공학박사 학위논문

Visual-Inertial Navigation System on Matrix Lie Group with Semantic Objects

의미론적 물체를 이용한 리-행렬군 상에서의 영상관성항법 시스템

2023 년 8 월

서울대학교 대학원

항공우주공학과

정 재 형

Visual-Inertial Navigation System on Matrix Lie Group with Semantic Objects

의미론적 물체를 이용한 리-행렬군 상에서의 영상관성항법 시스템

지도교수 박 찬 국

이 논문을 공학박사 학위논문으로 제출함

2023 년 5 월

서울대학교 대학원

항공우주공학과

정 재 형

정재형의 공학박사 학위논문을 인준함

2023 년 6 월

위 원 장 _____ (인)

부위원장 _____ (인)

위 원 _____ (인)

위 원 _____ (인)

위 원 _____ (인)

Visual-Inertial Navigation System on Matrix Lie Group with Semantic Objects

by

Jae Hyung Jung

Submitted to the Department of Aerospace Engineering
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

In Aerospace Engineering at the

SEOUL NATIONAL UNIVERSITY

August 2023

Accepted by _____

Prof. Youdan Kim
Department of Aerospace Engineering, Chairman of Committee

Certified by _____

Prof. Chan Gook Park
Department of Aerospace Engineering, Principal Advisor

_____

Prof. H. Jin Kim
Department of Aerospace Engineering

_____

Prof. Ayoung Kim
Department of Mechanical Engineering

_____

Dr. Sejong Heo
Hyundai Motor Company

To my beloved wife Yeji and my Family

Abstract

Visual-Inertial Navigation System on Matrix Lie Group with Semantic Objects

Jae Hyung Jung
Department of Aerospace Engineering
The Graduate School
Seoul National University

A visual-inertial navigation system (VINS) estimates a state of a moving platform based on visual and inertial sensing: The state includes the position and orientation of the platform and its surrounding map. This has been a backbone of perception systems in autonomy in which accurate and real-time state estimation is indispensable for safe operation. Despite of breakthroughs in literature, vulnerability to degraded conditions hinders its deployment in a fail-critical system. To bring VINS in fail-critical systems one step closer, the estimator should be *fail-safe* meaning it outputs reliable estimates under any circumstances and *fail-aware* so that failures can be automatically detected for recovery.

To this end, this study develops VINS under three design principles. First, the estimator outputs estimation confidence as well as expected value based on sensor uncertainties. State uncertainty is valuable information for itself acting as an index to decide perception failure and for downstream tasks, such as path planning and feedback control. Second, the state-space is modeled on matrix Lie groups: This representation is a natural tool to propagate uncertainty in

rigid body motion. Third, semantic objects are incorporated so that this high-level visual feature provides robustness to appearance and viewpoint changes.

Under the design principle, the theoretical contributions of this study are introducing the optimal image gradient, object-based SLAM formulation, and Gaussian merge on matrix Lie groups. The optimal image gradient minimizes the expectation of the linearization error squared within project uncertainty in an image domain. In object-based SLAM, the unobservable subspace is derived analytically to prove that the bases do not depend on linearized points so that the estimator yields consistent state uncertainty. In addition, a Gaussian midway-merge method is introduced to fuse Gaussian distributions on matrix Lie groups. Object-based SLAM with the presented merge addresses ambiguous object poses due to shape symmetry. As a technical contribution, this study finally integrates all developed elements to build an object-based VINS.

Through intensive simulations and real-world experiments, the optimal image gradient coupled with the photometric visual-inertial odometry shows robustness to the large initial velocity error up to 3 m/s. This shows a clear contrast to the conventional approach in which it fails to cope with such a large initial error. Object-based SLAM formulation on matrix Lie groups yields consistent estimates giving an average normalized estimation error squared as 1.07, while the conventional method gains spurious information along the yaw direction. Ambiguity-aware object SLAM mitigates the large rotation error along the symmetric axis from a six-dimensional pose detector by 49.5% averaged over 10k images. Lastly, this study shows that the integrated system achieves cm-level localization and object mapping accuracy in a room-scale environment.

Keywords: Visual-inertial navigation, state estimation, matrix Lie group, simultaneous localization and mapping

Student Number: 2019-32429

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

1.1 Background and Motivation

A visual-inertial navigation system (VINS) is a state estimator for a moving platform in a three-dimensional space using visual and inertial sensing. By virtue of the small budget and complementary characteristics of a camera and an inertial measurement unit (IMU), visual-inertial fusion has been intensively studied and commercialized in the past two decades. A camera captures rich visual textures that have high potential still not fully exploited in perceiving the world around the platform (robot). However, a camera cannot infer the absolute scale and the gravity direction of a scene without prior knowledge. Instead, an IMU outputs angular rates and specific forces with respect to an inertial frame so that the absolute scale and the gravity directions are encoded in its measurements. It is well understood that the error accumulates over time due to the integration of biased and noisy inertial readings [1]. The accumulated error from an inertial navigation system can be effectively mitigated by constraints from multiple-view images.

Real-world applications, as shown in Fig. 1.1, include the Mars helicopter [2], a robot vacuum cleaner [3], and an entertainment drone [4] to name a few. The strength of VINS lies in its independence to any infrastructure such as a map database or the global navigation satellite system (GNSS). For autonomous

(a) (b) (c)

Figure 1.1: Applications of visual and visual-inertial SLAM: (a) Mars helicopter, (b) entertainment drone, and (c) robot vacuum cleaner.

systems in indoor environments, building canyons, and space exploration in which GNSS signals are not available or reliable, VINS is a building block for autonomy.

Visual-inertial odometry (VIO) and visual-inertial simultaneous localization and mapping (SLAM) are instances to realize VINS. The objective of VIO is to estimate locally consistent poses and optionally three-dimensional maps in a given time window. Relative motion between a robot and its surroundings induces visual parallax by which the ego-motion is inferred as shown in Fig. 1.2a. IMU readings impose temporal constraints (factors) on camera poses, while visual measurements give relative geometry as well founded in [5]. It outputs accurate and reliable pose estimates in a short-term scenario but inevitably suffers from error drift due to the nature of odometry. On the other hand, SLAM builds a globally consistent map so that the drift is bounded. Typically, VIO provides initial estimates to a mapping module in which poses and the structure are refined and passed to VIO, as shown in Fig. 1.2b. Extended Kalman filter (EKF)-based SLAM augments its state vector with landmark positions [6], while EKF-based VIO stochastically clones past poses to marginalize landmark positions to bound the state dimension [7]. Since the pioneering works in visual SLAM [8, 9], it is *de facto* standard that the full SLAM system consists of a

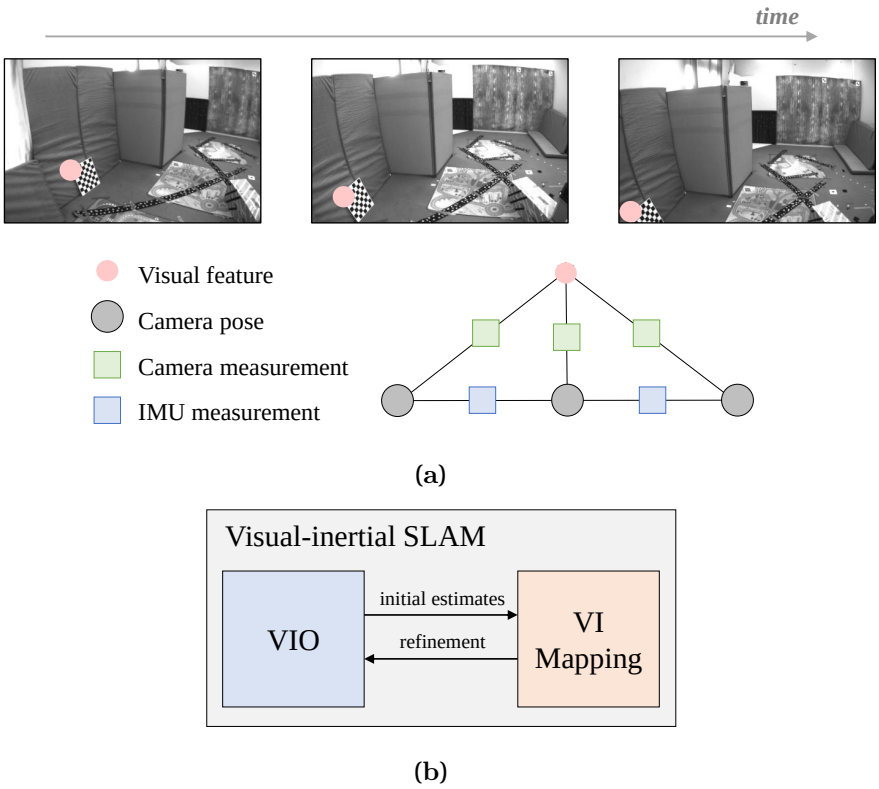


Figure 1.2: (a) Working principle of visual-inertial odometry (VIO), images are from the EuRoC dataset and (b) visual-inertial simultaneous localization and mapping (SLAM) components

VO/VIO module in a faster sampling time, local bundle adjustment for selected keyframes, and loop closure with pose graph optimization to reduce drift in a long baseline.

While SLAM with low-level visual features such as points, lines, and intensities has been widely studied, SLAM with high-level semantic objects is a relatively new topic [10]. In general terms, the goal of *object-level* SLAM is to solve robot and object poses along with associated semantic labels of objects and the surrounding given sensor measurements. In this dissertation, it is assumed that semantic labels are available from a neural network. Instead, there

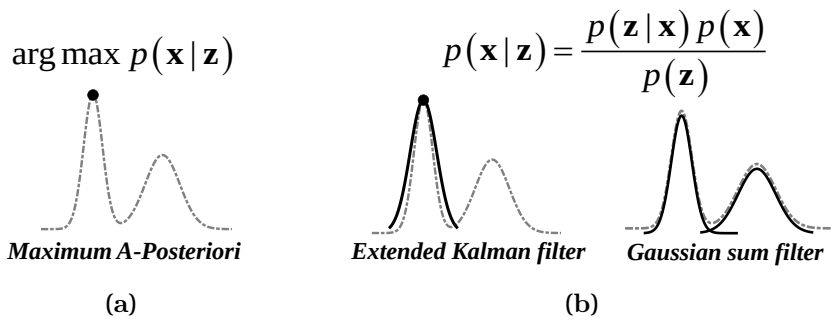


Figure 1.3: $\mathbf{x}$: state and $\mathbf{z}$: measurement; (a) Maximum a-posteriori (MAP) finds the most probable estimates, while (b) Bayesian filtering propagates the underlying probability density function with a problem-specific approximation.

is more focus on jointly estimating robot and object poses, and term this problem as *object-based* SLAM. On the other perspective, a Lie group is a natural tool to model kinematics in robotics. It is a very foundation that describes rigid body motion with rigorous uncertainty representation [11]. In this dissertation, state-space lives in a matrix Lie group so that the estimators are free from the inconsistency problem and the process model is a *group affine system* [12].

Despite of aforementioned seminal works in visual and visual-inertial SLAM, it is still challenging to have highly accurate localization and mapping with valid uncertainty under any circumstances, and it is not straightforward how to fuse inertial and object measurements. Specifically, the motivation for this study is threefold:

- A fail-safe system should cope with large initial uncertainty. Most estimators with linearized systems cannot guarantee stability, thus failure in addressing the large initial error would lead to error divergence and catastrophic outcomes in a fully autonomous system. This study tries to resolve this problem by introducing the optimal image gradient that gives an optimal update direction given state uncertainty.

- A fail-safe system should be robust to ambiguous measurements. Artificial objects such as cups and bricks exhibit shape symmetry from which a multi-modal likelihood function originates. Not properly handling this error source yields inaccurate localization and object mapping. This issue is tackled by tracking each hypothesis while bounding the number of hypotheses in an efficient way.

- Embracing valid state uncertainty paves the way for addressing the above challenges and eventually leads to a fail-aware system. Most previous approaches solve the most probable estimates without explicit estimation confidence, as shown in Fig 1.3a. Knowing the only expected quantity cannot tell whether the current estimate is reliable or not. Valid state uncertainty plays an important role to monitor estimation quality and detect failure to invoke a recovery step. This study formulates VINS under a framework of Bayesian filtering to account for the underlying probability density function, as depicted in Fig. 1.3b.

1.2 Objectives and Contributions

Toward a fail-safe and fail-aware VINS[1], the objective of this dissertation is to estimate the position, velocity, and orientation of a six-dimensional rigid body along with estimation uncertainties. The estimator propagates the underlying probability density function in the form of the expected state along with its covariance matrix, and it should be tractable in a real-time and low-powered system. In particular, for a fail-safe system, this dissertation focuses on robustness to the large initial velocity error and ambiguous object pose. For a fail-aware system, this study focuses on estimation consistency meaning it is

[1] *Failure* means a condition that the estimation error diverges such that this cannot be incorporated in downstream tasks.

unbiased and the covariance matrix reflects the actual errors so that failure can be detected for a fail recovery step. To this end, this study develops VINS under three design principles:

- The estimator outputs estimation confidence as well as expected value based on sensor uncertainties.
- The state-space is modeled on matrix Lie groups for rigorous uncertainty estimation.
- Semantic objects are incorporated so that this high-level visual feature provides robustness to appearance and viewpoint changes.

