, off-road terrains, and highly dynamic scenarios is necessary. Additionally, exploring the performance of DI-EKF-VIO with different sensor configurations, such as varying camera resolutions or incorporating additional sensor modalities, would broaden its applicability to a wider range of VIO applications.

One of the main challenges for achieving generalizability is the dependence of DI-EKF-VIO on training data for parameter tuning. This is a common hurdle faced by many visual-inertial odometry methods, particularly those with deep learning elements. The effectiveness of DI-EKF-VIO can be significantly impacted by the quality and quantity of data used. To achieve robustness in diverse environments, the system may require a substantial amount of training data encompassing a wide range of conditions.

Recent work [9] proposes a potential solution to this challenge. Their approach explores training DI-EKF-VIO on synthetic datasets. Synthetic datasets offer a significant advantage in terms of environmental variability. By generating diverse and controlled virtual environments, researchers can train the system on a much larger and more varied dataset compared to real-world data collection. This approach holds promise for improving the generalizability and robustness of DI-EKF-VIO in real-world applications.

Finally, real-world sensor measurements are inherently susceptible to noise from both cameras and IMUs. Sensor noise significantly impacts DI-EKF-VIO's performance, as evidenced in the EuroC dataset section, where high-noise data led to poor EKF performance. This is a well-known weakness of EKFs. We will investigate alternative filtering techniques specifically designed to handle sensor noise and improve DI-EKF-VIO's robustness in noisy environments. This could involve incorporating noise models into the EKF framework or exploring Kalman filter variants better suited for handling non-Gaussian noise.

V. CONCLUSION AND FUTURE RESEARCH

DI-EKF-VIO is a novel visual-inertial odometry (VIO) system that combines a deep learning network with a Robocentric-Iterated Extended Kalman Filter (EKF) for accurate relative pose estimation. The deep learning network provides pose estimates and uncertainties, which are fused with IMU data in the Robocentric Iterated EKF block to obtain absolute robot poses. DI-EKF-VIO's performance was evaluated on both KITTI and EuroC datasets. On KITTI, DI-EKF-VIO achieved promising results with an average translation error of 2.27% and rotational error of 0.226 degrees, outperforming other deep learning-based VIO methods and competing favorably with traditional approaches. Although less favorable on EuroC, DI-EKF-VIO still outperformed other deep learning methods.

To enhance the performance of DI-EKF-VIO's deep learning components for relative pose estimation, future research will explore a multifaceted approach. Novel deep learning architectures, inspired by recent breakthroughs in computer vision like transformers or GAN, hold promise for significant improvements.

Furthermore, addressing the challenge of data scarcity, a common hurdle in deep learning, will be crucial. Unsupervised and self-supervised learning techniques offer a compelling solution, allowing DI-EKF-VIO to leverage unlabeled data and alleviate the burden of meticulously labeled training sets.

Additionally, researchers are investigating the integration of statistical and Bayesian methods. This would enable DI-EKF-VIO to move beyond a single point estimate for pose

and instead quantify the uncertainty associated with each estimation. This probabilistic approach would lead to a more robust and reliable understanding of the environment.

Finally, synthetic data augmentation is emerging as a promising avenue for future development. By meticulously crafting simulated datasets, researchers can provide a vast and diverse training ground for deep learning models. This approach has the potential to significantly improve accuracy and generalizability without the need for extensive real-world data collection.

ACKNOWLEDGMENT

(Khac Duy Nguyen and Dinh Tuan Tran contributed equally to this work.)

REFERENCES

[1] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2007, pp. 3565–3572.

[2] E. S. Jones and S. Soatto, "Visual-inertial navigation, mapping and localization: A scalable real-time causal approach," *Int. J. Robot. Res.*, vol. 30, no. 4, pp. 407–430, Apr. 2011.

[3] M. Li and A. I. Mourikis, "High-precision, consistent EKF-based visual-inertial odometry," *Int. J. Robot. Res.*, vol. 32, no. 6, pp. 690–711, May 2013.

[4] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *Int. J. Robot. Res.*, vol. 34, no. 3, pp. 314–334, 2015.

[5] H. J. Kashyap, C. C. Fowlkes, and J. L. Krichmar, "Sparse representations for object- and ego-motion estimations in dynamic scenes," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 6, pp. 2521–2534, Jun. 2021.

[6] S. Wang, R. Clark, H. Wen, and N. Trigoni, "DeepVO: Towards end-to-end visual odometry with deep recurrent convolutional neural networks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 2043–2050.

