Photometric Visual-Inertial Navigation With Uncertainty-Aware Ensembles

Jae Hyung Jung, *Student Member, IEEE*, Yeongkwon Choe, and Chan Gook Park, *Member, IEEE*

***Abstract*—In this article, we propose a visual-inertial navigation system that directly minimizes a photometric error without an explicit data-association. We focus on the photometric error parametrized by pose and structure parameters that is highly nonconvex due to the nonlinearity of image intensity. The key idea is to introduce an *optimal intensity gradient* that accounts for a projective uncertainty of a pixel. Ensembles sampled from the state uncertainty contribute to the proposed gradient and yield a correct update direction even in a bad initialization point. We present two sets of experiments to demonstrate the strengths of our framework. First, a thorough Monte Carlo simulation in a virtual trajectory is designed to reveal robustness to large initial uncertainty. Second, we show that the proposed framework can achieve superior estimation accuracy with efficient computation time over state-of-the-art visual-inertial fusion methods in a real-world UAV flight test, where most scenes are composed of a featureless floor.**

***Index Terms*—Iterated extended Kalman filter (EKF), matrix Lie groups, stochastic linearization, visual-inertial navigation.**

I. Introduction

VISUAL navigation is a fundamental building block for higher-level tasks such as autonomous flight in space exploration [1] and semantic perception [2]. While a camera provides rich information for localization and surrounding perception, an inertial measurement unit (IMU) ensures interoceptive measurements without outliers that predict motion between images in a faster sampling time. Visual measurements reduce or bound an error accumulation in a noisy integration of IMU readings. There has been intensive research on visual-inertial navigation in the last decade [3]. Previous research has suggested fusion methods either by filtering or optimization-based estimator, a programming architecture composed of tracking frontend and mapping backend, and visual-inertial measurement processing techniques.

Depending on how an image measurement is formulated, one can minimize either geometric (*indirect*) or photometric (*direct*) error. The former has a rather long history, where the crucial step includes feature extraction, solving data association, and minimizing a reprojection error [4]. The latter *directly* minimizes a photometric error that measures an intensity discrepancy between consecutive images [5]. Apart from a subtle difference in a feature extraction strategy, the key difference lies in the dependence on a repeatable feature. While the geometric method has to detect visual features repeatedly across images to build the reprojection error, the photometric approach relies on an intensity gradient by which the discrepancy is minimized. There has been a lot of discussions on the literature to answer the question: *Which is better?* At least, it has been reported that the photometric method shows a robust short-term pose estimation performance over its alternatives in low-textured environments [6], [7].

However, a cost function formed by the photometric error is highly nonconvex in terms of pose and structure parameters [6]. The main reason for that is the nonlinearity in image intensities. Except for a gradual brightness change, intensities do not exhibit linearity. This leads to a huge sensitivity on an initial point to reach an optimal point. To circumvent this problem, previous work adopts the coarse-to-fine scheme to flatten local minima over a multiresolution in a practical point of view [8]–[10]. Others employ image patches that account for neighboring pixels [6], [11], [12], provide a better initial point based on an inertial sensor [13], [14], or train a deep neural network to generate a desirable feature map for the optimization problem [15], [16]. However, ensuring a highly accurate and robust solution for minimizing the photometric error parameterized by the pose and structure in real time is still a challenging problem.

To achieve high robustness against bad initialization, we focus on an intensity gradient given a projective uncertainty originated from geometric errors. Inspired by the stochastic linearization in random vibration [17], we derive an *optimal* image gradient in the sense that it minimizes the linearization error within the uncertainty. We realize the proposed gradient by sampling ensembles from the state uncertainty in a framework of photometric visual-inertial odometry (VIO). We claim four key contributions of this article as follows.

1) A framework of photometric VIO based on iterated extended Kalman filter (EKF) is introduced, where the state

Manuscript received 5 August 2021; accepted 20 December 2021. Date of publication 19 January 2022; date of current version 8 August 2022. This work was supported in part by Unmanned Vehicles Core Technology Research and Development Program through the Unmanned Vehicle Advanced Research Center and National Research Foundation of Korea (NRF) funded by the Korea government (MSIT) under Grant 2020M3C1C1A01086408, and in part by Hyundai NGV Company. This paper was recommended for publication by Associate Editor Margarita Chli and Editor Francois Chaumette upon evaluation of the reviewers' comments. *(Corresponding author: Chan Gook Park.)*

Jae Hyung Jung and Yeongkwon Choe are with the Department of Aerospace Engineering and Automation and Systems Research Institute, Seoul National University, Seoul 08826, South Korea (e-mail: lastflowers@snu.ac.kr; veritasbbo@snu.ac.kr).

Chan Gook Park is with the Navigation and Electronic System Laboratory, Department of Aerospace Engineering, and Institute of Advanced Aerospace Technology, Seoul National University, Seoul 08826, South Korea (e-mail: chanpark@snu.ac.kr).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TRO.2021.3139964.

Digital Object Identifier 10.1109/TRO.2021.3139964

space is modeled on matrix Lie groups. The photometric method makes our system robust to low-textured scenes, while most visual-inertial navigation systems adhere to repeatable and salient features.

2) We derive an optimal intensity gradient that accounts for its projective uncertainty in our proposed pipeline, and this leads to robustness to the bad initialization.
3) We present a thorough Monte Carlo simulation to demonstrate the effectiveness of the proposed image gradient.
4) We implement the proposed method in real time using C++ and analyze its estimation accuracy, consistency, and computation time in a real-world UAV flight, where most scenes are constituted by a featureless floor. We open our source code[1] for the benefit of the research community.

The rest of this article is organized as follows. In Section II, we review related work in the context of visual-inertial navigation and the photometric approach for localization. Definitions on coordinate frames and notations along with the extended pose group are given in Section III. We develop the photometric VIO starting from the state-space definition in Section IV. After laying the foundation, we derive the proposed intensity gradient in Section V. In Section VI, a Monte Carlo simulation and real-world flight test demonstrate our proposed framework. Finally, Section VII concludes this article.

II. Related Work

We review relevant research in the line of visual-inertial navigation and photometric approaches for pose and map estimation.

A. Visual-Inertial Navigation

One of the earliest seminal works in visual-inertial navigation includes the multistate constraint Kalman filter (MSCKF) [18] by Mourikis and Roumeliotis. The key idea was to marginalize feature positions in the state space by stochastically cloning the history of camera poses. This has been the backbone of follow-up studies. MSCKF 2.0 [19] remedied the filter inconsistency by using the first estimate Jacobian and introduced the term VIO, which implies the nature of estimation drift due to sequence-to-sequence motion estimation. Sun *et al.* [20] implemented a stereo measurement in a framework of MSCKF. More recently, a unified framework called OpenVINS [21] for monocular and stereo configuration was published.

On the other side, the Hessian matrix-based approach has been popular by virtue of its estimation accuracy and efficient implementation, exploiting the sparsity of the Hessian matrix. Leutenegger *et al.* [22] followed the principle of the keyframe [23] and introduced a marginalization procedure in VIO that preserves the sparsity pattern of the Hessian matrix. With the advent of the IMU preintegration [24], [25], visual-inertial navigation has become more mature. Qin *et al.* [26] proposed a visual-inertial navigation system that includes in-flight initialization, visual-inertial bundle adjustment (BA), and appearance-based loop detection with a pose-graph optimization. This was extended to [27] and [28] that includes a multisensor configuration and GNSS measurements. ORB-SLAM3 by Campos *et al.* [7] built on its predecessor [29], [30] features a tracking thread using ORB features, local BA thread, and a multimap data association to seamlessly fuse previously mapped areas.

Regardless of its implementation methodology, visual-inertial navigation systems are heading toward robustness to a system failure in a constrained computing platform. Eckenhoff *et al.* [31] developed a multi-IMU multicamera system that overcomes measurement depletion due to a limited field of view. The asynchronous multisensor measurements were interpolated to efficiently model the state space at a low computational budget. Huang *et al.* [32] extended an initialization procedure from a single camera-IMU pair to a stereo camera configuration. Carlone and Karaman [33] introduced a feature selection strategy by maximizing pose estimation accuracy at limited computational resources. Zhang *et al.* [34] devised the motion manifold that constraints a ground vehicle for efficient 6-D pose estimation.

In contrast to previous works, we focus on the photometric measurement that fuses visual and inertial measurements in a much deeper way than the geometric model in the sense that the fusion involves feature tracking and consequently spares explicit feature tracking in a sequence of temporal images. Therefore, our method does not suffer from outliers from feature mismatching and implicitly solves the feature correspondence by minimizing the photometric error.

B. Photometric Approaches

A photometric approach, also known as the direct method, minimizes intensity differences rather than a geometric error. It was successfully employed in 2-D sparse feature trackers [35], [36]. Extending an optimization parameter to a 6-DOF pose, real-time dense visual odometry (VO) was presented in [8] that maximizes photoconsistency. Kerl *et al.* [9] showed that the photometric residual is well expressed by the t-distribution and suggested a weight function that is robust to outliers. Relaxed from an assumption of dense depth measurements, J. Engel *et al.* [37] introduced semidense VO. The key idea was to track pixels with nonnegligible gradients by modeling photometric as well as geometric disparity uncertainties. This was extended to LSD-SLAM [38] and direct sparse odometry (DSO) [6]. In DSO, the key contribution was the real-time photometric BA on a CPU that exploits the sparsity structures of the corresponding Hessian matrix. This seminal work was extended to stereo DSO [39], DSO with loop closure [40], visual-inertial DSO [13], and direct sparse mapping [10]. More recently, a multidimensional feature map was trained for the direct image alignment in a long-baseline and multiweather condition [15], [16].

Hybrid approaches [12], [41], [42] use both photometric and geometric errors, while the photometric model provides accurate pose estimation over short-term tracking without data association, the geometric model gives robustness for a large baseline. A representative work by Forster *et al.* [12] proposed semidirect VO, where the short-term tracking is solved by minimizing the

[1] [Online]. Available: https://github.com/lastflowers/envio

photometric error, while windowed BA minimizes a reprojection error built from previously established matching pairs.