Under the design principle, the main contributions of this dissertation are as follows:

- The *optimal image gradient* is introduced. The novelty lies in the stochastic modeling of an image gradient leading to high robustness to the initial state uncertainty. The gradient is obtained by minimizing the expectation of the linearization error squared and reduced to the conventional gradient in a deterministic setting. In photometric VIO, the proposed image gradient in iterated EKF reflects the state uncertainty giving a more plausible convergent direction. High robustness and consistency to the initial velocity error are shown in a photo-realistic simulation. Real-world drone flight tests demonstrate highly accurate pose estimation when compared to state-of-the-art VINS and real-time feasibility in a low-powered CPU.
- An object-based SLAM problem is formulated on matrix Lie groups. The contribution is an extension of the classical keypoint measurements to object poses so that the consistency problem is resolved in an object-based SLAM problem. This study analytically derives that the unobservable

subspace does not depend on a linearized point. In other words, the estimated covariance correctly captures the actual yaw uncertainty along the gravity direction. The filter consistency and robustness to large orientation errors are thoroughly investigated through a Monte-Carlo simulation. Coupled with a deep neural network for six-dimensional pose detection, evaluation on an open-source driving dataset demonstrates the effectiveness of exploiting object poses and comparable estimation accuracy when compared to the state-of-the-art object-based SLAM methods.

- The *Gaussian mixture midway-merge* method is introduced. The key idea is to predetermine the common tangent space to merge probability density functions on a matrix Lie group. This simple yet effective approach reduces information loss in merging especially when compared to the conventional merge method. This study presents a promising application in an object-based SLAM problem with shape symmetry. It is experimentally shown that the six-dimensional pose detector suffers from a large orientation error for symmetric objects. To solve this challenge, this study adopts Gaussian sum filter to embrace each hypothesis, and the proposed merge method bounds the exponentially increasing hypotheses. The effectiveness of the midway-merge method is demonstrated by preventing estimation failure that occurs in the conventional method due to the large error of pose measurements. Evaluation tests in an open-source dataset with household objects show that the presented ambiguity-aware SLAM yields high robustness to ambiguous pose measurements and faster computing time in merging compared to the previous method.

- Presented methods are integrated for object-based visual-inertial SLAM in a modular basis. The system is constructed by combining the pose de-

tector, VIO, and ambiguity-aware SLAM. Validation in a photo-realistic simulator demonstrates that cm-level localization and object mapping accuracy are achievable in a room-scale environment.

1.3 Organization of the Dissertation

The rest of this dissertation is organized as follows. Section 1.4 reviews related literature divided into three categories depending on how the visual measurement is processed in the context of visual-inertial SLAM.

Chapter 2 covers mathematical preliminaries to build estimators from scratch in this dissertation. The basic notion of matrix Lie groups, useful formulae, and invariant error are reviewed. Bayesian filtering and its realization with certain assumptions are also covered.

Chapter 3 proposes the optimal image gradient and its application in photometric VIO. It starts with the state-space representation on matrix Lie groups with process and measurement models. Then, this chapter presents the key contribution and a practical way to implement it. Simulations and real-world flight experiments along with implementation details are shown.

Chapter 4 presents object-based SLAM formulated on matrix Lie groups. Beginning with the state-space representation, the unobservable subspace is derived to prove filter consistency. Validation in simulated and real-world driving datasets is followed to describe the effectiveness of object modeling on manifold.

Chapter 5 introduces a Gaussian mixture merge method on matrix Lie groups. Detailed mathematical derivation indicates that the proposed approach yields a lower approximation error than the conventional method. This is further supported by a numerical dissimilarity measure on a three-dimensional rotational group. As a promising example, the merge method is coupled with Gaussian sum filter to address ambiguous measurements of symmetric objects

in object-based SLAM.

Chapter 6 integrates all elements developed in this study to build a framework for visual-inertial object-based SLAM. After presenting the overall architecture, simulation results show the effectiveness of introducing object pose measurements in the estimator.

Lastly, Chapter 7 remarks on theoretical contributions and their real impacts on practical applications. This dissertation is concluded with future research that would be built on this study toward fail-critical perception systems.

1.4 Related Work

This dissertation reviews relevant research in the line of visual-inertial navigation in three categories: feature-based, intensity-based, and object-based methods. This literature review is not limited to *visual-inertial* sensing modality but also covers vision-only and multi-sensor approaches that have been developed closely along with VINS.

1.4.1 Feature-based methods

One of the earliest seminal works in visual-inertial navigation includes the multi-state constraint Kalman filter (MSCKF) [7] 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 [13] remedied the filter inconsistency by using the first estimate Jacobian and introduced the term *visual-inertial odometry* (VIO), which implies the nature of estimation drift due to sequence-to-sequence motion estimation. Sun *et al.* [14] implemented a stereo measurement in a framework of MSCKF. More recently, a unified framework called OpenVINS [15] for monocular and stereo configuration was open-sourced.

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.* [16] followed the principle of the keyframe [8] and introduced a marginalization procedure in VIO that preserves the sparsity pattern of the Hessian matrix. With the advent of the IMU preintegration [17, 18], visual-inertial navigation has become more mature. Qin *et al.* [19] proposed a VINS that includes in-flight initialization, visual-inertial bundle adjustment (BA), and appearance-based loop detection with a pose-graph optimization. This was extended to [20, 21] that includes a

multi-sensor configuration and GNSS measurements. ORB-SLAM3 by Campos *et al.* [22] built on its predecessor [23, 24] features a tracking thread using ORB features, local BA thread, and a multi-map data association to seamlessly fuse previously mapped areas. Toward globally consistent localization and mapping, an efficient way to consider loop closure in sliding windowed factor graph optimization has been proposed [25, 26].

Regardless of its implementation methodology, VINS is heading toward robustness to a system failure in a constrained computing platform. Eckenhoff *et al.* [27] developed a multi-IMU multi-camera system that overcomes measurement depletion due to a limited field of view. Similarly, Zhang *et al.* [28] proposed a multi-camera system that tracks visual features across multiple cameras to maintain longer baselines. The asynchronous multi-sensor measurements were interpolated to efficiently model the state space at a low computational budget. Huang *et al.* [29] extended an initialization procedure from a single camera-IMU pair to a stereo camera configuration. Carlone and Karaman [30] introduced a feature selection strategy by maximizing pose estimation accuracy at limited computational resources. Zhang *et al.* [31] devised the motion manifold that constraints a ground vehicle for efficient 6D pose estimation.

In contrast to previous works, this study focuses 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, the presented method does not suffer from outliers from feature mismatching and implicitly solves the feature correspondence by minimizing the photometric error.

1.4.2 Intensity-based methods

A photometric approach, also known as the direct method, minimizes intensity differences rather than a geometric error. It was successfully employed in 2D sparse feature trackers [32, 33]. Extending an optimization parameter to a 6-DOF pose, real-time dense visual odometry (VO) was presented in [34, 35] that maximizes photoconsistency. Kerl *et al.* [36] 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 semi-dense VO. The key idea was to track pixels with non-negligible gradients by modeling photometric as well as geometric disparity uncertainties. This was extended to LSD-SLAM [38] and direct sparse odometry (DSO) [39]. 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 [40], DSO with loop closure [41], visual-inertial DSO [42], and direct sparse mapping [43]. More recently, a multi-dimensional feature map was trained for the direct image alignment in a long-baseline and multi-weather condition [44, 45].

Hybrid approaches [46–48] 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.* [46] proposed semi-direct VO where the short-term tracking is solved by minimizing the photometric error, while windowed BA minimizes a reprojection error built from previously established matching pairs.

VINS with the photometric measurement includes [42, 49, 50], [43-46], where motion prediction from an IMU provides a good initialization for tracking con-

vergence. Among these, the most relevant work to this dissertation is Robust VIO (ROVIO) by Bloesch *et al.* [49], 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. ROVIO as an odometry module was also extended to globally consistent mapping frameworks [51, 52] and multi-sensor fusion [53]. On the other hand, the presented 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, this study suggests an image gradient that is *optimal* in the sense of a linearization error within a projective uncertainty.

1.4.3 Object-based methods

Since a pioneering work of SLAM at the level of objects [10], semantic objects are known to possess geometrically as well as semantically meaningful information for localization and mapping. For instance, semantic objects have a high signal-to-noise ratio in place recognition with a long baseline and appearance change. On top of that, similar objects of a certain class can be efficiently represented by their 3D models and individual poses. Otherwise, every point cloud or voxel should be stored for each object. The early work has been extended to Fusion++ [54] releasing the assumption of having the prior object database, multi-instance dynamic (MID)-Fusion [55] releasing the static object assumption, and visual-inertial MID-Fusion [56] incorporating inertial measurements for robust pose tracking.

In contrast to the aforementioned volumetric representation for objects, primitive shapes are employed to model objects such as spheres [57], ellip-

soids [58], cuboids [59], and cylinders [60]. The insight is that most of the artificial objects and specific natural objects, such as trees are well-fitted in basic shapes. The intersection over union between measured and predicted shapes often describes a likelihood function.

In the aspect of estimator consistency, [61] proved that unobservable bases do not depend on a linearization point in a framework of the invariant extended Kalman filter (IEKF). This is a generalization of keypoint-based SLAM problems. However, previous approaches did not explicitly consider *ambiguous* object measurements.

To deal with multiple hypotheses due to the ambiguity, PoseRBPF [62] represents the marginalized pose distribution by the augmented autoencoder [63], but their method only includes a single object for a detector that ignores the correlation between objects. Fu *et al.* [64] utilized a max-mixture [65] for multimodal pose estimates in the back-end implementation, but the approximation is vulnerable to bad initialization. To overcome the sensitivity in initialization, Lu *et al.* [66] proposed a heuristic technique to re-initialize a hypothesis, but their noise assumption for a mug did not reflect the real-world noise characteristic. Merrill *et al.* [67] introduced the prior keypoint heatmap as an input to a deep neural network, but they did not explicitly consider the noise behavior of symmetric objects. This study focuses on representing the actual noise distribution of a symmetric object and formulating a Kalman filter to account explicitly for measurement uncertainty.

Chapter 2

Preliminaries

2.1 Coordinate Frame Definition and Notations

Throughout this dissertation, 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 and leveled in the gravity direction. Its heading is aligned to that of $\{b\}$ at the beginning. The IMU frame is coincident with $\{b\}$ pointing in forward, right, and down directions. The left camera frame is denoted as $\{c\}$[1] located on the optical center of a camera model pointing in right, down, and forward directions. The right camera frame $\{r\}$ is defined analogously. The object frame $\{o\}$ is coincident with that of a three-dimensional object model. If it is needed to specify a time instance, this study adopts a subscript to a coordinate frame, for example, $\{b_k\}$ means $\{b\}$ at time t_k. It is assumed that spatial and temporal extrinsic parameters are calibrated for $\{c\}$, $\{r\}$, and $\{b\}$.

This study expresses a vector (or a scalar) and a matrix as small and capital letters such as x and X. When a coordinate frame is placed on 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, respectively. Their

[1] This study reuses the notation in a case of a monocular camera where its meaning is clear depending on the context.

dimensions should be clear in the context.

2.2 Matrix Lie Groups

A matrix Lie group G is a *group* as well as a *smooth manifold* where its elements are matrices. A *group* is a set of elements along with an operation that satisfies the four axioms: closure, associativity, identity, and invertibility [11]. In a viewpoint of state estimation, a *smooth manifold* is a constrained surface in a higher dimensional space with unique tangent spaces at every point [68]. Due to the nature of a rotation representation, a state space often evolves on a manifold. A Lie algebra identified on the identity matrix consists of a vector space $\mathfrak{g}$ with a Lie bracket. The hat operator $(\cdot)^\wedge$ transforms an element in $\mathfrak{g}$ to a vector element and the inverse mapping is designated as $(\cdot)^\vee$. Elements in both structures are related through the matrix exponential and logarithm map,

$$\exp(A) = \sum_{n=0}^{\infty} \frac{1}{n!} A^n \tag{2.1}$$

$$\ln(A) = \sum_{n=0}^{\infty} \frac{(-1)^{n-1}}{n} (A - Id)^n \tag{2.2}$$

for a square matrix A. Therefore,

$$\exp(a^\wedge) = A \tag{2.3}$$

$$\ln(A)^\vee = a \tag{2.4}$$

where $a \in \mathbb{R}^N$, $a^\wedge \in \mathfrak{g}$, and $A \in G$[2]. This section will review rotation and pose groups in a three-dimensional (3D) space as a minimum tool to develop materials in this dissertation.