[7] R. Li, S. Wang, Z. Long, and D. Gu, "UnDeepVO: Monocular visual odometry through unsupervised deep learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 7286–7291.

[8] C. Li and S. L. Waslander, "Towards end-to-end learning of visual inertial odometry with an EKF," in *Proc. 17th Conf. Comput. Robot Vis. (CRV)*, May 2020, pp. 190–197.

[9] B. Wagstaff, E. Wise, and J. Kelly, "A self-supervised, differentiable Kalman filter for uncertainty-aware visual-inertial odometry," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics (AIM)*, Jul. 2022, pp. 1388–1395.

[10] S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart, "A robust and modular multi-sensor fusion approach applied to MAV navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 3923–3929.

[11] 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.

[12] T. A. Ciarfuglia, G. Costante, P. Valigi, and E. Ricci, "Evaluation of non-geometric methods for visual odometry," *Robot. Auto. Syst.*, vol. 62, no. 12, pp. 1717–1730, Dec. 2014.

[13] R. Roberts, H. Nguyen, N. Krishnamurthi, and T. Balch, "Memory-based learning for visual odometry," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2008, pp. 47–52.

[14] H. Zhan, R. Garg, C. S. Weerasekera, K. Li, H. Agarwal, and I. M. Reid, "Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 340–349.

[15] Z. Yin and J. Shi, "GeoNet: Unsupervised learning of dense depth, optical flow and camera pose," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1983–1992.

[16] A. Ranjan, V. Jampani, L. Balles, K. Kim, D. Sun, J. Wulff, and M. J. Black, "Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12232–12241.

[17] J. Graeter, A. Wilczynski, and M. Lauer, "LIMO: LiDAR-monocular visual odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 7872–7879.

[18] T. Shan, B. Englot, C. Ratti, and D. Rus, "LVI-SAM: Tightly-coupled LiDAR-visual-inertial odometry via smoothing and mapping," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 5692–5698.

[19] B. Li, M. Hu, S. Wang, L. Wang, and X. Gong, "Self-supervised visual-LiDAR odometry with flip consistency," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 3843–3851.

[20] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. V. D. Smagt, D. Cremers, and T. Brox, "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2758–2766.

[21] X. Han, Y. Tao, Z. Li, R. Cen, and F. Xue, "SuperPointVO: A lightweight visual odometry based on CNN feature extraction," in *Proc. 5th Int. Conf. Autom., Control Robot. Eng. (CACRE)*, Sep. 2020, pp. 685–691.

[22] 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, vol. 32, no. 1, pp. 1–24.

[23] W. Liu, D. Caruso, E. Ilg, J. Dong, A. I. Mourikis, K. Daniilidis, V. Kumar, and J. Engel, "TLIO: Tight learned inertial odometry," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 5653–5660, Oct. 2020.

[24] J. Zhang and S. Singh, "LOAM: LiDAR odometry and mapping in real-time," in *Robot Science System X*. Berkeley, CA, USA: Univ. of California, Jul. 2014, pp. 1–9.

[25] Q. Li, S. Chen, C. Wang, X. Li, C. Wen, M. Cheng, and J. Li, "LO-Net: Deep real-time LiDAR odometry," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8465–8474.

[26] 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.

[27] Z. Teed and J. Deng, "RAFT: Recurrent all-pairs field transforms for optical flow," in *Proc. Eur. Conf. Comput. Vis.*, Aug. 2020, pp. 402–419.

[28] W. Wang, Y. Hu, and S. Scherer, "Tartanvo: A generalizable learning-based VO," in *Proc. Conf. Robot Learn.*, 2021, pp. 1761–1772.

[29] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 3–19.

[30] S. Wang, R. Clark, H. Wen, and N. Trigoni, "End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks," *Proc. Int. J. Robot. Res.*, vol. 37, pp. 513–542, 2018.

[31] A. Kendall, M. Grimes, and R. Cipolla, "PoseNet: A convolutional network for real-time 6-DOF camera relocalization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2938–2946.

[32] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[33] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The EuRoC micro aerial vehicle datasets," *Int. J. Robot. Res.*, vol. 35, no. 10, pp. 1157–1163, Sep. 2016.

[34] 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.

[35] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[36] H. Xu, J. Zhang, J. Cai, H. Rezatofighi, and D. Tao, "GMFlow: Learning optical flow via global matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 8111–8120.