Visual-inertial navigation systems with the photometric measurement include [11], [13], [14], [43-46], where motion prediction from an IMU provides a good initialization for tracking convergence. Among these, the most relevant work to ours is robust VIO (ROVIO) by Bloesch *et al.* [11], in which pyramidal image patches are tracked in a framework of the iterated EKF. The key idea was to formulate the state space in a robocentric frame to reduce nonlinearity in a measurement model. They also introduced multiple hypotheses for pixel positions to avoid a tracking failure. On the other hand, our feature selection strategy adopts locally high gradient pixels that are uniformly distributed across an image instead of a small set of feature patches. Aside from the difference in the feature extraction and the filter formulation, we suggest an image gradient that is *optimal* in the sense of a linearization error within a projective uncertainty.

III. Preliminaries

A. Coordinate Frame Definition and Notations

Throughout this article, the global frame $\{g\}$ is defined as a local tangent plane frame fixed at the starting point of the body frame $\{b\}$ of a robot. Its heading is aligned to that of $\{b\}$ at the beginning. The IMU frame is coincident with $\{b\}$, and the left camera frame is denoted as $\{c\}$ located on the optical center of a camera model pointing right, down, and forward direction. The right camera frame $\{r\}$ is defined analogously. If we need to specify a time instance, we adopt a subscript to a coordinate frame, for example, $\{b_k\}$ means $\{b\}$ at time t_k. We assume that spatial and temporal extrinsic parameters are calibrated for $\{c\}$, $\{r\}$, and $\{b\}$.

We express a vector (or a scalar) and a matrix as small and capital letters such as x and X. When we place a coordinate frame to the upper-right side of a vector or matrix, it indicates reference and resolved frames. A subscript means a target frame. For instance, p_b^g is a position of $\{b\}$ referenced at $\{g\}$. Identity and zero matrices are expressed as Id and 0. Their dimensions should be clear in the context.

B. Extended Pose Group

We model the state space on matrix Lie groups and derive their corresponding errors on the vector elements of the Lie algebra. As introduced in invariant EKF [43], the so-called extended pose is defined as

$$SE_2(3) = \left\{ \begin{bmatrix} R & p & v \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \middle| \; R \in SO(3),\ p, v \in \mathbb{R}^3 \right\} \tag{1}$$

where R, p, and v represent attitude, position, and velocity of a robot with respect to a reference frame. Note that, we express robot's attitude in the special orthogonal group in three dimensions. Its associated Lie algebra is

$$\mathfrak{se}_2(3) = \left\{ \begin{bmatrix} \theta^\wedge & \alpha & \beta \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \middle| \; \theta,\ \alpha,\ \beta \in \mathbb{R}^3 \right\} \tag{2}$$

where θ, α, and β are associated with attitude, position, and velocity of a robot in (6).

An element in the Lie algebra is associated with a vector by the hat $(\cdot)^\wedge$ and vee $(\cdot)^\vee$ operator

$$x = \begin{bmatrix} \theta^T & \alpha^T & \beta^T \end{bmatrix}^T \in \mathbb{R}^9 \tag{3}$$

$$x^\wedge = \begin{bmatrix} \theta^\wedge & \alpha & \beta \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \text{and} \quad x = \left(x^\wedge\right)^\vee. \tag{4}$$

In the case of $\mathfrak{so}(3)$, $(\cdot)^\wedge$ corresponds to the skew-symmetric matrix operator.

Elements of $X \in SE_2(3)$ and $x \in \mathfrak{se}_2(3)$ are exactly converted to each other by the matrix exponential and logarithm mapping

$$\begin{aligned} X &= \exp(x^\wedge) \quad \text{and} \\ x &= \log(X)^\vee. \end{aligned} \tag{5}$$

The closed-form formula of $\exp(\cdot)$ for $SE_2(3)$ is derived as similar to $SE(3)$ [44]

$$\exp\left(x^\wedge\right) = \begin{bmatrix} \exp\left(\theta^\wedge\right) & J_l(\theta)\alpha & J_l(\theta)\beta \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

where the left Jacobian J_l is

$$J_l(\theta) = Id + \frac{1 - \cos\|\theta\|}{\|\theta\|^2}\theta^\wedge + \frac{\|\theta\| - \sin\|\theta\|}{\|\theta\|^3}\left(\theta^\wedge\right)^2. \tag{7}$$

The exponential mapping for the rotation vector has the closed-form formula as (Rodrigues' formula)

$$\exp\left(\theta^\wedge\right) = Id + \frac{\sin\|\theta\|}{\|\theta\|}\theta^\wedge + \frac{1 - \cos\|\theta\|}{\|\theta\|^2}(\theta^\wedge)^2. \tag{8}$$

By inverting (6) in $\|\theta\| < \pi,\ \|\theta\| \neq 0$, the inverse map $\log(\cdot)$ is

$$\begin{aligned} \|\theta\| &= \cos^{-1}\left(\frac{\mathrm{tr}(R) - 1}{2}\right) \\ \theta &= \frac{\|\theta\|}{2\sin\|\theta\|}\left(R - R^T\right)^\vee \\ \alpha &= J_l^{-1}(\theta)\, p \\ \beta &= J_l^{-1}(\theta)\, v. \end{aligned} \tag{9}$$

Note that, $\exp(\cdot)$ and $\log(\cdot)$ are locally bijective mappings due to the ambiguity in every $\|\theta\| = 2\pi n$ with n a nonzero integer. We obtain $SE(3)$ when eliminating the velocity entries of $SE_2(3)$.

C. Right-Invariant Error

The right-invariant error δX [43] is defined as

$$\delta X = \hat{X} X^{-1} \tag{10}$$

where $X \in SE_2(3)$ and the overhead hat $\hat{(\cdot)}$ represents an estimate for the corresponding quantity. This is a generalization of the vector subtraction in the vector space. This error matrix δX is associated with the tangent space element at the identity as

$$\begin{aligned}\xi &= \log\left(\hat{X} X^{-1}\right)^{\vee} \\ &= \begin{bmatrix}\phi^T & \rho^T & \nu^T\end{bmatrix}^T \end{aligned} \tag{11}$$

where $\xi^{\wedge} \in \mathfrak{se}_2(3)$ and $\exp(\cdot)$, $\log(\cdot)$ are defined in Section III-B.

IV. Visual-Inertial State Estimation

A. Problem Definition

Given three-axis angular rates $\omega_m(t_{0:k})$, specific force measurements $a_m(t_{0:k})$, and image intensities $I_{0:k}$ from time t_0 to t_k, our objective is to estimate the current pose of a robot $T_b^g(t_k) \in SE(3)$ and its surrounding feature map p_f^b with their estimate confidences.

Inspired by the direct sparse odometry [6], we define the state space as the current extended pose $X_b^g(t)$, IMU biases $B(t)$, the previous pose when an image is captured $T_{b_l}^g(t)$, and depths function at the previous camera pose $D(t)$, that is

$$\mathcal{X}(t) = \begin{bmatrix} X_b^g(t) & 0 & 0 & 0 \\ 0 & B(t) & 0 & 0 \\ 0 & 0 & T_{b_l}^g(t) & 0 \\ 0 & 0 & 0 & D(t) \end{bmatrix} \tag{12}$$

where m is the number of features being tracked in the filter state. The current and previous poses are

$$X_b^g(t) = \begin{bmatrix} R_b^g(t) & p_b^g(t) & v_b^g(t) \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \in SE_2(3) \tag{13}$$

$$T_{b_l}^g(t) = \begin{bmatrix} R_{b_l}^g(t) & p_{b_l}^g(t) \\ 0 & 1 \end{bmatrix} \in SE(3). \tag{14}$$

The bias and depth function matrices are

$$B(t) = \begin{bmatrix} Id & b_a(t) & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & Id & b_g(t) \\ 0 & 0 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{8\times 8} \tag{15}$$

$$D(t) = \begin{bmatrix} 1 & d_1(t) & & 0 & 0 \\ 0 & 1 & & 0 & 0 \\ & & \ddots & & \\ 0 & 0 & & 1 & d_m(t) \\ 0 & 0 & & 0 & 1 \end{bmatrix} \in \mathbb{R}^{2m\times 2m} \tag{16}$$

where b_a, b_g are accelerometer and gyroscope biases, and d_j is the jth depth parameterization referenced at t_l that would be an inverse depth $d_j = z_j^{-1}$ or a depth $d_j = z_j$. We will discuss the depth parameterization in Section IV-C.

We simplify the matrix expression and omit the coordinate frame and time argument to ease the readability if the context is clear such that

$$\mathcal{X} = (X, b_a, b_g, T, d_1, \ldots, d_m). \tag{17}$$

B. Process Model

IMU measurements are modeled as the true quantity corrupted by the time-varying bias and zero-mean white Gaussian processes

$$\begin{aligned} a_m(t) &= a_t(t) + b_a(t) + n_a(t) \\ \omega_m(t) &= \omega_t(t) + b_g(t) + n_g(t) \end{aligned} \tag{18}$$

where noises are $n_a(t) \sim GP(0,\, Q_a\delta(t-\tau))$ and $n_g(t) \sim GP(0,\, Q_g\delta(t-\tau))$. $GP(m, P)$ stands for the multivariate Gaussian process whose mean and covariance are m and P, and Q_a, Q_g are power spectral density matrices.

The extended pose and biases are governed by the following differential equations:

$$\begin{aligned} \dot{R}(t) &= R(t)\left(\omega_m(t) - b_g(t) - n_g(t)\right)^{\wedge} \\ \dot{p}(t) &= v(t) \\ \dot{v}(t) &= R(t)\left(a_m(t) - b_a(t) - n_a(t)\right) + g \\ \dot{b}_a(t) &= n_{wa}(t) \\ \dot{b}_g(t) &= n_{wg}(t) \end{aligned} \tag{19}$$

where g is the gravity in $\{g\}$ and biases are modeled as random walks with their densities $n_{wa}(t) \sim GP(0,\, Q_{wa}\delta(t-\tau))$ and $n_{wg}(t) \sim GP(0,\, Q_{wg}\delta(t-\tau))$. The previous pose T and jth depth functions d_j are modeled as random constants.