[2]The matrix logarithm is one-to-many, but it is uniquely defined, for instance, if the magnitude of the rotational part in $SO(3)$ is $\|\phi\| < \pi$.

2.2.1 Special Orthogonal Group in 3D

The special orthogonal group in 3D designated as $SO(3)$ is a set of elements with the matrix multiplication and is defined as

$$SO(3) = \left\{ R \mid R \in \mathbb{R}^{3\times 3},\ R^T R = Id,\ \det(R) = 1 \right\} \tag{2.5}$$

where $(\cdot)^T$ is transpose of a matrix and $\det(\cdot)$ is a determinant of a square matrix. In other words, (2.5) means a set of rotation matrices that transform the resolved frame. At the identity, the Lie algebra $\mathfrak{so}(3)$ is a vector space together with the Lie bracket $[\cdot, \cdot]$ where

$$\mathfrak{so}(3) = \left\{ \Phi \mid \Phi \in \mathbb{R}^{3\times 3},\ \Phi = \phi^\wedge \right\} \tag{2.6}$$

$$[\Phi_1, \Phi_2] = \Phi_1 \Phi_2 - \Phi_2 \Phi_1 \tag{2.7}$$

where $(\cdot)^\wedge$ is a skew-symmetric operator in $SO(3)$ such that

$$\phi^\wedge = \begin{bmatrix} 0 & -\phi_z & \phi_y \\ \phi_z & 0 & -\phi_x \\ -\phi_y & \phi_x & 0 \end{bmatrix}. \tag{2.8}$$

The exponential mapping for the rotation vector $R = \exp(\phi^\wedge)$, $\phi \in \mathbb{R}^3$ has a closed-form expression called *Rodrigues' formula.*

$$\exp\left(\phi^\wedge\right) = Id + \frac{\sin\|\phi\|}{\|\phi\|}\phi^\wedge + \frac{1-\cos\|\phi\|}{\|\phi\|^2}(\phi^\wedge)^2. \tag{2.9}$$

where $\|\phi\| = \sqrt{\phi^T \phi}$ is a 2-norm. The logarithm mapping $\phi = \ln(R)^\vee$ is

$$\|\phi\| = \cos^{-1}\left(\frac{\mathrm{tr}(R) - 1}{2}\right) \tag{2.10}$$

$$\phi = \frac{\|\phi\|}{2\sin\|\phi\|}\left(R - R^T\right)^\vee \tag{2.11}$$

where $\mathrm{tr}(\cdot)$ is a trace of a square matrix and it is assumed that $\|\phi\| < \pi$ and $\phi \neq 0$. Given that the $\cos(\cdot)$ is an even function, the ambiguity on the sign is resolved by testing $\exp(\cdot)$ that yields the correct matrix [11].

2.2.2 Special Euclidean Group in 3D

Pose

The special Euclidean group in 3D designated as $SE(3)$ is a set of elements with the matrix multiplication and is defined as

$$SE(3) = \left\{ T = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} \middle| \, T \in \mathbb{R}^{4\times 4},\ R \in SO(3),\ p \in \mathbb{R}^3 \right\} \tag{2.12}$$

where R and p are a rotation matrix in (2.5) and a position, respectively. In other words, (2.12) is a rigid body transformation that conserves the Euclidean distance. At the identity, the Lie algebra $\mathfrak{se}(3)$ is a vector space with the Lie bracket $[\cdot, \cdot]$ where

$$\mathfrak{se}(3) = \left\{ \Xi \mid \Xi \in \mathbb{R}^{4\times 4},\ \Xi = \xi^{\wedge} \right\} \tag{2.13}$$

$$[\Xi_1, \Xi_2] = \Xi_1 \Xi_2 - \Xi_2 \Xi_1. \tag{2.14}$$

In this expression, $(\cdot)^{\wedge}$ in $SE(3)$ is defined as

$$\xi^{\wedge} = \begin{bmatrix} \phi^{\wedge} & \rho \\ 0 & 0 \end{bmatrix} \tag{2.15}$$

$$\xi = \begin{bmatrix} \phi \\ \rho \end{bmatrix} \in \mathbb{R}^6 \tag{2.16}$$

where $\phi^\wedge$ is defined in (2.8)[3]. The exponential mapping in $SE(3)$, $T = \exp(\xi^\wedge)$, has a closed-form expression as

$$\exp(\xi^\wedge) = \begin{bmatrix} \exp(\phi^\wedge) & J_l(\phi)\rho \\ 0 & 1 \end{bmatrix} \tag{2.17}$$

where the closed-form expression for ϕ was shown in (2.9). The Jacobian $J_l(\phi)$ can be derived as

$$J_l(\phi) = Id + \frac{1 - \cos\|\phi\|}{\|\phi\|^2}\phi^\wedge + \frac{\|\phi\| - \sin\|\phi\|}{\|\phi\|^3}\left(\phi^\wedge\right)^2. \tag{2.18}$$

The Jacobian matrix is known as the left Jacobian of $SO(3)$ [11]. The logarithm mapping $\xi = \ln(T)^\vee$ includes steps of obtaining ϕ from (2.11) and

$$p = J_l^{-1}(\phi)\rho. \tag{2.19}$$

The matrix, $T \in SE(3)$ can be constructed from the above step. The adjoint matrix of $T \in SE(3)$ satisfies $T\xi^\wedge T^{-1} = (\mathrm{Ad}_T\xi)^\wedge$ and is

$$\mathrm{Ad}_T = \begin{bmatrix} R & 0 \\ p^\wedge R & R \end{bmatrix} \in \mathbb{R}^{6\times 6}. \tag{2.20}$$

Extended Pose

This study models the state space of a visual-inertial system on a matrix Lie group and derives their corresponding errors on the vector elements of the Lie algebra. As introduced in the invariant extended Kalman filter [12], the so-

[3] As the convention in robotics [11], the hat operator is overloaded where its definition is dependent on the input argument.

called extended pose is defined as

$$SE_2(3) = \left\{ X = \begin{bmatrix} R & p & v \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \middle| \, R \in SO(3),\ p, v \in \mathbb{R}^3 \right\} \tag{2.21}$$

where R, p, and v represent the attitude, position, and velocity of a robot with respect to a reference frame. That is the velocity is augmented in the pose group. Note that this study expresses the robot's attitude as in Section 2.2.1. Its associated Lie algebra is

$$\mathfrak{se}_2(3) = \left\{ Z \mid Z \in \mathbb{R}^{5\times 5},\ Z = \zeta^\wedge \right\} \tag{2.22}$$

$$[Z_1, Z_2] = Z_1 Z_2 - Z_2 Z_1. \tag{2.23}$$

In this expression, $(\cdot)^\wedge$ in $SE_2(3)$ is defined as

$$\zeta^\wedge = \begin{bmatrix} \phi^\wedge & \rho & \nu \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{2.24}$$

$$\zeta = \begin{bmatrix} \phi \\ \rho \\ \nu \end{bmatrix} \in \mathbb{R}^9 \tag{2.25}$$

where $\phi^\wedge$ is defined in (2.8). As before, elements of $X \in SE_2(3)$ and $\zeta \in \mathfrak{se}_2(3)$ are exactly converted to each other by the matrix exponential and logarithm mapping,

$$X = \exp(\zeta^\wedge) \tag{2.26}$$

$$\zeta = \ln(X)^\vee. \tag{2.27}$$

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

$$\exp\left(\zeta^{\wedge}\right) = \begin{bmatrix} \exp\left(\phi^{\wedge}\right) & J_l(\phi)\rho & J_l(\phi)\nu \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.28}$$

where the closed-form for $SO(3)$ and the left Jacobian J_l can be found in (2.18) and (2.9), respectively. The logarithm mapping $\zeta = \ln(X)^{\vee}$ includes steps of obtaining ϕ from (2.11) and

$$\rho = J_l^{-1}(\phi)\, p \tag{2.29}$$

$$\nu = J_l^{-1}(\phi)\, v. \tag{2.30}$$

Note that $\exp(\cdot)$ and $\ln(\cdot)$ are locally bijective mappings due to the ambiguity in every $\|\phi\| = 2\pi n$ with n a non-zero integer. $SE(3)$ is obtained when eliminating the velocity entries of $SE_2(3)$.

2.3 Baker-Campbell-Hausdorff formula

The *Baker-Campbell-Hausdorff* (BCH) formula expresses the compound of the multiplication of matrix exponential In general, the BCH formula is expressed by an infinite series. For completeness, this section reproduces the first several terms of the formula from [11],

$$\begin{aligned} \ln(\exp(A)\exp(B)) = A + B &+ \frac{1}{2}[A,B] \\ &+ \frac{1}{12}[A,[A,B]] - \frac{1}{12}[B,[A,B]] - \frac{1}{24}[B,[A,[A,B]]] \\ &- \frac{1}{720}\left([[[[A,B],B],B],B] + [[[[B,A],A],A],A]\right) \\ &+ \frac{1}{360}\left([[[[A,B],B],B],A] + [[[[B,A],A],A],B]\right) \\ &+ \frac{1}{120}\left([[[[A,B],A],B],A] + [[[[B,A],B],A],B]\right) + \cdots \end{aligned} \tag{2.31}$$

where A and B are square matrices, $[\cdot,\cdot]$ is a Lie bracket, for instance, as defined in (2.7). However, an approximated formula is required to develop error equations. Therefore, given $x \in \mathbb{R}^N$, $x^\wedge \in \mathfrak{g}$, an approximated BCH formula is,

$$\ln\left(\exp(x_1^\wedge)\exp(x_2^\wedge)\right)^\vee \approx x_1 + \mathcal{J}_l(-x_1)^{-1}x_2 \tag{2.32}$$

where the higher-order term $O(\|x_2\|^2)$ is assumed to be 0. As a general expression in matrix Lie groups the Jacobian matrix is

$$\mathcal{J}_l(x) = \sum_{n=0}^{\infty} \frac{\mathrm{ad}(x)^n}{(n+1)!} \tag{2.33}$$

where $\mathrm{ad}(x)$ is an adjoint of a Lie algebra. A special case for $SO(3)$ was given in (2.18).

Based on (2.32), a useful equation is obtained

$$\begin{aligned}\ln\left(\exp(-x_1^\wedge)\exp((x_1+x_2)^\wedge)\right)^\vee &\approx \ln\left(\exp(-x_1^\wedge)\exp(x_1^\wedge)\exp((\mathcal{J}_l(-x_1)x_2)^\wedge)\right)^\vee \\ &= \mathcal{J}_l(-x_1)\,x_2 \end{aligned} \tag{2.34}$$

if x_2 is small [69].

2.4 Invariant Errors

This study uses the right-invariant error δX [12] that is defined as

$$\delta X = \hat{X}X^{-1} \tag{2.35}$$

where $X \in G$ 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

$$\xi = \log\left(\hat{X}X^{-1}\right)^\vee \tag{2.36}$$

where $\xi^\wedge \in \mathfrak{g}$ and $\exp(\cdot)$, $\log(\cdot)$ are defined in Section 2.2.

2.5 Gaussian Distribution on Matrix Lie Groups

A Gaussian distribution on matrix Lie groups is defined through a vector element at the identity [70],

$$X = \exp(-\xi^\wedge)\hat{X}. \tag{2.37}$$

This study follows the right-invariant error convention [12], $\hat{X} \in G$ is a mean matrix, and $\xi \sim N(0, P)$, a Gaussian distribution with a zero-mean and covariance P. Assuming that ξ is concentrated at the identity and by changing the coordinate $dX = |J_l(\xi)|\, d\,\xi$,

$$\begin{aligned} 1 &= \int_{\mathbb{R}^N} (2\pi)^{-\frac{N}{2}} |P|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\xi^T P^{-1}\xi\right) d\,\xi \\ &= \int_G \eta \exp\left(-\frac{1}{2}\ln(\hat{X}X^{-1})^{\vee^T} P^{-1} \ln(\hat{X}X^{-1})^\vee\right) dX \end{aligned} \tag{2.38}$$

where $\eta = (2\pi)^{-\frac{N}{2}} \left|J_l(\xi)PJ_l(\xi)^T\right|^{-\frac{1}{2}}$. The last line of (2.38) is denoted as $X \sim N_G(\hat{X}, P)$. As introduced in [69], a GM on matrix Lie groups is expressed as

$$\sum_i w_i N_G(\hat{X}_i, P_i) \tag{2.39}$$

where a weight of the ith component w_i satisfies $\sum_i w_i = 1$.