KHAC DUY NGUYEN received the degree from Hanoi University of Science and Technology, in 2023. In 2022, he started his research with the Intelligent Vision System and Robotics Laboratory. While at university, he was recognized for his strong problem-solving skills and aptitude for machine learning. Upon the completion of his Engineering degree, he aspires to contribute to cutting-edge AI research and development. His research interests include computer vision, intelligent robots, and deep learning. To expand his knowledge in his research areas, he took additional coursework in artificial intelligence, robotics engineering, and neural networks.

DINH TUAN TRAN (Member, IEEE) received the B.E., M.E., and Ph.D. degrees in information science and engineering from Ritsumeikan University, Japan, in 2012, 2016, and 2019, respectively. From 2019 to 2020, he was a Postdoctoral Researcher with the College of Information Science and Engineering, Ritsumeikan University, where he is currently an Assistant Professor with the College of Information Science and Engineering. From 2017 to 2019, he was a short-term Visiting Researcher with the Technical University of Munich, Munich, Germany, in 2017, Budapest University of Technology and Economics, Budapest, Hungary, in 2018, The University of Auckland, Auckland, New Zealand, in 2018, the University of Greenwich, London, U.K., in 2018, Cardiff University, Cardiff, U.K., in 2018, and The University of Melbourne, Melbourne, Australia, in 2019. His research interests include machine learning, image processing, computer vision, robot vision, reinforcement learning, imitation learning, human process modeling, and medical imaging.

VAN QUYEN PHAM received the Dipl.Eng. degree from Hanoi University of Science and Technology (HUST), Vietnam, in 2023. From 2019 to 2023, he served as a Research Assistant at the Intelligent Vision System and Robotics Laboratory, School of Electrical and Electronic Engineering (SEEE), HUST. He has now transitioned to developing human-machine interface (HMI) applications for mid-range vehicles. His research interests encompass various areas, including computer vision, machine learning, deep learning for robotics; meta-learning; and telecommunication systems.

DINH TUAN NGUYEN graduated from the School of Electrical and Electronic Engineering, Hanoi University of Science and Technology, Vietnam, in 2023. Previously, from 2019 to 2023, he served as a Research Assistant at Intelligent Vision System and Robotics Laboratory (IVSR Lab), Hanoi University of Science and Technology. Throughout his tenure, his research primarily focused on artificial intelligence (AI), machine learning (ML), deep learning (DL), and their applications in the realm of robotics.

KATSUMI INOUE (Member, IEEE) received the D.Eng. degree from Kyoto University, in 1993. He is currently a Professor with the Principles of Informatics Research Division, National Institute of Informatics, and the Informatics Program, Graduate Institute for Advanced Studies, SOKENDAI, and a Visiting Professor with the School of Computing, Tokyo Institute of Technology. His research interests include artificial intelligence, logic programming, and computer science in general.

JOO-HO LEE (Senior Member, IEEE) received the B.E. and M.E. degrees in electrical engineering from Korea University, Seoul, South Korea, in 1993 and 1995, respectively, and the Ph.D. degree in electrical engineering from The University of Tokyo, Tokyo, Japan, in 1999. He is currently a Professor with the Department of Information Science and Engineering, Ritsumeikan University, Shiga, Japan. From 1999 to 2003, he was a JSPS Postdoctoral Researcher with the Institute of Industrial Science, The University of Tokyo. From 2003 to 2004, he was a Research Associate with Tokyo University of Science, Japan, and in 2004, he joined Ritsumeikan University as an Associate Professor. From 2008 to 2009, he was a Visiting Scholar with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA. In 2017, he was a Research Professor with the Department of Mechanical Engineering, Korea University. His research interests include intelligent environments, intelligent robots, computer vision, machine learning, and medical/healthcare applications. He is a member of the RSJ, JSME, SICE, HIS, IEICE, KROS, and IEEJ.

ANH QUANG NGUYEN received the Dipl.Eng. and M.Sc. degrees from the School of Electronics and Telecommunication (SET), Hanoi University of Science and Technology, Vietnam (HUST), in 2011 and 2013, respectively, and the Ph.D. degree from Ritsumeikan University, Japan, in 2017. His Ph.D. research focuses on ultra-high-speed image sensor development. Since September 2017, he has been an Assistant Professor with the School of Electrical and Electronics Engineering (SEEE), HUST, where he has been leading the Intelligent Vision System and Robotics Laboratory, SEEE, since 2019. His research interests include the IoT systems, AI and computer vision technologies, autonomous drones and robotics, and ultra-high-speed imaging.