The right-invariant error for the state $\mathcal{X}(t)$ is

$$\delta\mathcal{X}(t) = \exp\left(\zeta(t)^{\wedge}\right) = \hat{\mathcal{X}}(t)\mathcal{X}(t)^{-1}. \tag{20}$$

The vector element at the corresponding tangent space is

$$\zeta = \begin{bmatrix}\phi^T & \rho^T & \nu^T & \delta b_a^T & \delta b_g^T & \phi_l^T & \rho_l^T & \delta d_1 & \cdots & \delta d_m\end{bmatrix}^T \tag{21}$$

where ϕ, ρ, and ν are defined in (11) and ϕ_l, ρ_l are a pose error at the previous time t_l. Except for the current extended pose and the previous pose, the rest of errors are defined by the vector subtraction as defined in (20).

The error-state ζ up to the second-order term is evolved by

$$\dot{\zeta}(t) \approx F(t)\,\zeta(t) + G(t)\,w(t) \tag{22}$$

where F, G are Jacobian matrices to ζ and the noise vector $w = \begin{bmatrix} n_a^T & n_g^T & n_{wa}^T & n_{wg}^T \end{bmatrix}^T$. It is worthwhile to note that, the linearized (22) is perfect when $\delta b_a = \delta b_g = 0$ and $w = 0$ [43]. State uncertainties are well captured by the invariant error (11) as in $SE(3)$ [45], [46]. As previously derived in [47], the Jacobian matrix is turned to be

$$F(t) = \begin{bmatrix} F_I(t) & 0 \\ 0 & 0 \end{bmatrix}$$

$$F_I(t) = \begin{bmatrix} 0 & 0 & 0 & 0 & -\hat{R}(t) \\ 0 & 0 & Id & 0 & -\hat{p}(t)^\wedge \hat{R}(t) \\ g^\wedge & 0 & 0 & -\hat{R}(t) & -\hat{v}(t)^\wedge \hat{R}(t) \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{23}$$

In our implementation, (19) and (22) are discretized to propagate the mean $\hat{\mathcal{X}}$ and the covariance matrix $P = E[\zeta\, \zeta^T]$.

C. Photoconsistency Model

The photoconsistency assumption states that intensities are the same regardless of the viewpoint of a camera if a ray hits the Lambertian surface. This has been successfully employed in the direct visual odometry [8] and with illumination parameter estimation [6] to track the 6-DOF pose of a camera. We adopt this as a filter measurement to spare the explicit 2-D feature tracking.

For the jth feature at t_k, this is written as

$$\begin{aligned} y_j(\mathcal{X}) &= h\left(\varphi(\mathcal{X},\, u_j^l)\right) + n_j \\ &= I_l\left(u_j^l\right) - I_k\left(\varphi(\mathcal{X},\, u_j^l)\right) + n_j \end{aligned} \tag{24}$$

where $u_j^l \in \mathbb{R}^2$ is the jth pixel coordinate at the reference t_l. n^j is the zero-mean white Gaussian noise $n_j \sim N(0, \sigma_j^2)$ independent to the process noise w. I_l and I_k are images at t_l and t_k, respectively. Note that, $y_j = 0$ without the noise. The warping function φ is

$$\varphi\left(\mathcal{X},\, u_j^l\right) = \Pi\left(\left(T_{b_k}^g T_c^b\right)^{-1} T_{b_l}^g T_c^b \begin{bmatrix} p_j^{c_l} \\ 1 \end{bmatrix}\right) \tag{25}$$

where Π is a perspective projection model. The jth feature position viewed at the previous camera frame $\{c_l\}$ is

$$p_j^{c_l} = \Pi^{-1}\left(u_j^l,\, d_j\right). \tag{26}$$

The nonlinear function h is linearized to incrementally minimize the photometric error

$$\begin{aligned} \delta y_j &= y_j - \hat{y}_j \\ &\approx H_j \zeta + n_j. \end{aligned} \tag{27}$$

The Jacobian matrix is derived using the chain rule

$$\begin{aligned} H_j &= -\frac{\partial I_k}{\partial \zeta} \\ &= -\frac{\partial I_k}{\partial u_j^k} \frac{\partial u_j^k}{\partial p_j^{c_k}} \frac{\partial p_j^{c_k}}{\partial \zeta} \end{aligned} \tag{28}$$

where u_j^k is the jth pixel coordinate at I_k and $p_j^{c_k}$ is the 3-D jth feature position referenced at the current camera frame $\{c_k\}$.

The first block is an image gradient at the predicted pixel coordinate

$$\frac{\partial I_k}{\partial u_j^k} = \nabla I_k(\hat{u}_j^k) \tag{29}$$

from which most of linearization error is originated. We will propose an image gradient that minimizes a linearization error in Section V. The second block is 2-D-to-3-D feature point Jacobian

$$\frac{\partial u_j^k}{\partial p_j^{c_k}} = \begin{bmatrix} f_u(\hat{p}_{j,z}^{c_k})^{-1} & 0 & -f_u\, \hat{p}_{j,x}^{c_k}(\hat{p}_{j,z}^{c_k})^{-2} \\ 0 & f_v(\hat{p}_{j,z}^{c_k})^{-1} & -f_v\, \hat{p}_{j,y}^{c_k}(\hat{p}_{j,z}^{c_k})^{-2} \end{bmatrix} \tag{30}$$

where the pin-hole projection model is used with horizontal and vertical focal lengths f_u and f_v. $\hat{p}_{j,x}^{c_k}$ indicates the first element of $\hat{p}_j^{c_k}$ and so on. The last block is filled by the pose and corresponding depth blocks

$$\begin{aligned} &\frac{\partial p_j^{c_k}}{\partial \zeta} = \hat{R}_k^T \\ &\begin{bmatrix} -\left(\hat{p}_j^g\right)^\wedge & Id & \cdots & \left(\hat{p}_j^g\right)^\wedge & -Id & \cdots & \hat{R}_l \hat{p}_j^{c_l} \hat{d}_j^{-1} & \cdots \end{bmatrix} \end{aligned} \tag{31}$$

where $\hat{R}_k = \hat{R}_{c_k}^g$ and $\hat{p}_j^g$ is the jth 3-D feature position referenced at $\{g\}$.

The inverse depth parameterization [48] has been broadly used because it yields the high linearity index in a pixel projection function, and exhibits a long tail in a far region. However, in our proposed filter we use a photometric measurement, where the majority of nonlinearity comes from an image intensity. We initialize a feature depth by a stereo baseline with enough parallax. This is why we choose the depth parameterization in the current implementation. However, our approach can include far features using inverse depth parameterization as suggested in [49] without any difficulties.

D. Iterated EKF on Matrix Lie Groups

The iterated EKF is a local maximum *a posteriori* estimator in a single step [44] that iteratively minimizes a weighted sum of costs until convergence. In the ROVIO [11], the authors presented iterated EKF formulations that account for rotations and bearing vectors that live in a manifold. In this article, however, we derive the filter update step in matrix Lie groups that includes $SE_2(3)$ that is a proper group representation for an inertial navigation system.

The objective is to maximize

$$\begin{aligned} \hat{\mathcal{X}}_k &= \underset{\mathcal{X}_k}{\operatorname{argmax}}\, p\left(\mathcal{X}_k \middle|\, \mathbf{y}_{0:k},\, a_m(t_{0:k}),\, \omega_m(t_{0:k})\right) \\ &= \underset{\mathcal{X}_k}{\operatorname{argmax}}\, p\left(\mathbf{y}_k \middle|\, \mathcal{X}_k\right) p\left(\mathcal{X}_k \middle|\, \mathbf{y}_{0:k-1},\, a_m(t_{0:k}),\, \omega_m(t_{0:k})\right) \end{aligned} \tag{32}$$

where a density function of the matrix Lie group is indirectly defined by its corresponding Lie algebra [45] and $\mathcal{X}_k = \mathcal{X}(t_k)$,

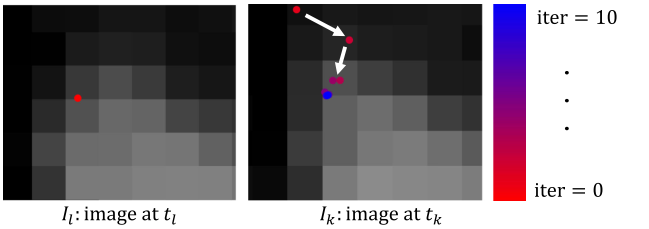


Fig. 1. Converged example in the VIODE dataset [50]: After a couple of update iterations the pixel point reaches the photometrically as well as the geometrically consistent region.

$\mathbf{y}_{0:k} = \mathbf{y}(t_{0:k})$. Here, $\mathbf{y}_k$ is a vector that collects all measurements at t_k. This is equivalent to

$$\begin{aligned}\hat{\mathcal{X}}_k &= \underset{\mathcal{X}_k}{\operatorname{argmin}} \|\mathbf{y}_k - \mathbf{h}(\mathcal{X}_k)\|^2_{R_k^{-1}} + \left\| \log\left(\hat{\mathcal{X}}_k^- \mathcal{X}_k^{-1}\right)^\vee \right\|^2_{(P_k^-)^{-1}} \\ &\approx \underset{\zeta_{k,i}}{\operatorname{argmin}} \left\| \mathbf{y}_k - \mathbf{h}(\mathcal{X}^+_{k,i-1}) - \mathbf{H}_{i-1}\zeta_{k,i} \right\|^2_{R_k^{-1}} \\ &\quad + \left\| \log\left(\hat{\mathcal{X}}_k^-(\mathcal{X}^+_{k,i-1})^{-1}\right)^\vee + \zeta_{k,i} \right\|^2_{(P_k^-)^{-1}} \qquad (33)\end{aligned}$$

where

$$\mathbf{h}(\mathcal{X}_k) = \begin{bmatrix} h\left(\varphi(\mathcal{X}_k, u_1^l)\right) & \cdots & h\left(\varphi(\mathcal{X}_k, u_m^l)\right)\end{bmatrix}^T \qquad (34)$$

$$R_k = \begin{bmatrix} \sigma_1^2 & & \\ & \ddots & \\ & & \sigma_m^2 \end{bmatrix}. \qquad (35)$$

In this expression, P_k^- is the covariance matrix before the filter update $P_k^- = E[\zeta_k^- \, (\zeta_k^-)^T]$. *A priori* covariance is propagated according to (22). $\hat{\mathcal{X}}_k^-$ is *a priori* of $\mathcal{X}_k$. In the second line in (33), we have substituted the current ith *a posteriori* from the $(i-1)$th iteration: $\mathcal{X}_k = \exp(-\zeta^\wedge_{k,i})\hat{\mathcal{X}}^+_{k,i-1}$ up to the higher order terms. $\mathbf{H}_{i-1}$ is stacked from (28) and linearized at $\hat{\mathcal{X}}^+_{k,i-1}$.