2.6 Moment-Preserving Gaussian Merge in a Vector Space

It is straightforward to merge two Gaussian distributions in a vector space by preserving the first and second-moment [71]. Suppose that a GM model with two components is given,

$$w_1^* N\left(\hat{x}_1, P_1\right) + w_2^* N\left(\hat{x}_2, P_2\right) \tag{2.40}$$

where w_1^* and w_2^* are normalized weights. Then, their moment-preserving merged distribution is $N(\hat{x}_m, P_m)$ where

$$\begin{aligned}
\hat{x}_m &= w_1^* \hat{x}_1 + w_2^* \hat{x}_2 \\
P_m &= w_1^* P_1 + w_2^* P_2 + w_1^* w_2^* (\hat{x}_1 - \hat{x}_2)(\hat{x}_1 - \hat{x}_2)^T. \qquad (2.41)
\end{aligned}$$

2.7 Bayesian filtering

The goal of the Bayesian filtering is to solve for the marginalized distribution at the current time step t_k given a prior and likelihood distribution. This includes the two procedures: *prediction* and *update* steps.

Prediction

$$p\left(x_k|z_{k-1},\cdots,z_0\right)=\int_{x_{k-1}}p\left(x_k|x_{k-1}\right)p\left(x_{k-1}|z_{k-1},\cdots,z_0\right)dx_{k-1} \tag{2.42}$$

Update

$$\begin{aligned}p\left(x_k|y_k,\cdots,y_0\right)&=\frac{p\left(z_k|x_k\right)p\left(x_k|z_{k-1},\cdots,z_0\right)}{p\left(z_k|z_{k-1},\cdots.z_0\right)}\\&=\frac{p\left(z_k|x_k\right)p\left(x_k|z_{k-1},\cdots,z_0\right)}{\int_{x_k}p\left(z_k|x_k\right)p\left(x_k|z_{k-1},\cdots,z_0\right)dx_k}\end{aligned} \tag{2.43}$$

In these expressions, x_k and z_k are the filter state and measurement at the time step t_k, respectively. If the system is linear and Gaussian, then the Bayesian filtering boils down to the Kalman filtering. However, most systems in the real-world exhibit nonlinear or non-Gaussian behavior which is the challenging part to implement Bayesian filtering. Approximated solutions will be covered to develop estimators used in this dissertation.

2.7.1 Kalman Filtering

Given a linear and Gaussian system,

$$\begin{aligned}x_k&=F_{k-1}x_{k-1}+w_{k-1}\\z_k&=H_kx_k+v_k\end{aligned} \tag{2.44}$$

where $w \sim N(0, Q)$ and $v \sim N(0, R)$ are white Gaussian noise vectors and mutually uncorrelated, a closed-form solution is tractable. The prediction step is

$$\begin{aligned} \hat{x}_k^- &= F_{k-1}\hat{x}_{k-1}^+ \\ P_k^- &= F_{k-1}P_{k-1}^+F_{k-1}^T + Q_{k-1} \end{aligned} \tag{2.45}$$

where the superscripts, " + " and " − " means *a-posteriori* and *a-priori*. More specifically,

$$\begin{aligned} \hat{x}_{k-1}^+ &= \mathbb{E}\left[x_{k-1}|y_{k-1}, \cdots, y_0\right] \\ P_{k-1}^+ &= \mathbb{E}\left[(x_{k-1} - \hat{x}_{k-1}^+)(x_{k-1} - \hat{x}_{k-1}^+)^T\right] \end{aligned} \tag{2.46}$$

and

$$\begin{aligned} \hat{x}_k^- &= \mathbb{E}\left[x_k|y_{k-1}, \cdots, y_0\right] \\ P_k^- &= \mathbb{E}\left[(x_k - \hat{x}_k^-)(x_k - \hat{x}_k^-)^T\right]. \end{aligned} \tag{2.47}$$

The difference lies in whether it considers the up-to-date measurement at the current estimate. The update step includes

$$\begin{aligned} S_k &= H_kP_k^-H_k^T + R_k \\ K_k &= P_k^-H_k^TS_k^{-1} \\ \hat{x}_k^+ &= \hat{x}_k^- + K_k(y_k - H_k\hat{x}_k^-) \\ P_k^+ &= (I - K_kH_k)P_k^-(I - K_kH_k)^T + K_kR_kK_k^T. \end{aligned} \tag{2.48}$$

In this expression, S is known as the covariance matrix of the filter innovation $(y_k - H_k\hat{x}_k^-)$ and K is the Kalman gain. This completes a single recursion of the Bayesian filtering.

2.7.2 Extended Kalman Filtering

Given a nonlinear system[4],

$$\begin{aligned} x_k &= f(x_{k-1}) + w_{k-1} \\ y_k &= h(x_k) + v_k \end{aligned} \tag{2.49}$$

where $f(\cdot)$ and $h(\cdot)$ are nonlinear models. $w \sim N(0, Q)$ and $v \sim N(0, R)$ are white Gaussian noise vectors and mutually uncorrelated. The extended Kalman filter (EKF) linearizes the nonlinear models at the current estimate. The prediction step is

$$\begin{aligned} \hat{x}_k^- &= f(\hat{x}_{k-1}^+) \\ P_k^- &= \hat{F}_k P_{k-1}^+ \hat{F}_k^T + Q_{k-1} \end{aligned} \tag{2.50}$$

where the Jacobian matrix is

$$\hat{F}_k = \left.\frac{df}{dx}\right|_{\hat{x}_{k-1}^+}. \tag{2.51}$$

As in the Kalman filtering, the update step is

$$\begin{aligned} S_k &= \hat{H}_k P_k^- \hat{H}_k^T + R_k \\ K_k &= P_k^- \hat{H}_k^T S_k^{-1} \\ \hat{x}_k^+ &= \hat{x}_k^- + K_k(y_k - h(\hat{x}_k^-)) \\ P_k^+ &= (I - K_k \hat{H}_k) P_k^- (I - K_k \hat{H}_k)^T + K_k R_k K_k^T. \end{aligned} \tag{2.52}$$

where the measurement Jacobian matrix is

$$\hat{H}_k = \left.\frac{dh}{dx}\right|_{\hat{x}_k^-}. \tag{2.53}$$

[4]This section assumes that the noise is linear to ease notations, but this can be generalized with no difficulties.

This finishes a single recursion of the EKF.

2.7.3 Iterated Extended Kalman Filtering

When the linearization point is not close enough to the true state, EKF cannot reasonably capture the true underlying distribution. To improve the linearization error the iterated EKF iterates for better linearization in the update step. To be specific, the update step is iterated until convergence. At the ith iteration,

$$\begin{aligned} S_{k,i} &= \hat{H}_{k,i} P_k^- \hat{H}_{k,i}^T + R_k \\ K_{k,i} &= P_k^- \hat{H}_{k,i}^T S_{k,i}^{-1} \\ \hat{x}_{k,i}^+ &= \hat{x}_k^- + K_{k,i} \left(y_k - h(\hat{x}_{k,i}^-) - \hat{H}_{k,i}(\hat{x}_k^- - \hat{x}_{k,i}^+) \right) \end{aligned} \tag{2.54}$$

where

$$\hat{H}_{k,i} = \left. \frac{dh}{dx} \right|_{\hat{x}_{k,i}^+}. \tag{2.55}$$

After the convergence, the posterior covariance is

$$P_k^+ = (I - K_{k,i}\hat{H}_{k,i}) P_k^- (I - K_{k,i}\hat{H}_{k,i})^T + K_{k,i} R_k K_{k,i}^T. \tag{2.56}$$

2.7.4 Gaussian Sum Filtering

A Gaussian mixture model approximates the non-Gaussian posterior distribution where the early idea dates back to the 70s [72]. The non-Gaussian property possibly stems from nonlinear functions or multi-modal noise distributions. Assume that a linear system with multi-modal noises is given.

$$\begin{aligned} x_k &= F_{k-1} x_{k-1} + \mathrm{w}_{k-1} \\ z_k &= H_k x_k + \mathrm{v}_k \end{aligned} \tag{2.57}$$

$$p(\mathrm{w}) = \sum_i \alpha^i \mathcal{N}\left(\bar{\mathrm{w}}^i,\, Q^i\right)$$
$$p(\mathrm{v}) = \sum_j \beta^j \mathcal{N}\left(\bar{\mathrm{v}}^j,\, R^j\right) \tag{2.58}$$

where the upper index means the corresponding *hypothesis*. If the previous posterior distribution is a Gaussian mixture,

$$p\left(x_{k-1}|z_{k-1}, \cdots, z_0\right) = \sum_l w_{k-1}^l \mathcal{N}\left(\hat{x}_{k-1}^{l+},\, P_{k-1}^{l+}\right), \tag{2.59}$$

substituting this expression into the prediction, (2.42) yields

$$\begin{aligned} &p\left(x_k|z_{k-1}, \cdots, z_0\right) \\ &= \int_{x_{k-1}} \sum_i \alpha_{k-1}^i \mathcal{N}\left(F_{k-1}x_{k-1} + \bar{\mathrm{w}}_{k-1}^i,\, Q_{k-1}^i\right) \sum_l w_{k-1}^l \mathcal{N}\left(\hat{x}_{k-1}^{l+},\, P_{k-1}^{l+}\right) dx_{k-1} \\ &= \sum_{i,l} w_k^{i,l-} \mathcal{N}\left(\hat{x}_k^{i,l-},\, P_k^{i,l-}\right). \end{aligned} \tag{2.60}$$

In this expression, $\hat{x}_k^{i,l-}$, $P_k^{i,l-}$ are the (i,l)th estimates obtained from (2.45) with the corresponding hypothesis in the process noise and the previous distribution. The weight is

$$w_k^{i,l-} = \eta\, \alpha_{k-1}^i w_{k-1}^l \tag{2.61}$$

with the normalizer η to make the sum over all hypotheses 1.

Now rephrase the prior distribution as

$$p\left(x_k|z_{k-1}, \cdots, z_0\right) = \sum_m w_k^{m-} \mathcal{N}\left(\hat{x}_k^{m-},\, P_k^{m-}\right) \tag{2.62}$$

and assume the likelihood function is given as

$$p\left(z_k|x_k\right) = \sum_j \beta^j \mathcal{N}\left(H_k x_k + \bar{\mathrm{v}}_k^j,\, R^j\right). \tag{2.63}$$

Substituting (2.62) and (2.63) into (3.27) yields

$$p\left(x_k|z_k,\cdots,z_0\right)=\sum_{m,j} w_k^{m,j}\,\mathcal{N}\left(\hat{x}_k^{m,j+},\,P_k^{m,j+}\right) \tag{2.64}$$

where $\hat{x}_k^{m,j+}$, $P_k^{m,j+}$ are the (m,j)th estimates computed from (2.48) with the corresponding hypothesis in the prior distribution and measurement noise distribution, respectively. The weight is recursively updated as

$$w_k^{m,j}=\eta\, w_k^{m-}\beta_k^j\gamma_k^{m,j} \tag{2.65}$$

where η is the normalizer. $\gamma^{m,j}$ is the $(m,j)$the model evidence as below.

$$\begin{aligned}
\gamma_k^{m,j} &= (2\pi)^{-\frac{\dim(z)}{2}}\left|S_k^{m,j}\right|^{-\frac{1}{2}}\exp\left(-\frac{1}{2}\epsilon_k^{m,j\,T}(S_k^{m,j})^{-1}\epsilon_k^{m,j}\right)\\
\epsilon_k^{m,j} &= z_k^j-H_k\hat{x}_k^{m-}\\
S_k^{m,j} &= H_kP_k^{m-}H_k^T+R^j
\end{aligned} \tag{2.66}$$

where z_k^j is the realization of the measurement with jth noise hypothesis. This finishes a single recursion of Bayesian filtering.

Chapter 3

Ensemble Visual-Inertial Odometry

This chapter contains the contents of the following journal publication:

J. H. Jung, Y. Choe, and C. G. Park, "**Photometric Visual-Inertial Navigation With Uncertainty-Aware Ensembles**,"
IEEE Transactions on Robotics, vol. 38, no. 4, pp. 2039–2052, Aug. 2022, doi: 10.1109/TRO.2021.3139964.

This chapter describes a visual-inertial navigation system that directly minimizes a photometric error without an explicit data association. The photometric error parametrized by pose and structure parameters is considered where the error 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. This study presents two sets of experiments to demonstrate the strengths of the framework. First, a thorough Monte-Carlo simulation in a virtual trajectory is designed to reveal robustness to large initial uncertainty. Second, it is shown that the proposed framework achieves superior accuracy with efficient computation time over state-of-the-art visual-inertial estimators in a real-world UAV flight, where most scenes are composed

of a featureless floor.

3.1 Introduction

Visual navigation is a fundamental building block for higher-level tasks such as autonomous flight in space exploration [73] and semantic perception [10]. 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 [74]. 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 [75]. The latter *directly* minimizes a photometric error that measures an intensity discrepancy between consecutive images [76]. 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 have been a lot of discussions in 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 [22, 39].

However, a cost function formed by the photometric error is highly non-convex in terms of pose and structure parameters [39]. 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 multi-resolution in a practical point of view [35, 36, 43]. Others employ image patches that account for neighboring pixels [39, 46, 49], provide a better initial point based on an inertial sensor [42, 50], or train a deep neural network to generate a desirable feature map for the optimization problem [44, 45]. 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, this study focuses on an intensity gradient given a projective uncertainty that originates from geometric errors. Inspired by the stochastic linearization in random vibration [77], an *optimal* image gradient is derived in the sense that it minimizes the linearization error within the uncertainty. The proposed gradient is implemented by sampling ensembles from the state uncertainty in a framework of photometric visual-inertial odometry (VIO). There are four key contributions of this chapter as follows.

- A framework of photometric VIO based on iterated extended Kalman filter (EKF) is introduced where the state space is modeled on matrix Lie groups. The photometric method makes the system robust to low-textured scenes, while most visual-inertial navigation systems adhere to repeatable and salient features.

- An optimal intensity gradient is derived so that it accounts for its projective uncertainty in the proposed pipeline, and this leads to robustness to the bad initialization.

- To demonstrate the effectiveness of the proposed image gradient, a thorough Monte-Carlo simulation is presented.

- The proposed method is implemented in real-time using C++ and its estimation accuracy, consistency, and computation time are analyzed in a real-world UAV flight, where most scenes are constituted by a featureless floor. The open-source code [1] is available for the benefit of the research community.

The rest of this chapter is organized as follows. The photometric VIO is developed starting from the state space definition in Section 3.2. After laying the foundation, the proposed intensity gradient is derived in Section 3.3. In Section 3.4, a Monte-Carlo simulation and real-world flight test demonstrate the proposed framework. Finally, Section 3.5 concludes this chapter.

3.2 Visual-Inertial State Estimation

3.2.1 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, the 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 [39], the state space is defined as the current extended pose $X_b^g(t)$, IMU biases $B(t)$, the previous pose when an

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

image is captured $T^g_{b_l}(t)$, and depths function at the previous camera pose $D(t)$, that is

$$\mathcal{X}(t) = \begin{bmatrix} X^g_b(t) & 0 & 0 & 0 \\ 0 & B(t) & 0 & 0 \\ 0 & 0 & T^g_{b_l}(t) & 0 \\ 0 & 0 & 0 & D(t) \end{bmatrix} \tag{3.1}$$

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

$$X^g_b(t) = \begin{bmatrix} R^g_b(t) & p^g_b(t) & v^g_b(t) \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \in SE_2(3) \tag{3.2}$$

$$T^g_{b_l}(t) = \begin{bmatrix} R^g_{b_l}(t) & p^g_{b_l}(t) \\ 0 & 1 \end{bmatrix} \in SE(3). \tag{3.3}$$

The bias and depth function matrices are

$$B(t) = \begin{bmatrix} Id & b_a(t) & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & Id & b_g(t) \\ 0 & 0 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{8\times 8} \tag{3.4}$$

$$D(t) = \begin{bmatrix} 1 & d_1(t) & & 0 & 0 \\ 0 & 1 & & 0 & 0 \\ & & \ddots & & \\ 0 & 0 & & 1 & d_m(t) \\ 0 & 0 & & 0 & 1 \end{bmatrix} \in \mathbb{R}^{2m\times 2m} \tag{3.5}$$

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 $d_j = z_j$. The depth parameterization will be discussed in Section 3.2.3.

The coordinate frame and time argument are omitted in the matrix expression to ease the readability if the context is clear such that

$$\mathcal{X} = (X, b_a, b_g, T, d_1, \cdots, d_m)\,. \tag{3.6}$$

3.2.2 Process Model

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

$$\begin{aligned} a_m(t) &= a_t(t) + b_a(t) + n_a(t) \\ \omega_m(t) &= \omega_t(t) + b_g(t) + n_g(t) \end{aligned} \tag{3.7}$$

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

$$\begin{aligned} \dot{R}(t) &= R(t)\,(\omega_m(t) - b_g(t) - n_g(t))^\wedge \\ \dot{p}(t) &= v(t) \\ \dot{v}(t) &= R(t)\,(a_m(t) - b_a(t) - n_a(t)) + g \\ \dot{b}_a(t) &= n_{wa}(t) \\ \dot{b}_g(t) &= n_{wg}(t) \end{aligned} \tag{3.8}$$

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

$$\delta\mathcal{X}(t) = \exp\left(\zeta(t)^\wedge\right) = \hat{\mathcal{X}}(t)\mathcal{X}(t)^{-1}. \tag{3.9}$$

The vector element at the corresponding tangent space is

$$\zeta = \begin{bmatrix} \phi^T & \rho^T & \nu^T & \delta b_a^T & \delta b_g^T & \phi_l^T & \rho_l^T & \delta d_1 & \cdots & \delta d_m \end{bmatrix}^T \tag{3.10}$$

where ϕ, ρ, and ν are defined in (2.28) 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 (3.9).

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

$$\dot{\zeta}(t) \approx F(t)\,\zeta(t) + G(t)\,w(t) \tag{3.11}$$

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 equation (3.11) is perfect when $\delta b_a = \delta b_g = 0$ and $w = 0$ [12]. State uncertainties are well-captured by the invariant error (2.28) as in $SE(3)$ [70, 78]. As detailed in Appendix B, the Jacobian matrix is turned to be

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

$$F_I(t) = \begin{bmatrix} 0 & 0 & 0 & 0 & -\hat{R}(t) \\ 0 & 0 & Id & 0 & -\hat{p}(t)^\wedge \hat{R}(t) \\ g^\wedge & 0 & 0 & -\hat{R}(t) & -\hat{v}(t)^\wedge \hat{R}(t) \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{3.12}$$

In the implementation, (3.8) and (3.11) are discretized to propagate the mean $\hat{\mathcal{X}}$ and the covariance matrix $P = E\left[\zeta\,\zeta^T\right]$.

3.2.3 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 [35] and with illumination parameter estimation [39] to track the 6-DOF pose of a camera. This model is adopted as a filter measurement to spare the explicit 2D feature tracking.

For the jth feature at t_k, this is written as

$$\begin{aligned} y_j(\mathcal{X}) &= h\left(\varphi(\mathcal{X},\ u_j^l)\right) + n_j \\ &= I_l\left(u_j^l\right) - I_k\left(\varphi(\mathcal{X},\ u_j^l)\right) + n_j \end{aligned} \tag{3.13}$$

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

$$\varphi\left(\mathcal{X},\ u_j^l\right) = \Pi\left(\left(T_{b_k}^g T_c^b\right)^{-1} T_{b_l}^g T_c^b \begin{bmatrix} p_j^{c_l} \\ 1 \end{bmatrix}\right) \tag{3.14}$$

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

$$p_j^{c_l} = \Pi^{-1}\left(u_j^l,\ d_j\right). \tag{3.15}$$

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

$$\begin{aligned} \delta y_j &= y_j - \hat{y}_j \\ &\approx H_j \zeta + n_j. \end{aligned} \tag{3.16}$$

The Jacobian matrix is derived using the chain rule

$$\begin{aligned} H_j &= -\frac{\partial I_k}{\partial \zeta} \\ &= -\frac{\partial I_k}{\partial u_j^k}\frac{\partial u_j^k}{\partial p_j^{c_k}}\frac{\partial p_j^{c_k}}{\partial \zeta} \end{aligned} \tag{3.17}$$

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

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

$$\frac{\partial I_k}{\partial u_j^k} = \nabla I_k(\hat{u}_j^k) \tag{3.18}$$

from which most of the linearization errors originate. An image gradient that minimizes a linearization error will be introduced in Section 3.3. The second block is the 2D-to-3D feature point Jacobian,

$$\frac{\partial u_j^k}{\partial p_j^{c_k}} = \begin{bmatrix} f_u(\hat{p}_{j,z}^{c_k})^{-1} & 0 & -f_u\,\hat{p}_{j,x}^{c_k}(\hat{p}_{j,z}^{c_k})^{-2} \\ 0 & f_v(\hat{p}_{j,z}^{c_k})^{-1} & -f_v\,\hat{p}_{j,y}^{c_k}(\hat{p}_{j,z}^{c_k})^{-2} \end{bmatrix} \tag{3.19}$$

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

$$\frac{\partial p_j^{c_k}}{\partial \zeta} = \hat{R}_k^T \begin{bmatrix} -\left(\hat{p}_j^g\right)^\wedge & Id & \cdots & \left(\hat{p}_j^g\right)^\wedge & -Id & \cdots & \hat{R}_l\hat{p}_j^{c_l}\hat{d}_j^{-1} & \cdots \end{bmatrix} \tag{3.20}$$

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

The inverse depth parameterization [79] 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, the proposed filter uses a photometric measurement where the majority of nonlinearity comes from an image intensity. A feature depth is initialized by a stereo baseline with enough parallax. That is

why this study chooses the depth parameterization in the current implementation. However, the presented approach can include far features using inverse depth parameterization as suggested in [80] without any difficulties.

3.2.4 Iterated EKF on Matrix Lie Groups

The iterated EKF is a local maximum a posteriori estimator in a single step [11] that iteratively minimizes a weighted sum of costs until convergence. In the robust VIO (ROVIO) [49], the authors presented iterated EKF formulations that account for rotations and bearing vectors that live in a manifold. In this chapter, however, the filter update step is derived in matrix Lie groups that include $SE_2(3)$, which is a proper group representation for an inertial navigation system.

The objective is to maximize

$$\begin{aligned}\hat{\mathcal{X}}_k &= \underset{\mathcal{X}_k}{\operatorname{argmax}}\, p\left(\mathcal{X}_k \middle| \mathbf{y}_{0:k},\, a_m(t_{0:k}),\, \omega_m(t_{0:k})\right) \\ &= \underset{\mathcal{X}_k}{\operatorname{argmax}}\, p\left(\mathbf{y}_k \middle| \mathcal{X}_k\right) p\left(\mathcal{X}_k \middle| \mathbf{y}_{0:k-1},\, a_m(t_{0:k}),\, \omega_m(t_{0:k})\right) \end{aligned} \tag{3.21}$$

where a density function of the matrix Lie group is indirectly defined by its corresponding Lie algebra [70] and $\mathcal{X}_k = \mathcal{X}(t_k)$, $\mathbf{y}_{0:k} = \mathbf{y}(t_{0:k})$. Here, $\mathbf{y}_k$ is a vector that collects all measurements at t_k. This is equivalent to

$$\begin{aligned}\hat{\mathcal{X}}_k &= \underset{\mathcal{X}_k}{\operatorname{argmin}}\, \|\mathbf{y}_k - \mathbf{h}(\mathcal{X}_k)\|^2_{R_k^{-1}} + \left\|\log\left(\hat{\mathcal{X}}_k^- \mathcal{X}_k^{-1}\right)^\vee\right\|^2_{(P_k^-)^{-1}} \\ &\approx \underset{\zeta_{k,i}}{\operatorname{argmin}} \left\|\mathbf{y}_k - \mathbf{h}(\mathcal{X}^+_{k,i-1}) - \mathbf{H}_{i-1}\zeta_{k,i}\right\|^2_{R_k^{-1}} \\ &\qquad + \left\|\log\left(\hat{\mathcal{X}}_k^- (\mathcal{X}^+_{k,i-1})^{-1}\right)^\vee + \zeta_{k,i}\right\|^2_{(P_k^-)^{-1}} \end{aligned} \tag{3.22}$$

where

$$\mathbf{h}(\mathcal{X}_k) = \begin{bmatrix} h\left(\varphi(\mathcal{X}_k, u_1^l)\right) & \cdots & h\left(\varphi(\mathcal{X}_k, u_m^l)\right)\end{bmatrix}^T, \tag{3.23}$$

$$R_k = \begin{bmatrix} \sigma_1^2 & & \\ & \ddots & \\ & & \sigma_m^2 \end{bmatrix}. \tag{3.24}$$

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 (3.11). $\hat{\mathcal{X}}_k^-$ is a priori of $\mathcal{X}_k$. In the second line in (3.22), the current ith a posteriori has been substituted by the $(i-1)$th iteration, $\mathcal{X}_k = \exp(-\zeta_{k,i}^\wedge)\hat{\mathcal{X}}_{k,i-1}^+$ up to the higher order terms. $\mathbf{H}_{i-1}$ is stacked from (3.17) and linearized at $\hat{\mathcal{X}}_{k,i-1}^+$.