By differentiating the cost in (33) with respect to $\zeta_{k,i}$, the update step is given as

$$\begin{aligned}\zeta_{k,i} &= K_{i-1}\left(\mathbf{y}_k - \mathbf{h}(\hat{\mathcal{X}}^+_{k,i-1})\right) \\ &\quad - (Id - K_{i-1}\mathbf{H}_{i-1}) \log\left(\hat{\mathcal{X}}_k^-(\hat{\mathcal{X}}^+_{k,i-1})^{-1}\right)^\vee \qquad (36)\end{aligned}$$

where $K_{i-1} = (\mathbf{H}_{i-1}^T R_k^{-1} \mathbf{H}_{i-1} + (P_k^-)^{-1})^{-1} \mathbf{H}_{i-1}^T R_k^{-1}$ is the Kalman gain linearized at $(i-1)$th estimation. We define

$$\bar{\zeta}_{k,i-1} = \log\left(\hat{\mathcal{X}}_k^-(\hat{\mathcal{X}}^+_{k,i-1})^{-1}\right)^\vee, \quad \bar{\zeta}_{k,0} = 0 \qquad (37)$$

and incrementally update *a posteriori*

$$\hat{\mathcal{X}}^+_{k,i} = \exp\left(-\bar{\zeta}^\wedge_{k,i}\right) \hat{\mathcal{X}}_k^- \qquad (38)$$

where

$$\bar{\zeta}_{k,i} \approx K_{i-1}\left((\mathbf{y}_k - \mathbf{h}(\hat{\mathcal{X}}^+_{k,i-1}) + \mathbf{H}_{i-1}\bar{\zeta}_{k,i-1}\right). \qquad (39)$$

If $\bar{\zeta}_{k,i}$ is converged we update the covariance matrix as

$$P_k^+ = (Id - K_i \mathbf{H}_i) \, P_k^-. \qquad (40)$$

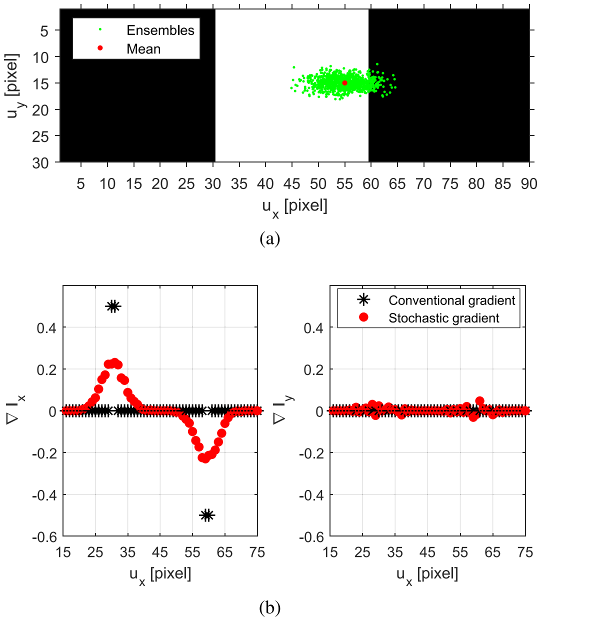


Fig. 2. Motivating example in a toy problem. (a) The point on the black and white image moves from $u_x = 15$ to $u_x = 75$ with its ensembles (small green dots) sampled from a Gaussian distribution. (b) The conventional image gradient (at the mean) and the proposed stochastic gradient (46) when traveling to the x-direction.

This is a generalization of the iterated EKF on the vector space: If we replace $\log(\hat{\mathcal{X}}\mathcal{X}^{-1})^\vee$ by the vector subtraction we arrived at the equivalent formulation.

Fig. 1 shows that the iteration step (38) and (39) is converged to the photometrically as well as geometrically consistent area by minimizing visual-inertial costs (33) in a sequence of temporal images.

E. Feature Initialization, Tracking, and Marginalization

We process input stereo images as a set of feature points that includes a pixel coordinate and its initial depth estimate on the left camera frame. First, we undistort incoming stereo images and convert the left grayscale image into a gradient magnitude map. Then, we divide the gradient map into 25×15 grids and select the locally strongest pixel greater than a minimum threshold. To maintain uniformly distributed points over an image, we manage an image mask to ensure a minimum distance among features. As noted in the DSO [6], this strategy does not depend on corner features and performs well in low-textured environments.

The depth is initialized by epipolar line search evaluated by the sum of squared differences (SSD) within a 13×13 patch in the stereo baseline. We reject badly triangulated features based on a ratio of the minimum and the second minimum SSD, and an inner product of image gradient direction and a unit epipolar line. After passing the quality check, the feature depth is augmented in the filter state with a sufficiently large initial uncertainty $\sigma_z = 1.5$m.

Features in the state space are tracked by minimizing the visual and inertial costs (33). After convergence of an update step, features at t_l are warped to t_k using *a posteriori*. In this step, we evaluate normalized cross correlations (NCC) in 13×13 patches centered at $\hat{u}_j^l$ and $\hat{u}_j^k$, and marginalize features if the NCC is smaller than a certain threshold. After the feature tracking and marginalization, we replace the previous pose at t_l with the current pose at t_k as noted in line 12 of Algorithm 1. In a covariance domain, marginalization erases the corresponding depth blocks in the covariance matrix.

Algorithm 1: Ensemble Visual-Inertial Odometry.

```
    Input: $\hat{\mathcal{X}}_l^+,\ P_l^+,\ a_m(t_{l:k}),\ \omega_m(t_{l:k}),\ I_l,\ I_k,\ \{u_j^l\}_{j=1:m}$
    Output: $\hat{\mathcal{X}}_k^+,\ P_k^+,\ \{u_j^k\}_{j=1:m}$ $(\hat{\mathcal{X}}_0^+,\ P_0^+) \leftarrow$
      Initialization($a_m(t_{0:n_i}),\ \omega_m(t_{0:n_i})$)
 1: $(\hat{\mathcal{X}}_k^-,\ P_k^-) \leftarrow$ Time-
    propagation($\hat{\mathcal{X}}_l^+,\ P_l^+,\ a_m(t_{l:k}),\ \omega_m(t_{l:k})$)
 2: for $i = 1$ to $n$ do
 3:   for $j = 1$ to $m$ do
 4:     $\mathcal{H}_j \leftarrow$
        StochasticGradient($\hat{\mathcal{X}}_{k,i-1}^+,\ P_k^-,\ I_k,\ u_j^l$)
 5:     $H_j \leftarrow$
        MeasurementJacobian($\mathcal{H}_j,\ \hat{\mathcal{X}}_{k,i-1}^+,\ u_j^l$)
 6:     $\delta y_j \leftarrow$
        FilterInnovation($\hat{\mathcal{X}}_{k,i-1}^+,\ I_l,\ I_k,\ u_j^l$)
 7:   end for
 8:   $\hat{\mathcal{X}}_{k,i}^+ \leftarrow$ Update($\hat{\mathcal{X}}_{k,i-1}^+,\ P_k^-,\ \mathbf{H}_{k,i},\ \delta\mathbf{y}_{k,i}$)
 9: end for
10: $P_k^+ \leftarrow$ CovarianceUpdate($P_k^-,\ \mathbf{H}_{k,n}$)
11: Feature tracking: $\{u_j^k\}_{j=1:m} \leftarrow \varphi(\hat{\mathcal{X}}_k^+, \{u_j^l\}_{j=1:m})$
12: Replace the previous pose to the current one: $T_l \leftarrow T_k$
13: if ($m < n_{\min}$) then
14:   Initialize new features.
15: end if
```

Due to the nature of the tracking mechanism, the measurement noise n_j in (24) is colored noise. This can be handled by Kalman filter with a colored noise [51]. From a practical point of view, we inflate the measurement noise σ_j to tackle this unmodeled error.

V. Stochastic Gradient

A. *Motivating Example*

We interpret tracked points in an image as an estimate revealed from its projective uncertainty due to camera pose and depth uncertainties. A simple black and white image in Fig. 2(a) shows a red pixel that travels from $u_x = 15$ to $u_x = 75$, plotting its image gradient in the horizontal and vertical directions in Fig. 2(b). We assume that the red pixel is the mean of a 2-D Gaussian distribution, where ensembles are sampled from the distribution.

In the vicinity of the edges, image gradients are zero: There is no information to minimize the photometric error. However, our approach gives nonnegligible image gradients derived from the pixel uncertainty as in Fig. 2(b). That is, it is reasonable to account for the probabilistic property when computing an image gradient. We introduce a *stochastic gradient* that reflects the projective uncertainty inspired by stochastic linearization [17].

Previous approaches handle the intensity nonlinearity, including this extreme case, by using an iteration over an image pyramid to flatten local minima (coarse-to-fine scheme) [8]–[10] and image patches to include neighboring pixels [6], [11]. However, our approach guarantees an *optimal gradient* in the sense of a linearization error that helps to converge to the correct direction.

B. *Derivation of Stochastic Gradient*

In deriving the stochastic gradient, we focus on the image gradient which is the first matrix block in (28). We repeat the associated jth feature intensity in time t_k

$$\mathcal{Y}(u_j^k) = I_k\left(\varphi(\mathcal{X}, u_j^l)\right) + n_{kj} \tag{41}$$

where $u_j^k = \varphi(\mathcal{X}, u_j^l)$ and n_{kj} is a zero-mean white Gaussian noise that contributes to the noise n_j in (24). A naive approach is to linearize (41) starting from the filter state $\mathcal{X}$. However, we found that the nonlinearity in an image intensity is higher than that of the perspective projection. Furthermore, the naive approach will turn to require 13×13 dense matrix inversion per a feature. This is why we decide to linearize (41) at the pixel position u_j^k that requires only 2×2 matrix inversion per a feature.