By differentiating the cost in (3.22) with respect to $\zeta_{k,i}$, the update step is given as

$$\begin{aligned} \zeta_{k,i} &= K_{i-1}\left(\mathbf{y}_k - \mathbf{h}(\hat{\mathcal{X}}_{k,i-1}^+)\right) \\ &\quad - (Id - K_{i-1}\mathbf{H}_{i-1}) \log\left(\hat{\mathcal{X}}_k^- (\hat{\mathcal{X}}_{k,i-1}^+)^{-1}\right)^\vee \end{aligned} \tag{3.25}$$

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

$$\bar{\zeta}_{k,i-1} = \log\left(\hat{\mathcal{X}}_k^- (\hat{\mathcal{X}}_{k,i-1}^+)^{-1}\right)^\vee, \quad \bar{\zeta}_{k,0} = 0 \tag{3.26}$$

and a posteriori is updated incrementally

$$\hat{\mathcal{X}}_{k,i}^+ = \exp\left(-\bar{\zeta}_{k,i}^\wedge\right) \hat{\mathcal{X}}_k^- \tag{3.27}$$

where

$$\bar{\zeta}_{k,i} \approx K_{i-1}\left((\mathbf{y}_k - \mathbf{h}(\hat{\mathcal{X}}_{k,i-1}^+) + \mathbf{H}_{i-1}\bar{\zeta}_{k,i-1}\right). \tag{3.28}$$

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

$$P_k^+ = (Id - K_i \mathbf{H}_i) \, P_k^-. \tag{3.29}$$

This is a generalization of the iterated EKF on the vector space: if $\log(\hat{\mathcal{X}}\mathcal{X}^{-1})^\vee$

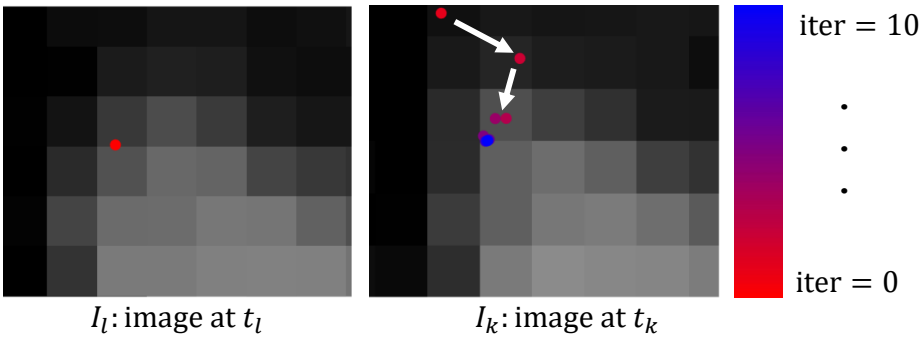


Figure 3.1: A converged example in the VIODE dataset: after a couple of update iterations the pixel point reaches the photometrically as well as the geometrically consistent region.

is replaced by the vector subtraction, the derivation arrives at the equivalent formulation.

Fig. 3.1 shows that the iteration step (3.27) and (3.28) is converged to the photometrically as well as geometrically consistent area by minimizing visual-inertial costs (3.22) in a sequence of temporal images.

3.2.5 Feature Initialization, Tracking, and Marginalization

Input stereo images are processed as a set of feature points that includes a pixel coordinate and its initial depth estimate on the left camera frame. First, incoming stereo images are undistorted and the left grayscale image is converted into a gradient magnitude map. Then, the gradient map is divided into 25×15 grids and the locally strongest pixel greater than a minimum threshold is selected. To maintain uniformly distributed points over an image, an image mask is maintained to ensure a minimum distance among features. As noted in the DSO [39], 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. Badly triangulated features are rejected 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 (3.22). After the convergence of an update step, features at t_l are warped to t_k using a posteriori. In this step, normalized cross-correlations (NCC) are computed in 13×13 patches centered at $\hat{u}_j^l$ and $\hat{u}_j^k$, and features are marginalized if the NCC is smaller than a certain threshold. After the feature tracking and marginalization, the previous pose at t_l is replaced by the current pose at t_k as noted in the 15 line of Algorithm 1. In a covariance domain, marginalization erases the corresponding depth blocks in the covariance matrix.

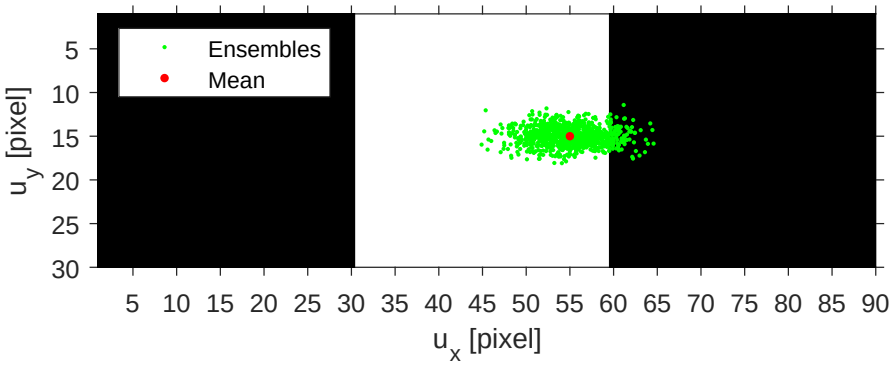
Due to the nature of the tracking mechanism, the measurement noise n_j in (3.13) is colored noise. This can be handled by Kalman filter with a colored noise [81]. From a practical point of view, the measurement noise σ_j is inflated to tackle this unmodeled error.

3.3 Stochastic Gradient

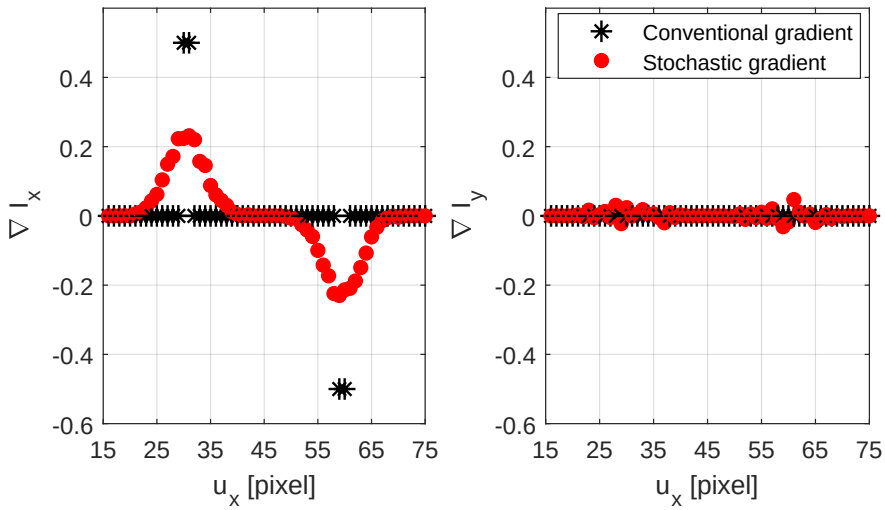
3.3.1 Motivating Example

The tracked points in an image are interpreted as an estimate revealed from its projective uncertainty due to camera pose and depth uncertainties. A simple black and white image in Fig. 3.2a 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. 3.2b. It is assumed that the red pixel is the mean of a 2D Gaussian distribution where ensembles are sampled from the distribution.

In the vicinity of the edges, image gradients are zero: there is no information



(a)



(b)

Figure 3.2: A 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 (3.35) when traveling to the x-direction.

to minimize the photometric error. However, the approach gives non-negligible image gradients derived from the pixel uncertainty as in Fig. 3.2b. That is, it is reasonable to account for the probabilistic property when computing an image gradient. This study introduces a *stochastic gradient* that reflects the projective uncertainty inspired by stochastic linearization [77].

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) [35,36,43] and image patches to include neighboring pixels [39, 49]. However, the proposed approach guarantees an *optimal gradient* in the sense of a linearization error that helps to converge to the correct direction.

3.3.2 Derivation of Stochastic Gradient

In deriving the stochastic gradient, this study focuses on the image gradient which is the first matrix block in (3.17). For convenience, the associated jth feature intensity in time t_k is shown,

$$\mathcal{Y}(u_j^k) = I_k\left(\varphi(\mathcal{X}, u_j^l)\right) + n_{kj} \tag{3.30}$$

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 (3.13). A naive approach is to linearize (3.30) starting from the filter state $\mathcal{X}$. However, it is 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 (3.30) is linearized at the pixel position u_j^k that requires only 2×2 matrix inversion per a feature.

A loss function is defined as

$$L(\mathcal{H}) = I(u) - (I(\hat{u}) + \mathcal{H}\,\delta u) \tag{3.31}$$

where $I(u) = I_k(u_j^k)$, $u = \hat{u} + \delta u$ and $\mathcal{H}$ is an image gradient that has to be found. Then, the expectation of the squared loss function is minimized,

$$\hat{\mathcal{H}} = \underset{\mathcal{H}}{\operatorname{argmin}} \; E\left[L^2\left(\mathcal{H}\right)\right]. \tag{3.32}$$

This can be rewritten as

$$\begin{aligned}\hat{\mathcal{H}} &= \underset{\mathcal{H}}{\operatorname{argmin}} \; E\left[\left(\mathcal{Y}(u) - n - I(\hat{u}) - \mathcal{H}\,\delta u\right)^2\right] \\ &= \underset{\mathcal{H}}{\operatorname{argmin}} \int_{\delta u}\int_{n} \left(\mathcal{Y}(u) - n - I(\hat{u}) - \mathcal{H}\,\delta u\right)^2 p(\delta u, n)\, dn\, d\delta u. \end{aligned} \tag{3.33}$$

Since it is assumed 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

$$\begin{aligned}\frac{dE\left[L^2(\mathcal{H})\right]}{d\mathcal{H}} &= -2\int_{\delta u} \mathcal{Y}(u)\,\delta u^T\, p(\delta u)\, d\delta u \\ &+ 2\, I(\hat{u})\int_{\delta u} \delta u^T p(\delta u)\, d\delta u + 2\,\mathcal{H}\int_{\delta u} \delta u\, \delta u^T p(\delta u)\, d\delta u \end{aligned} \tag{3.34}$$

where the zero-mean measurement noise assumption is employed. Equating (3.34) as zero gives

$$\begin{aligned}\hat{\mathcal{H}} &= \left(\int_{\delta u} \mathcal{Y}(u)\delta u^T p(\delta u)\, d\delta u - I(\hat{u})\int_{\delta u} \delta u^T p(\delta u)\, d\delta u\right)\left(\int_{\delta u} \delta u\, \delta u^T p(\delta u)\, d\delta u\right)^{-1} \\ &= \left(E\left[\mathcal{Y}(u)\,\delta u^T\right] - I(\hat{u})E\left[\delta u^T\right]\right)\left(E\left[\delta u\,\delta u^T\right]\right)^{-1}. \end{aligned} \tag{3.35}$$

It is interesting to note that (3.35) boils down to a numerical differentiation in a noise-free model:

$$\hat{\mathcal{H}} = \frac{I(\hat{u} + \delta u) - I(\hat{u})}{\delta u} \tag{3.36}$$

where $\delta u \in \mathbb{R}$. Note that (3.35) is an *optimal* gradient that minimizes the

mean square of the linearization error. The conventional image gradient (3.18) is replaced by (3.35).

3.3.3 Stochastic Gradient Implementation

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

$$\mathcal{X}^{(i)} = \exp\left(-\zeta^{(i)\wedge}\right)\hat{\mathcal{X}} \tag{3.37}$$

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

$$u_j^{k,(i)} = \varphi\left(\mathcal{X}^{(i)}, u_j^l\right). \tag{3.38}$$

Therefore, it is possible to compute statistical properties of u. The expectation of its deviation from the estimate is

$$E\left[\delta u^T\right] = \frac{1}{n_{en}}\sum_i \left(u_j^{k,(i)} - \hat{u}\right)^T. \tag{3.39}$$

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

$$E\left[\delta u\,\delta u^T\right] = \frac{1}{n_{en}-1}\sum_i \left(u_j^{k,(i)} - \hat{u}\right)\left(u_j^{k,(i)} - \hat{u}\right)^T, \tag{3.40}$$

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

$$E\left[\mathcal{Y}(u)\,\delta u^T\right] = \frac{1}{n_{en}-1}\sum_i \mathcal{Y}(u_j^{k,(i)})\left(u_j^{k,(i)} - \hat{u}\right)^T. \tag{3.41}$$

In the process of the filter update, each ensemble contributes to the stochas-

tic gradient. Thus the proposed method is named as *ensemble visual-inertial odometry* (EnVIO). Algorithm 1 summarizes the overall procedure of EnVIO.

Fig. 3.3 shows a pose tracking result in the parking lot sequence of the VIODE dataset [82] 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. 3.3b. Features are tracked by minimizing (3.22) using the conventional image gradient and the proposed stochastic gradient. Features are trapped in badly initialized points due to weak image gradients in Fig. 3.3c. However, the proposed method converges to the true minimum by virtue of the uncertainty-aware ensembles in Fig. 3.3d. A history of a representative feature in Fig. 3.4 is highlighted 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. 3.4a. The stochastic gradient computed from these ensembles pulls the pixel position to the correct direction in the minimization problem as in Fig. 3.4b. These ensembles exhibit non-negligible gradients, while the conventional gradient only at the mean point gives too weak gradient to move, as shown in Fig. 3.4c.