We define a loss function

$$L(\mathcal{H}) = I(u) - (I(\hat{u}) + \mathcal{H}\,\delta u) \tag{42}$$

where $I(u) = I_k(u_j^k)$, $u = \hat{u} + \delta u$ and $\mathcal{H}$ is an image gradient we wish to find. Then, we minimize the expectation of the squared loss function

$$\hat{\mathcal{H}} = \underset{\mathcal{H}}{\operatorname{argmin}}\ E\left[L^2(\mathcal{H})\right]. \tag{43}$$

This can be rewritten as

$$\begin{aligned}\hat{\mathcal{H}} &= \underset{\mathcal{H}}{\operatorname{argmin}}\ E\left[\left(\mathcal{Y}(u) - n - I(\hat{u}) - \mathcal{H}\,\delta u\right)^2\right]\\ &= \underset{\mathcal{H}}{\operatorname{argmin}} \int_{\delta u}\int_{n} \left(\mathcal{Y}(u) - n - I(\hat{u}) - \mathcal{H}\,\delta u\right)^2 p(\delta u, n)\, dn\, d\delta u.\end{aligned} \tag{44}$$

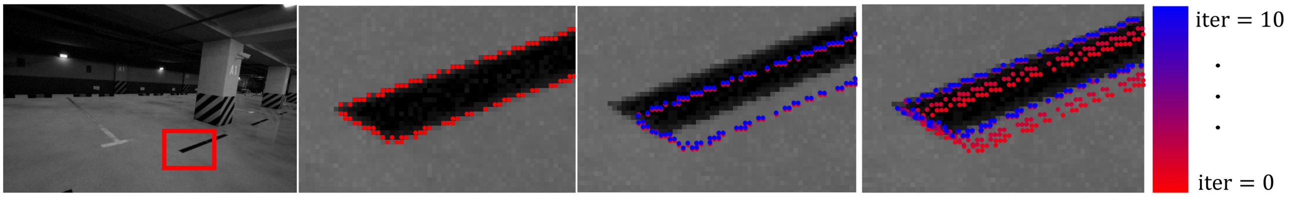
Since we assume that the measurement and process noises are independent, the joint density function is decomposed as $p(\delta u, n) = p(\delta u)\, p(n)$.

Differentiating with respect to the gradient yields

$$\begin{aligned}\frac{dE\left[L^2(\mathcal{H})\right]}{d\mathcal{H}} &= -2\int_{\delta u} \mathcal{Y}(u)\,\delta u^T\, p(\delta u)\, d\delta u\\ &+ 2\, I(\hat{u}) \int_{\delta u} \delta u^T p(\delta u)\, d\delta u + 2\,\mathcal{H}\int_{\delta u} \delta u\, \delta u^T p(\delta u)\, d\delta u\end{aligned} \tag{45}$$

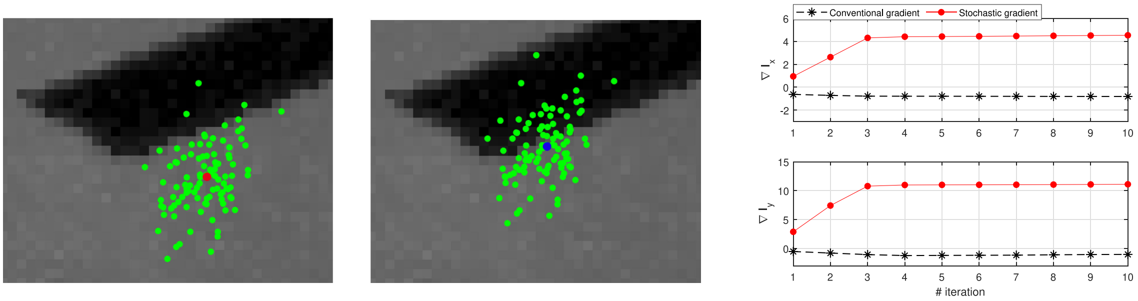
where the zero-mean measurement noise assumption is employed. Equating (45) as zero gives

$$\begin{aligned}\hat{\mathcal{H}} &= \left(\int_{\delta u} \mathcal{Y}(u)\delta u^T p(\delta u)\, d\delta u - I(\hat{u})\int_{\delta u} \delta u^T p(\delta u)\, d\delta u\right) \times\\ &\quad \left(\int_{\delta u} \delta u\, \delta u^T p(\delta u)\, d\delta u\right)^{-1}\\ &= \left(E\left[\mathcal{Y}(u)\,\delta u^T\right] - I(\hat{u})E\left[\delta u^T\right]\right)\left(E\left[\delta u\,\delta u^T\right]\right)^{-1}.\end{aligned} \tag{46}$$



(a) (b) (c) (d)

Fig. 3. Illustrative example in the VIODE parking lot dataset [50]. (a) A reference image at t_l. (b) A close-up of the lane at t_l with high gradient features. (c) Pose tracking result at the current time t_k using the conventional gradient. (d) The proposed stochastic gradient, where the red-to-blue color encodes iteration steps in the iterated EKF.



(a) (b) (c)

Fig. 4. Representative pixel coordinate among the extracted features in Fig. 3(d) with sampled ensembles ($n_{en} = 100$) at (a) the 1st iteration and (b) the 10th iteration. (c) Its intensity gradients during the update steps, where the black and red plot correspond to intensity gradients of the representative pixel in Fig. 3(c) and (d), respectively.

TABLE I
TRAJECTORY INFORMATION REPRODUCED FROM [50]

Parameters	parking_lot	city_day	city_night
Distance [m]	75.8	157.7	165.7
Duration [s]	59.6	66.4	61.6

TABLE II
IMU SPECIFICATION IN THE MONTE CARLO SIMULATION

Specifications	Gyroscope	Accelerometer
Sampling rate	200 Hz	200 Hz
Noise density	0.0135 deg/s/$\sqrt{\text{Hz}}$	0.23 mg/$\sqrt{\text{Hz}}$
Bias repeatability	0.5 deg/s	20 mg
Bias stability	14.5 deg/hr	0.25 mg

It is interesting to note that, (46) boils down to a numerical differentiation in a noise-free model

$$\hat{\mathcal{H}} = \frac{I(\hat{u} + \delta u) - I(\hat{u})}{\delta u} \tag{47}$$

where we set $\delta u \in \mathbb{R}$. Note that, (46) is an *optimal* gradient that minimizes the mean square of the linearization error. We replace the conventional image gradient (29) to (46).

C. Stochastic Gradient Implementation

It is not straightforward to compute the correlation between intensities and pixel position deviation $E[\mathcal{Y}(u)\,\delta u^T]$ analytically. Therefore, we compute the correlation by sampling ensembles according to the current state uncertainty. The ith ensemble is sampled through

$$\mathcal{X}^{(i)} = \exp\left(-\zeta^{(i)\wedge}\right)\hat{\mathcal{X}} \tag{48}$$

where $\zeta^{(i)}$ is sampled from the IMU-predicted covariance. Each feature point is projected to the current image plane at t_k. The ith ensemble of pixel coordinate at t_k is

$$u_j^{k,(i)} = \varphi\left(\mathcal{X}^{(i)}, u_j^l\right). \tag{49}$$

Therefore, we can compute statistical properties of u. The expectation of its deviation from the estimate is

$$E\left[\delta u^T\right] = \frac{1}{n_{en}} \sum_i \left(u_j^{k,(i)} - \hat{u}\right)^T. \tag{50}$$

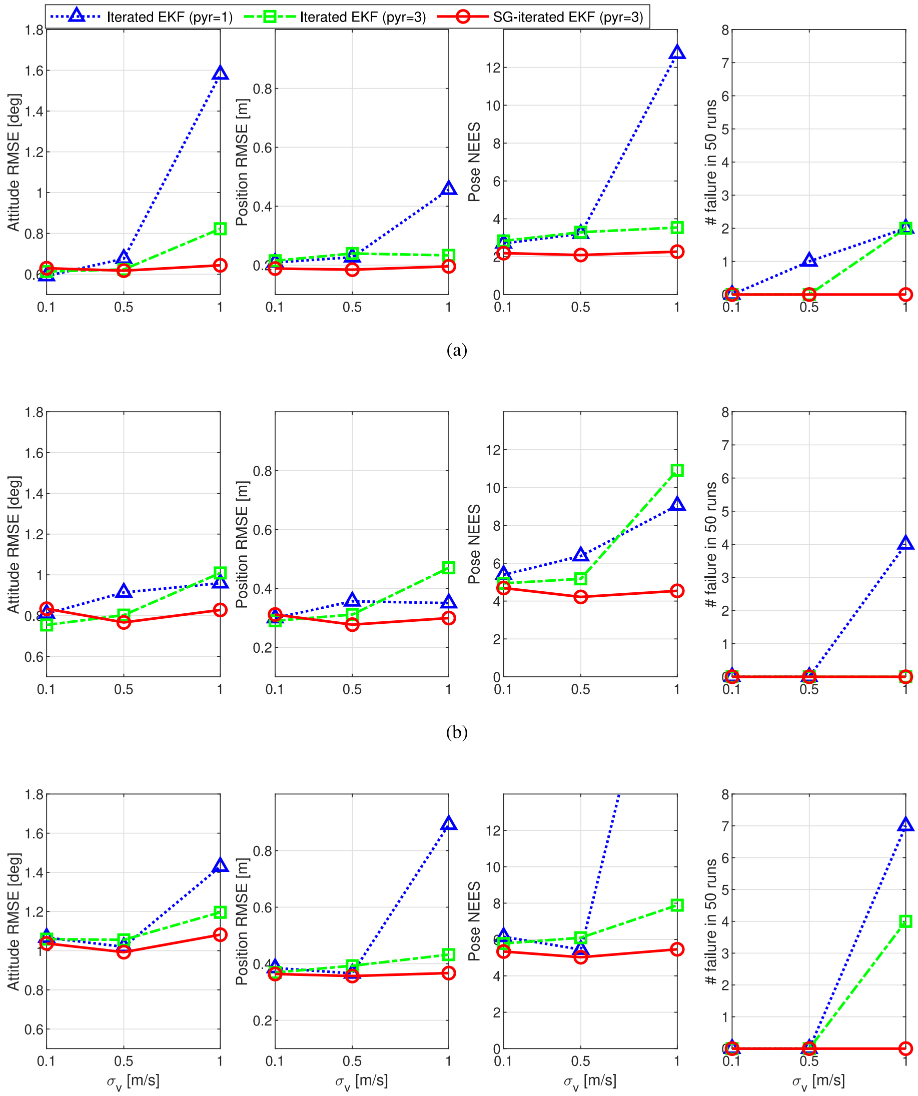
where n_{en} is a number of ensembles and the estimate is calculated as $\hat{u} = \varphi(\hat{\mathcal{X}}, u_j^l)$. The covariance of the projected pixel coordinate is

$$E\left[\delta u\,\delta u^T\right] = \frac{1}{n_{en} - 1} \sum_i \left(u_j^{k,(i)} - \hat{u}\right)\left(u_j^{k,(i)} - \hat{u}\right)^T \tag{51}$$

and the cross-correlation between the image intensity and the position deviation is

$$E\left[\mathcal{Y}(u)\,\delta u^T\right] = \frac{1}{n_{en} - 1} \sum_i \mathcal{Y}(u_j^{k,(i)})\left(u_j^{k,(i)} - \hat{u}\right)^T. \tag{52}$$

In the process of the filter update, each ensemble contributes to the stochastic gradient. Thus, we term our method as *ensemble VIO* (EnVIO). Algorithm 1 summarizes the overall procedure of EnVIO.



(a)

(b)

(c)

Fig. 5. Attitude and position RMSE, pose NEES, and the number of failures (position RMSE is larger than 5% of flight distance, or attitude RMSE is larger than 10 deg) of 50 Monte Carlo runs with the increasing initial velocity uncertainty $\sigma_v = \{0.1,\ 0.5,\ 1.0\}$ m/s in (a) parking_lot, (b) city_day, and (c) city_night.

Fig. 3 shows a pose tracking result in the parking lot sequence of the VIODE dataset [50] with a 1-m/s initial velocity error. Locally high gradient features on the lane mark are extracted in the image I_l, as shown in Fig. 3(b). Features are tracked by minimizing (33) using the conventional image gradient and the proposed stochastic gradient. Features are trapped in badly initialized points due to weak image gradients in Fig. 3(c). However, our method converges to the true minimum by virtue of the uncertainty-aware ensembles in Fig. 3(d). We highlight a history of a representative feature in Fig. 4 with its sampled ensembles. Remarkably, ensembles of the representative feature point can cover neighboring regions of its true position at the 1st iteration predicted by an IMU in Fig. 4(a). The stochastic gradient computed from these ensembles pulls the pixel position to the correct direction in the minimization problem as in Fig. 4(b). These ensembles exhibit nonnegligible gradients, while the conventional gradient only at the mean point gives too weak gradient to move, as shown in Fig. 4(c).

VI. Experiments

To evaluate EnVIO, we run two sets of experiments. First, we analyze robustness to bad initialization with an increasing initial velocity uncertainty in a virtual environment generated by AirSim [50], [52] in Section VI-A. Second, we evaluate EnVIO in a real-world experiment in Section VI-B. We compare EnVIO to the state-of-the-art methods [11], [26], [27] in terms of estimation accuracy and computation time in a visually low-textured environment, where a visual-inertial sensor is installed in an UAV. We set $n_{en} = 100$ in the following experiments that shows a good tradeoff between estimation accuracy and computation time.

A. Monte Carlo Simulation

We choose the VIODE dataset generated by AirSim to regulate error sources of visual and inertial sensor measurements. The camera nonlinear response function, auto exposure, and

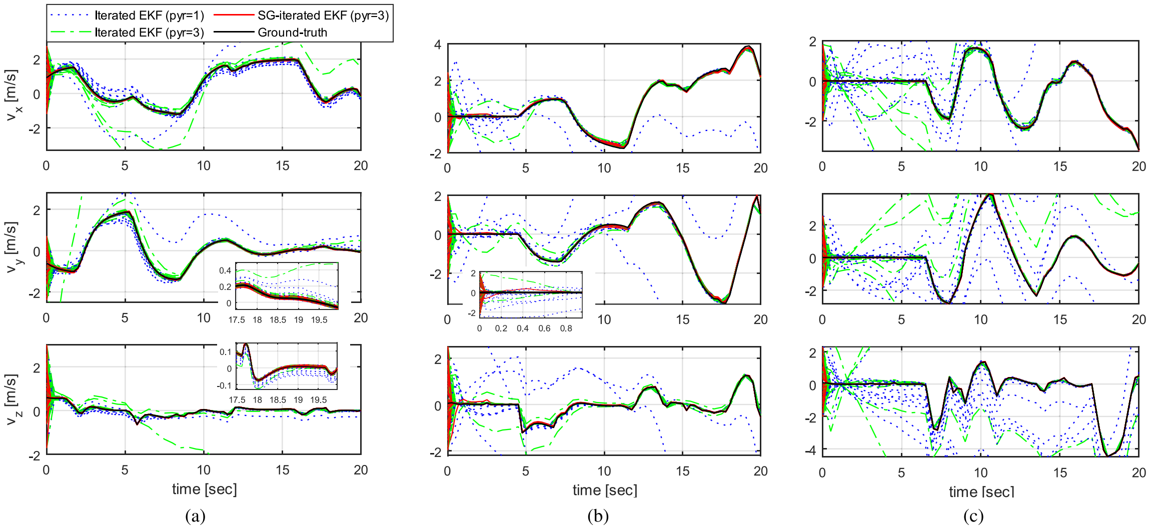


Fig. 6. Velocity estimates of all trials in the Monte Carlo simulation in the first 20 s for $\sigma_v = 1$ m/s in (a) parking_lot, (b) city_day, and (c) city_night.

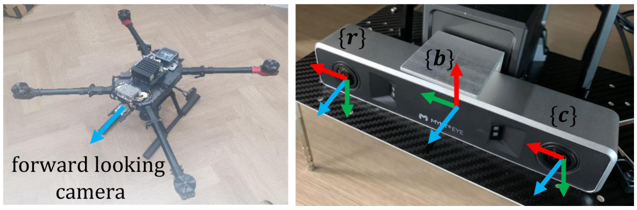


Fig. 7. Custom-built UAV and its MYNTEYE S1030 visual-inertial sensor.

vignetting effect can be calibrated for a real-world sensor as suggested in [53]. However, our objective of this test is to demonstrate the convergence behavior of the stochastic gradient in bad initialization.

Specifically, we adopt the three flight sequences without moving objects. Flight trajectory information is reproduced in Table I for convenience. We generate the true IMU measurement from the ground-truth pose and velocity and add time-varying biases and noises, where sensor specification is based on Analog Devices ADIS16448, as summarized in Table II. The virtual stereo camera outputs 752×480 images with 20 fps and a baseline of 5 cm corrupted by a zero-mean white Gaussian noise with 4 standard deviation in 8-bit intensity.

In the Monte Carlo simulation, random elements include the initial state uncertainty, IMU and camera error sources, and sampling of ensembles. In order to test the robustness to a bad initial point, we evaluate the pose root mean square error (RMSE), normalized estimation error squared (NEES), and the number of failures in the Monte Carlo runs with the increasing initial velocity uncertainty $\sigma_v = \{0.1,\ 0.5,\ 1.0\}$ m/s, as presented in Fig. 5. We declare a trial is failed if the position RMSE is larger than 5% of the flight distance or attitude RMSE is larger than 10 deg. The NEES evaluates the filter consistency and it is defined as

$$\text{NEES} = \frac{1}{n_{mc}\, n_s} \sum_{i=1}^{n_{mc}} \zeta_i^T P_i^{-1} \zeta_i \tag{53}$$

where $n_{mc} = 50$, n_s is the state dimension, and ζ_i, P_i are the actual error and filter covariance in the ith run, respectively.

In Fig. 5, all methods are implemented based on the proposed architecture but with different settings. While *Iterated EKF (pyr=1)* has the maximum 10 iterations on its original resolution, we set the maximum number of iterations as 4, 3, and 3 from the coarsest to the finest pyramid level for *Iterated EKF (pyr=3)* and *SG-iterated EKF (pyr=3)*. Note that, we downsample an image as half resolution at every pyramid level. The stochastic gradient (*SG*) is implemented for the latter.

1) Image Pyramid: The image pyramid can handle the measurement nonlinearity to some extent: *Iterated EKF (pyr=3)* shows better accuracy and consistency than *Iterated EKF (pyr=1)* at $\sigma_v = \{0.5,\ 1.0\}$ m/s in Fig. 5. This would be the reason why this technique is widely adopted in literature. However, the image pyramid still cannot remedy filter divergence due to the bad initialization ($\sigma_v = 1.0$ m/s). This is confirmed by the increasing NEES and failure cases among the Monte Carlo trials in Fig. 5.

2) Stochastic Gradient: The stochastic gradient in *SG-iterated EKF (pyr=3)* reflects image gradients within an uncertain region. In general, this reduces estimation errors, filter inconsistency, and failure runs in combination with the image pyramid in Fig. 5. Figs. 3 and 4 provide an intuitive description for our interpretation: Ensembles provide the correct direction to minimize the cost. We highlight velocity estimates for all trials in the Monte Carlo simulation in the first 20 s of the three virtual trajectories in Fig. 6. *SG-iterated EKF (iter=3)* shows the smallest deviations to the ground-truth among the three cases.

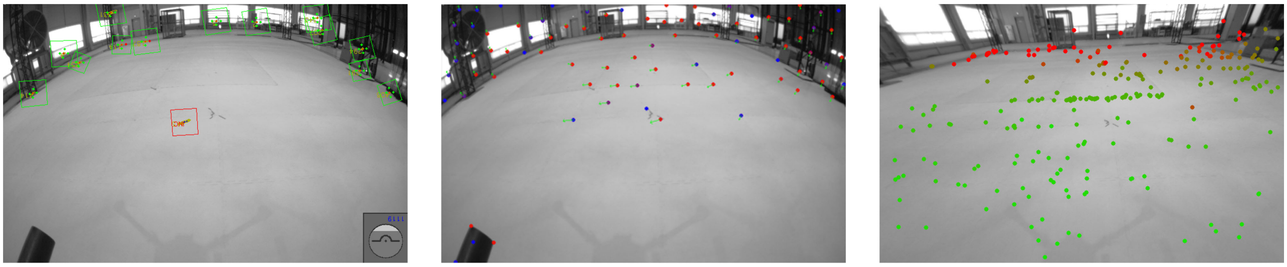
B. Flight Experiment

The objective of this test is to experimentally show that EnVIO can track a camera pose even in a low-textured area that is a huge challenge in visual-inertial navigation. The estimation accuracy is analyzed along with state-of-the-art methods. Furthermore, we investigate a computational budget and validity of the predicted filter covariance.