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`$(a_m(t_{0:n_i}),\ \omega_m(t_{0:n_i}))$

$(\hat{\mathcal{X}}_k^-,\ P_k^-) \leftarrow$ `Time-propagation`$(\hat{\mathcal{X}}_l^+,\ P_l^+,\ a_m(t_{l:k}),\ \omega_m(t_{l:k}))$

for $i = 1$ to n **do**

  for $j = 1$ to m **do**

    $\mathcal{H}_j \leftarrow$ `StochasticGradient`$(\hat{\mathcal{X}}_{k,i-1}^+,\ P_k^-,\ I_k,\ u_j^l) \cdots$ from (3.35)

    $H_j \leftarrow$ `MeasurementJacobian`$(\mathcal{H}_j,\ \hat{\mathcal{X}}_{k,i-1}^+,\ u_j^l) \cdots$ from (3.17)

    $\delta y_j \leftarrow$ `FilterInnovation`$(\hat{\mathcal{X}}_{k,i-1}^+,\ I_l,\ I_k,\ u_j^l) \cdots$ from (3.16)

  end for

  $\hat{\mathcal{X}}_{k,i}^+ \leftarrow$ `Update`$(\hat{\mathcal{X}}_{k,i-1}^+,\ P_k^-,\ \mathbf{H}_{k,i},\ \delta\mathbf{y}_{k,i}) \cdots$ from (3.27)

end for

$P_k^+ \leftarrow$ `CovarianceUpdate`$(P_k^-,\ \mathbf{H}_{k,n}) \cdots$ from (3.29)

Feature tracking: $\{u_j^k\}_{j=1:m} \leftarrow \varphi(\hat{\mathcal{X}}_k^+, \{u_j^l\}_{j=1:m})$

Replace the previous pose to the current one: $T_l \leftarrow T_k$

if $(m < n_{\min})$ **then**

  Initialize new features.

end if

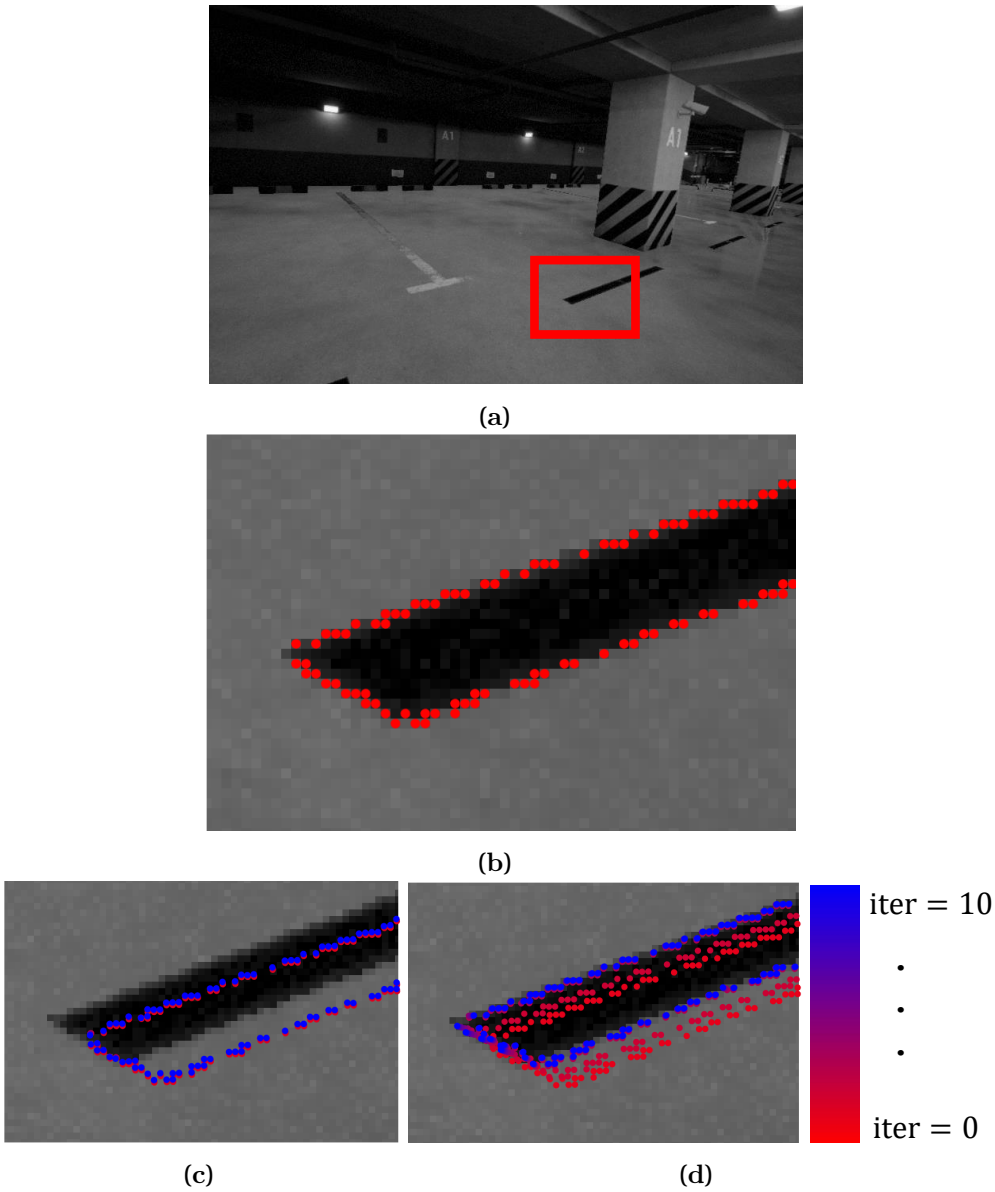


Figure 3.3: An illustrative example in the VIODE parking lot dataset. (a) a reference image at t_l, (b) a close-up of the lane at t_l with high gradient features, (c) pose tracking result at the current time t_k using the conventional gradient, and (d) the proposed stochastic gradient, where the red-to-blue color encodes iteration steps in the iterated EKF.

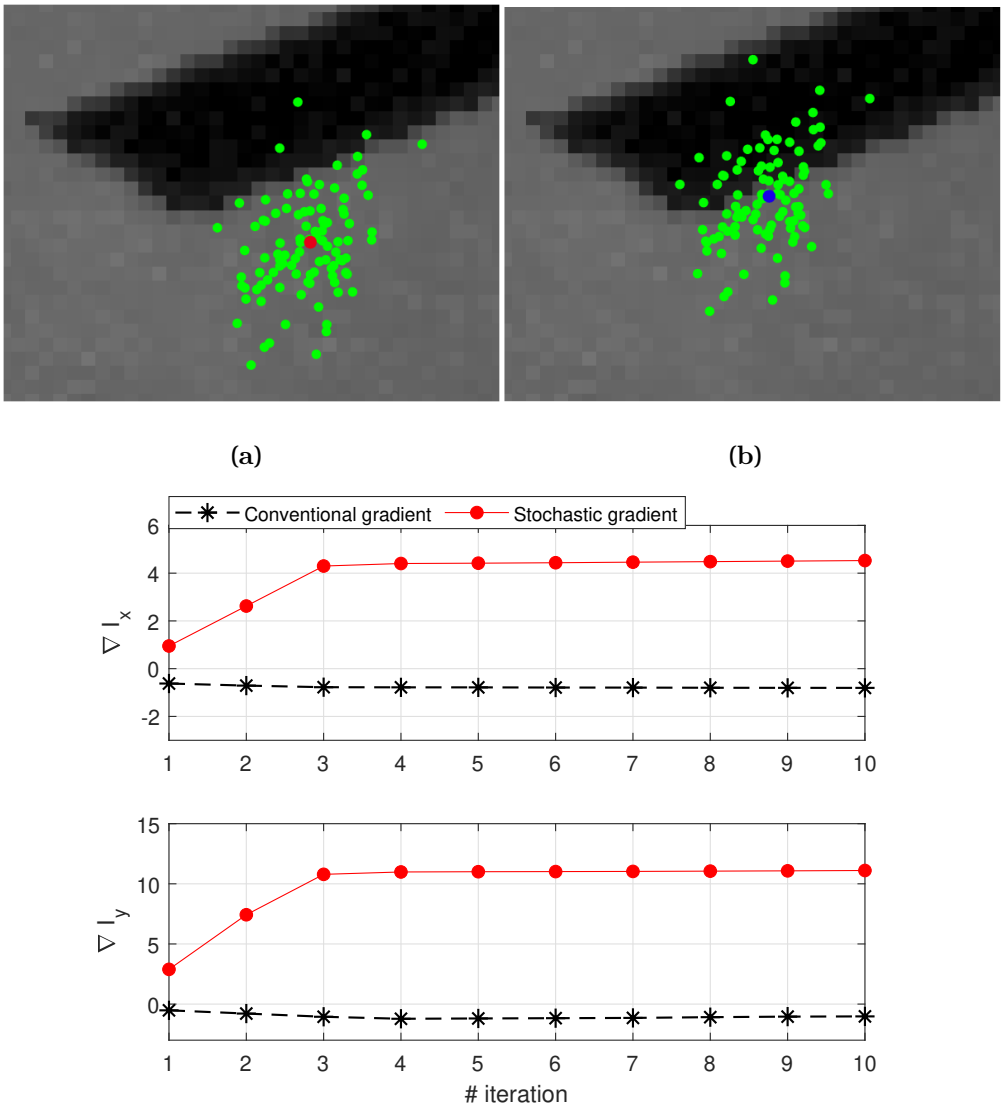


Figure 3.4: A representative pixel coordinate among the extracted features in Fig. 3.3d with sampled ensembles ($n_{en} = 100$) at (a) the 1st iteration and (b) the 10th iteration. (c) Its intensity gradients during the update steps, where the black and red plots correspond to intensity gradients of the representative pixel in Fig. 3.3c and Fig. 3.3d, respectively.

3.4 Experiments

To evaluate EnVIO, two sets of experiments have been conducted. First, this study analyzes robustness to bad initialization with an increasing initial velocity uncertainty in a virtual environment generated by AirSim [82, 83] in Section 3.4.1. Second, EnVIO is evaluated in a real-world experiment in Section 3.4.2. This section compares EnVIO to the state-of-the-art methods [19, 20, 49] in terms of estimation accuracy and computation time in a visually low-textured environment where a visual-inertial sensor is installed in a UAV. The number of ensembles is $n_{en} = 100$ in the following experiments that shows a good trade-off between estimation accuracy and computation time.

3.4.1 Monte-Carlo Simulation

To regulate error sources of visual and inertial sensor measurements, the VIODE dataset generated by AirSim is adopted. Sample images are shown in Fig. 3.5. The camera nonlinear response function, auto exposure, and vignetting effect can be calibrated for a real-world sensor as suggested in [84]. However, the objective of this test is to demonstrate the convergence behavior of the stochastic gradient in bad initialization.

Specifically, the three flight sequences without moving objects are chosen. Flight trajectory information is reproduced in TABLE 3.1 for convenience. The true IMU measurements are generated based on the ground-truth pose and velocity. Then, the true measurements are added by time-varying biases and noises, where sensor specification is based on Analog Devices ADIS16448 as summarized in TABLE 3.2. 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

Table 3.1: Trajectory information in the virtual environment

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

Table 3.2: 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

uncertainty, IMU and camera error sources, and sampling of ensembles. In order to test the robustness to a bad initial point, 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 are evaluated as presented in Fig. 3.6. A failure is declared 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

$$\text{NEES} = \frac{1}{n_{mc}\, n_s} \sum_{i=1}^{n_{mc}} \zeta_i^T P_i^{-1} \zeta_i \tag{3.42}$$

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. 3.6, Fig. 3.7, and Fig. 3.8, 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, while *SG-iterated EKF* (*pyr=1*) includes the stochastic gradient on top of that. The maximum number of iterations are 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 an image is downsampled as half-resolution at every pyramid level.

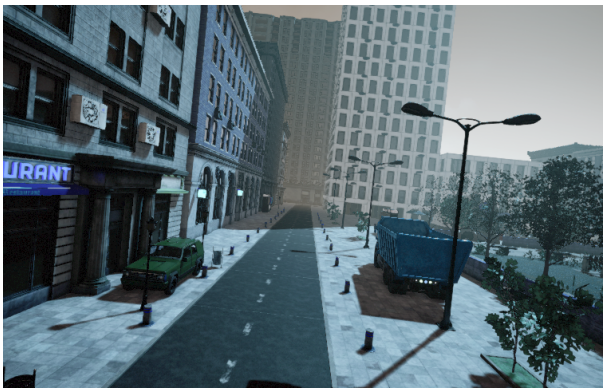
Image pyramid

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. 3.6. This would be the reason why this technique is widely adopted in the 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. 3.6.

Stochastic gradient

The stochastic gradient in *SG-iterated EKF* (*pyr=1*) and *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. 3.6. The more detailed pose ANEES with elapsed time is shown in Fig. 3.8. Fig. 3.3 and 3.4 provide an intuitive description for the interpretation: ensembles provide the correct direction to minimize the cost. Velocity estimates are highlighted for all trials in the Monte-Carlo simulation in the first 20 seconds of the three virtual trajectories in Fig. 3.7. *SG-iterated EKF* (*iter=3*) shows the smallest deviations to the ground-truth among the three cases.