TABLE III
ABSOLUTE TRAJECTORY ERROR AND AVERAGE COMPUTATION TIME PER A FRAME IN THE FLIGHT TEST

	ROVIO$^{a)}$ [11]			VINS-Fusion$^{a)}$ [27]			Iterated EKF			SG-iterated EKF$^{b)}$ (EnVIO)		
	[deg]	[m]	[ms]	[deg]	[m]	[ms]	[deg]	[m]	[ms]	[deg]	[m]	[ms]
#1 flight	1.167	0.246	**19.8**	1.146	0.189	44.3	**0.578**	0.127	38.1	0.597	**0.107**	38.8
#2 flight	1.165	0.322	**21.7**	2.448	0.447	40.7	1.040	**0.240**	36.2	**1.019**	0.253	37.5
#3 flight	1.815	0.319	**19.8**	1.381	0.145	46.3	0.362	0.152	33.9	**0.339**	**0.117**	35.6
#4 flight	2.991	0.711	**20.8**	2.067	0.241	39.4	0.456	0.250	29.0	**0.449**	**0.237**	33.1
Mean	1.785	0.400	**20.5**	1.761	0.256	42.7	0.609	0.192	34.3	**0.601**	**0.179**	36.3

a) Stereo + IMU configuration is set for ROVIO and VINS-Fusion.
b) EnVIO reports the median value over 5 runs due to the randomness in ensemble sampling.
The bold numbers are the best results (the smallest number) among each flight test.



(a) (b) (c)

Fig. 8. Representative onboard left images with extracted features of (a) ROVIO, (b) VINS-Fusion, and (c) EnVIO (proposed).

We recorded four trajectories using a custom-built UAV equipped with MYNTEYE S1030 (a stereo camera with an IMU) and visual markers for the ground-truth trajectory, as shown in Fig. 7. The sensor outputs a pair of stereo images at 20 fps and raw IMU measurements at 200 Hz. Intrinsic as well as extrinsic calibration parameters of the visual-inertial sensor are calibrated in advance using the Kalibr toolbox [54]. The ground-truth pose is provided by the Qualisys motion capture system with typical mm-level accuracy. The test environment shown in Fig. 8 features a featureless floor: It does not provide enough corners or edges for localization. Fig. 9 shows flight trajectories in which the first two are made by a human pilot, and the last two are controlled by an autopilot.

We implemented EnVIO in ROS Kinetic using C++. The recorded dataset was played on a laptop with Intel i7-7820 CPU at 2.90 GHz. We initialize new features if the current number of features falls below 250 ($n_{\min} = 250$) and set the maximum number of iterations as 10 at the original resolution ($n = 10$). We stop the filter iteration when the innovation change is less than 0.1% or the elapsed time reaches a threshold.

In order to evaluate the absolute trajectory error (ATE) [55], we align the first 100 estimated poses (5 s) to its corresponding ground-truth poses. Table III summarizes ATEs and an average computation time in the same CPU per a frame for ROVIO, VINS-Fusion, and EnVIO. Note that, we use open-source packages of ROVIO and VINS-Fusion (without loop-closure), and tune IMU noise parameters according to the sensor we use to compare them as fairly as possible.

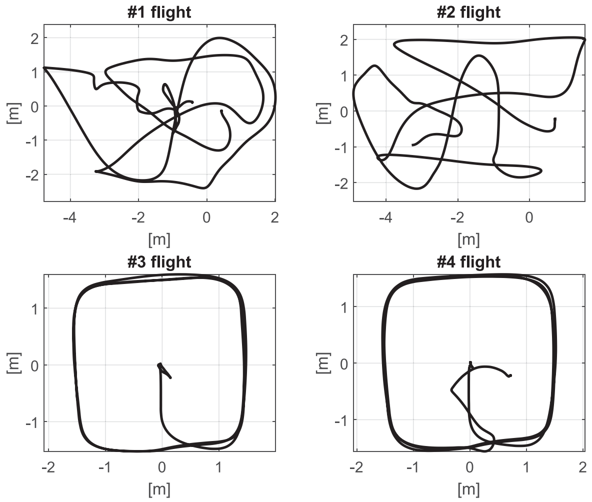
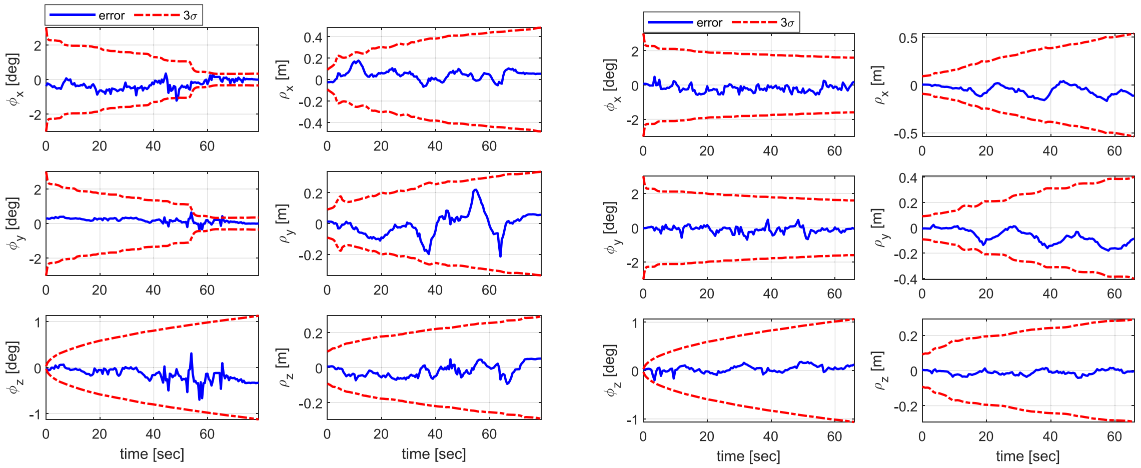


Fig. 9. Ground-truth trajectories in the flight test in which flight distances are 49.3, 44.7, 32.8, and 37.7 m, respectively.

1) ROVIO Versus EnVIO: ROVIO is one of the pioneering photometric VIO that employs pyramidal corner patches in robocentric formulation. High-scored FAST corners are initialized and tracked by minimizing intensity differences. A representative image with tracked feature patches is visualized in Fig. 8(a). The feature selection strategy that extracts a small set of the most salient corners leads to the fastest computation time, but the largest estimation error as reported in Table III. In contrast,



(a)

(b)

Fig. 10. Attitude and position error with their ±3σ bounds in (a) #1 flight and (b) #3 flight.

TABLE IV
TIMING STATISTICS PER A FRAME OF ENVIO

	Time propagation	Filter update	Feature initialization	Mean
Time [ms]	0.3±0.1	19.8 ± 8.4	16.2 ± 5.7	36.3 ± 14.2

EnVIO also utilizes pixels on the low-textured floor in Fig. 8(c), and it contributes to the more accurate pose estimation, as shown in Table III.

2) VINS-Fusion Versus EnVIO: VINS-Fusion extracts uniformly distributed Shi-Tomasi features tracked by the KLT tracker. A windowed BA minimizes reprojection errors to optimize poses and feature depths. Few features on the floor are extracted and tracked, but their tracking length is much shorter than visually rich regions, such as the windows in Fig. 8(b). Therefore, it cannot maintain long-baseline features across the whole image. It seems that this drawback leads to larger errors than our approach. Also, note that the computation time, which is longer than ours, only includes the BA thread.

In contrast, our method is robust to low-textured environments since it does not depend on repeatable features, such as corners and edges. Instead, EnVIO aligns pixel intensities if a nonnegligible image gradient is given. As a result, EnVIO outputs lower pose errors than VINS-Fusion. Furthermore, our lightweight two-view tracking shows 36.3 ms per a frame, as in Table III.

3) Iterated EKF Versus EnVIO: Iterated EKF is based on the proposed architecture without the stochastic gradient. The use of the stochastic gradient can further boost estimation accuracy. It is noticeable that the computation of stochastic gradient for each feature only adds 2.0 ms per a frame on average.

4) Computation Time: Table IV summarizes an average computation time in the four flights with its standard deviation for each crucial step in EnVIO. At the implementation, we divide the measurement Jacobian matrix into subblock matrices since it has a sparse structure for efficient matrix multiplication. The most time-consuming part is the filter update due to matrix inversion for the Kalman gain at each iteration. Our method can run at most 27 fps in terms of the mean computation time, but we believe it would increase with further optimization.

5) Filter Consistency: Fig. 10 draws estimation error along with 3σ bounds to validate the filter consistency. It can be seen that the uncertainty reflects the four unobservable bases (global translation and rotation around the gravity direction), and the autopilot in Fig. 10(b) leads to bigger uncertainties due to limited motion excitation. In the test time, errors are contained in the predicted uncertainty. This confirms the validity of the filter covariance.

VII. CONCLUSION

In this article, we have proposed EnVIO, a framework of photometric VIO coupled with the *stochastic gradient* using uncertainty-aware ensembles. Specifically, we formulated the brightness consistency and derived the filter iteration step on matrix Lie groups. As our key contribution, we derived an optimal image gradient termed the stochastic gradient by minimizing the linearization error within the state uncertainty. The effectiveness of the stochastic gradient was validated through the Monte Carlo simulation at the increasing velocity uncertainty. As expected, pixels with stochastic gradients converged to the true minimum even from bad initialization. Furthermore, the strength of our method was highlighted in the flight test, where most of the scenes were composed of the visually low-textured floor. Since our approach releases the dependence on repeatable visual features, the proposed method outperformed the state-of-the-art VIO in terms of estimation accuracy. The implementation showed the real-time feasibility at most 27 fps in terms of the mean computation time.

In future work, EnVIO can include illumination parameters for robustness to illumination change environments. The estimator can be reformulated as an information filter: The computation time would be further decreased by efficiently calculating the matrix inversion for the Kalman gain. Our interest also includes a visual-inertial mapping module to bound error drift and build a globally consistent map.

ACKNOWLEDGMENT

The authors would like to thank the UAM Vehicle Development Team at UAM Division, Hyundai Motor Group, for providing the aerial vehicle and flight test environment.

REFERENCES

[1] D. S. Bayard *et al.*, "Vision-based navigation for the NASA Mars helicopter," in *Proc. AIAA Scitech Forum*, 2019, pp. 1411–1432.

[2] R. F. S.-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, and A. J. Davison, "SLAM++: Simultaneous localisation and mapping at the level of objects," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 1352–1359.

[3] G. Huang, "Visual-inertial navigation: A concise review," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 9572–9582.

[4] D. Scaramuzza and F. Fraundorfer, "Visual odometry [Tutorial]," *IEEE Robot. Automat. Mag.*, vol. 18, no. 4, pp. 80–92, Dec. 2011.

[5] M. Irani and P. Anandan, "About direct methods," in *Proc. Int. Workshop Vis. Algorithms*, 1999, pp. 267–277.

[6] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, Mar. 2018.

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

[8] F. Steinbrücker, J. Sturm, and D. Cremers, "Real-time visual odometry from dense RGB-D images," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop*, 2011, pp. 719–722.

[9] C. Kerl, J. Sturm, and D. Cremers, "Robust odometry estimation for RGB-D cameras," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2013, pp. 3748–3754.

[10] J. Zubizarreta, I. Aguinaga, and J. M. M. Montiel, "Direct sparse mapping," *IEEE Trans. Robot.*, vol. 36, no. 4, pp. 1363–1370, Aug. 2020.

[11] M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Siegwart, "Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback," *Int. J. Robot. Res.*, vol. 36, no. 10, pp. 1053–1072, 2017.

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

[13] L. Von Stumberg, V. Usenko, and D. Cremers, "Direct sparse visual-inertial odometry using dynamic marginalization," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 2510–2517.

[14] V. Usenko, J. Engel, J. Stückler, and D. Cremers, "Direct visual-inertial odometry with stereo cameras," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2016, pp. 1885–1892.

[15] L. V. Stumberg, P. Wenzel, Q. Khan, and D. Cremers, "GN-Net: The Gauss-newton loss for multi-weather relocalization," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 890–897, Apr. 2020.

[16] L. Von Stumberg, P. Wenzel, N. Yang, and D. Cremers, "LM-Reloc: Levenberg–Marquardt based direct visual relocalization," in *Proc. Int. Conf. 3D Vis.*, 2020, pp. 968–977.

[17] I. Elishakoff and S. H. Crandall, "Sixty years of stochastic linearization technique," *Meccanica*, vol. 52, no. 1, pp. 299–305, 2017.

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

[19] 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, 2013.

[20] K. Sun *et al.*, "Robust stereo visual inertial odometry for fast autonomous flight," *IEEE Robot. Automat. Lett.*, vol. 3, no. 2, pp. 965–972, Apr. 2018.

[21] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "OpenVINS: A research platform for visual-inertial estimation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 4666–4672.

[22] 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.

[23] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. IEEE ACM Int. Symp. Mixed Augmented Reality*, 2007, pp. 225–234.

[24] T. Lupton and S. Sukkarieh, "Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions," *IEEE Trans. Robot.*, vol. 28, no. 1, pp. 61–76, Feb. 2012.

[25] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual-inertial odometry," *IEEE Trans. Robot.*, vol. 33, no. 1, pp. 1–21, Feb. 2017.

[26] 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.

[27] T. Qin, J. Pan, S. Cao, and S. Shen, "A general optimization-based framework for local odometry estimation with multiple sensors," 2019, *arXiv:1901.03638*.

[28] S. Cao, X. Lu, and S. Shen, "GVINS: Tightly coupled GNSS-Visual-Inertial fusion for smooth and consistent state estimation," 2021, *arXiv:2103.07899*.

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

[30] R. Mur-Artal and J. D. Tardós, "Visual-inertial monocular SLAM with map reuse," *IEEE Robot. Automat. Lett.*, vol. 2, no. 2, pp. 796–803, Apr. 2017.

[31] K. Eckenhoff, P. Geneva, and G. Huang, "MIMC-VINS: A versatile and resilient multi-IMU multi-camera visual-inertial navigation system," *IEEE Trans. Robot.*, vol. 37, no. 5, pp. 1360–1380, Oct. 2021.

[32] W. Huang, H. Liu, and W. Wan, "An online initialization and self-calibration method for stereo visual-inertial odometry," *IEEE Trans. Robot.*, vol. 36, no. 4, pp. 1153–1170, Aug. 2020.

[33] L. Carlone and S. Karaman, "Attention and anticipation in fast visual-inertial navigation," *IEEE Trans. Robot.*, vol. 35, no. 1, pp. 1–20, Feb. 2019.

[34] M. Zhang, X. Zuo, Y. Chen, Y. Liu, and M. Li, "Pose estimation for ground robots: On manifold representation, integration, reparameterization, and optimization," *IEEE Trans. Robot.*, vol. 37, no. 4, pp. 1081–1099, Aug. 2021.

[35] B. D. Lucas *et al.*, "An iterative image registration technique with an application to stereo vision," in *Proc. Int. Joint Conf. Artif. Intell.*, 1981, pp. 24–28.

[36] M. Hwangbo, J.-S. Kim, and T. Kanade, "Gyro-aided feature tracking for a moving camera: Fusion, auto-calibration and GPU implementation," *Int. J. Robot. Res.*, vol. 30, no. 14, pp. 1755–1774, 2011.

[37] J. Engel, J. Sturm, and D. Cremers, "Semi-dense visual odometry for a monocular camera," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1449–1456.

[38] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 834–849.

[39] R. Wang, M. Schworer, and D. Cremers, "Stereo DSO: Large-scale direct sparse visual odometry with stereo cameras," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 3903–3911.

[40] X. Gao, R. Wang, N. Demmel, and D. Cremers, "LDSO: Direct sparse odometry with loop closure," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 2198–2204.

[41] H. Luo, C. Pape, and E. Reithmeier, "Hybrid monocular SLAM using double window optimization," *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 4899–4906, Jul. 2021.

[42] S. H. Lee and J. Civera, "Loosely-coupled semi-direct monocular SLAM," *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 399–406, Apr. 2019.

[43] 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. 2016.

[44] T. D. Barfoot, *State Estimation for Robotics*. Cambridge, U.K.: Cambridge Univ. Press, 2017.

[45] T. D. Barfoot and P. T. Furgale, "Associating uncertainty with three-dimensional poses for use in estimation problems," *IEEE Trans. Robot.*, vol. 30, no. 3, pp. 679–693, Jun. 2014.

[46] M. Brossard, A. Barrau, P. Chauchat, and S. Bonnabel, "Associating uncertainty to extended poses for on lie group IMU preintegration with rotating earth," *IEEE Trans. Robot.*, to be published, doi: 10.1109/TRO.2021.3100156.

[47] S. Heo, "EKF-based visual-inertial navigation on matrix lie group with improved consistency," Ph.D. dissertation, Dept. Aerosp. Mech. Eng., Seoul Nat. Univ., Seoul, South Korea, 2018.

[48] J. Civera, A. J. Davison, and J. M. Montiel, "Inverse depth parametrization for monocular SLAM," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 932–945, Oct. 2008.

[49] L. M. Paz, P. Piniés, J. D. Tardós, and J. Neira, "Large-scale 6-DOF SLAM with stereo-in-hand," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 946–957, Oct. 2008.

[50] K. Minoda, F. Schilling, V. Wüest, D. Floreano, and T. Yairi, “VIODE: A simulated dataset to address the challenges of visual-inertial odometry in dynamic environments,” *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 1343–1350, Apr. 2021.

[51] D. Simon, *Optimal State Estimation: Kalman, H. Infinity, and Nonlinear Approaches*. Hoboken, NJ, USA: Wiley, 2006.

[52] S. Shah, D. Dey, C. Lovett, and A. Kapoor, “AirSim: High-fidelity visual and physical simulation for autonomous vehicles,” in *Proc. Field Serv. Robot.*, 2018, pp. 621–635.

[53] J. Engel, V. Usenko, and D. Cremers, “A photometrically calibrated benchmark for monocular visual odometry,” 2016, *arXiv:1607.02555*.

[54] P. Furgale, J. Rehder, and R. Siegwart, “Unified temporal and spatial calibration for multi-sensor systems,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 1280–1286.

[55] Z. Zhang and D. Scaramuzza, “A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 7244–7251.

Jae Hyung Jung (Student Member, IEEE) received the B.S. degree in aerospace engineering from Pusan National University, Busan, South Korea, in 2017, and the M.S. degree in aerospace engineering, in 2019, from Seoul National University, Seoul, South Korea, where he is currently working toward the Ph.D. degree in aerospace engineering with the Department of Aerospace Engineering.

His research interests include visual-inertial navigation for space and mobile robots.

Yeongkwon Choe received the B.S. and M.S. degrees in mechanical and aerospace engineering, in 2015 and 2017, respectively, from Seoul National University, Seoul, South Korea, where he is currently working toward the Ph.D. degree in aerospace engineering with the Department of Mechanical and Aerospace Engineering.

His research interests include the database referenced navigation system, SLAM, and nonlinear filtering.

Chan Gook Park (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in control and instrumentation engineering from Seoul National University, Seoul, South Korea, in 1985, 1987, and 1993, respectively.

In 1998, he was as a Postdoctoral Fellow with Prof. J. L. Speyer about peak seeking control for formation flight with the University of California at Los Angeles, Los Angeles, CA, USA. From 1994 to 2003, he was an Associate Professor with Kwangwoon University, Seoul. In 2003, he joined the Faculty with the School of Mechanical and Aerospace Engineering, Seoul National University, where he is currently a Professor. In 2009, he was a Visiting Scholar with the Department of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, USA. His research interests include advanced filtering techniques, high-precision inertial navigation system (INS), visual-inertial odometry, INS/GNSS/IBN integration, and smartphone-based/foot-mounted pedestrian dead reckoning systems.

Dr. Park was the Chair of IEEE Aerospace and Electronic Systems Society Korea Chapter until 2009.