(a)



(b)

(c)

Figure 3.5: Sample onboard images in the VIODE dataset in (a) parking_lot, (b) city_day, and (c) city_night.

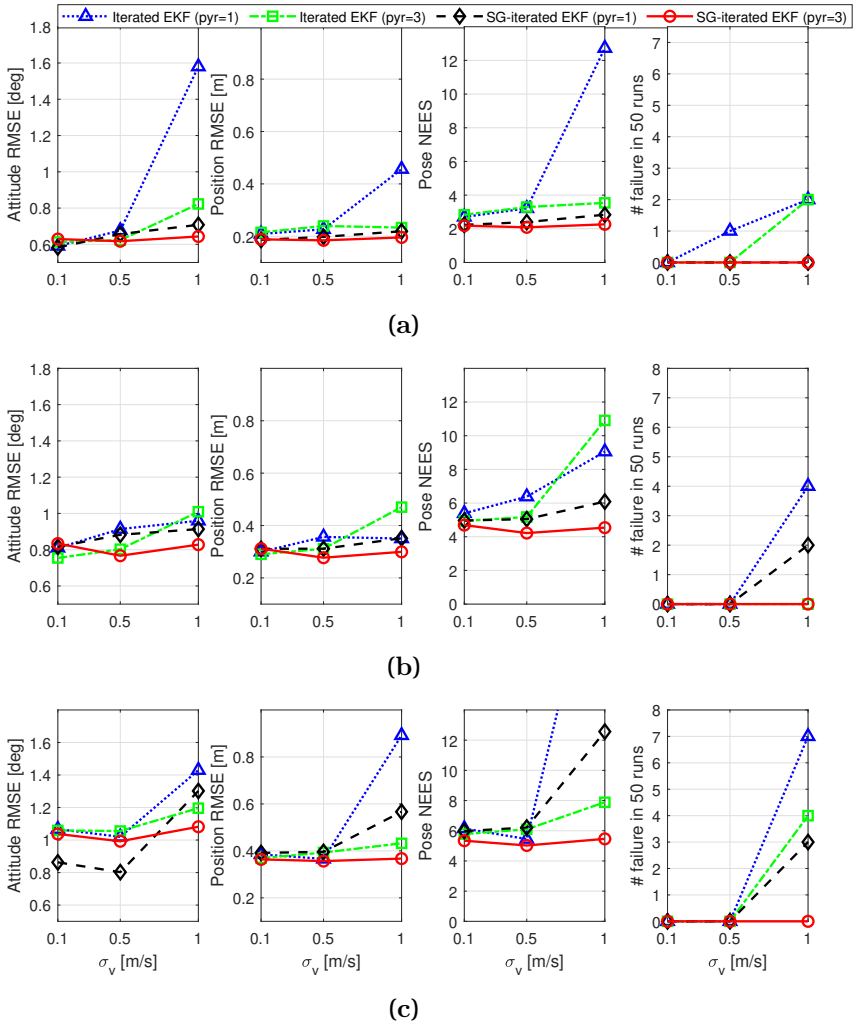


Figure 3.6: Attitude and position RMSE, pose NEES, and the number of failures (position RMSE is larger than 5% of flight distance, or attitude RMSE is larger than 10 deg) of 50 Monte-Carlo runs with the increasing initial velocity uncertainty $\sigma_v = \{0.1,\ 0.5,\ 1.0\}$ m/s in (a) parking_lot, (b) city_day, and (c) city_night.

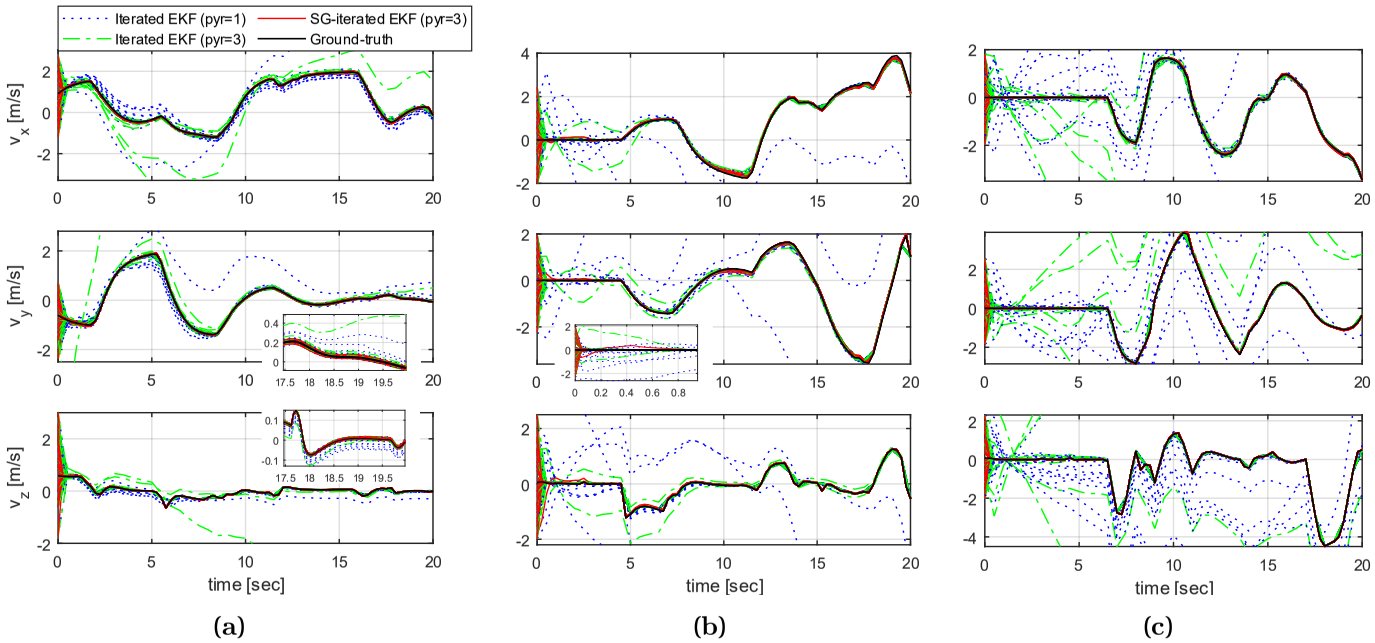
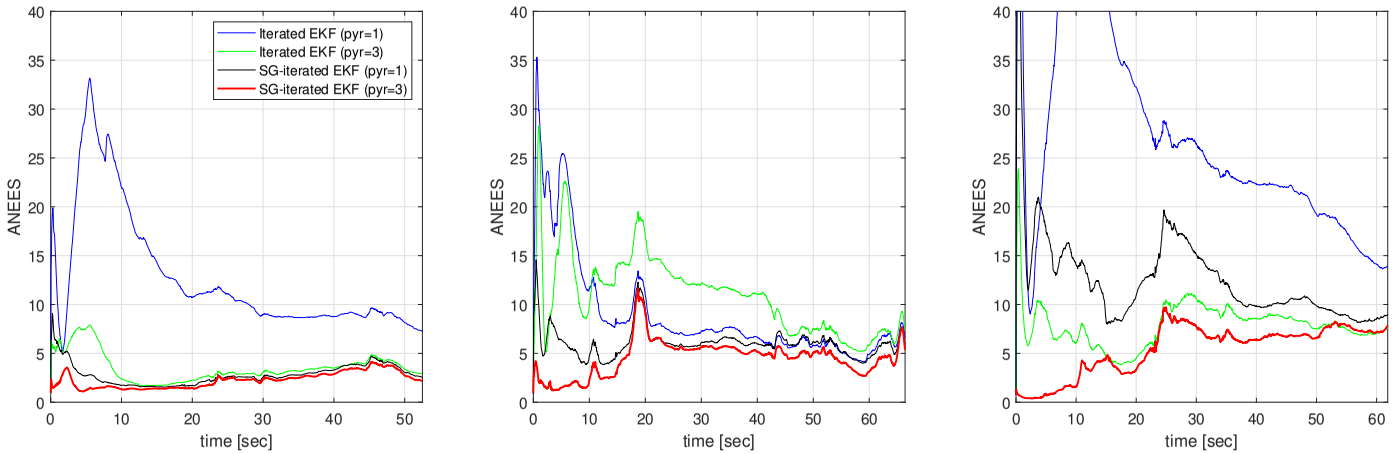


Figure 3.7: Velocity estimates of all trials in the Monte-Carlo simulation in the first 20 seconds for $\sigma_v = 1$ m/s in (a) parking_lot, (b) city_day, and (c) city_night. The results from *SG-iterated EKF* (*pyr=1*) are omitted for clarity.



(a) **(b)** **(c)**

Figure 3.8: Pose Average normalized estimation error squared (ANEES) in (a) parking_lot, (b) city_day, and (c) city_night.

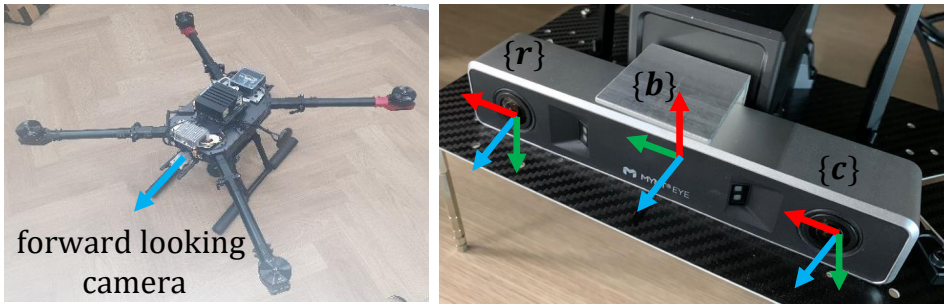


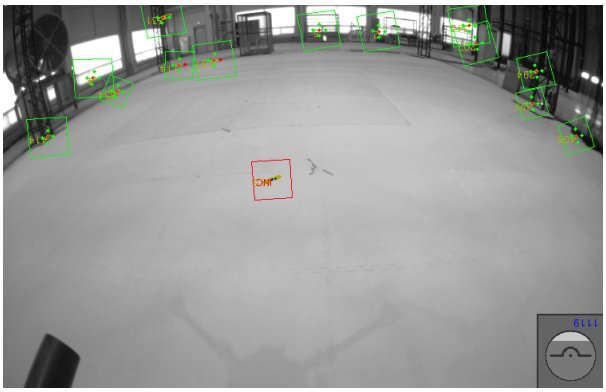
Figure 3.9: A custom-built UAV and its MYNTEYE S1030 visual-inertial sensor.

3.4.2 Flight Experiments

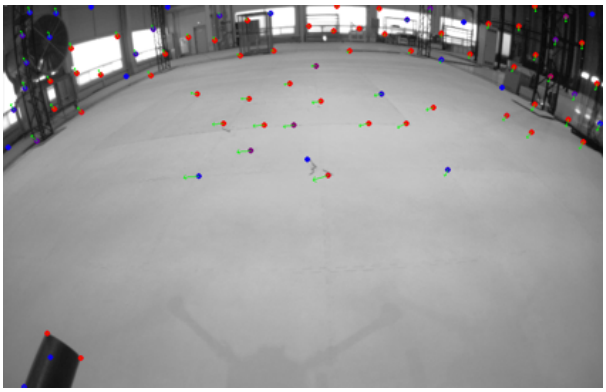
The objective of this test is to experimentally show that EnVIO can track a camera pose even in a low-textured area which is a huge challenge in visual-inertial navigation. The estimation accuracy is analyzed along with state-of-the-art methods. Furthermore, the computational budget and validity of the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

Four trajectories are recorded 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. 3.9. 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 [85]. The ground-truth pose is provided by the Qualisys motion capture system with typical mm-level accuracy. The test environment shown in Fig. 3.10 features a featureless floor: it does not provide enough corners or edges for localization. Fig. 3.11 shows flight trajectories in which the first two are made by a human pilot, and the last two are controlled by an autopilot.

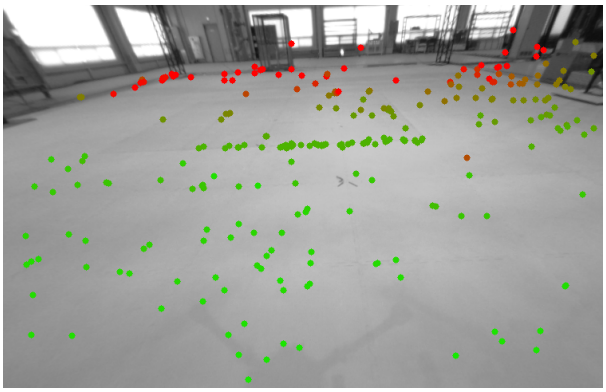
EnVIO is implemented in ROS Kinetic using C++. The recorded dataset



(a)



(b)



(c)

Figure 3.10: Representative onboard left images with extracted features of (a) ROVIO, (b) VINS-Fusion, and (c) EnVIO (proposed).

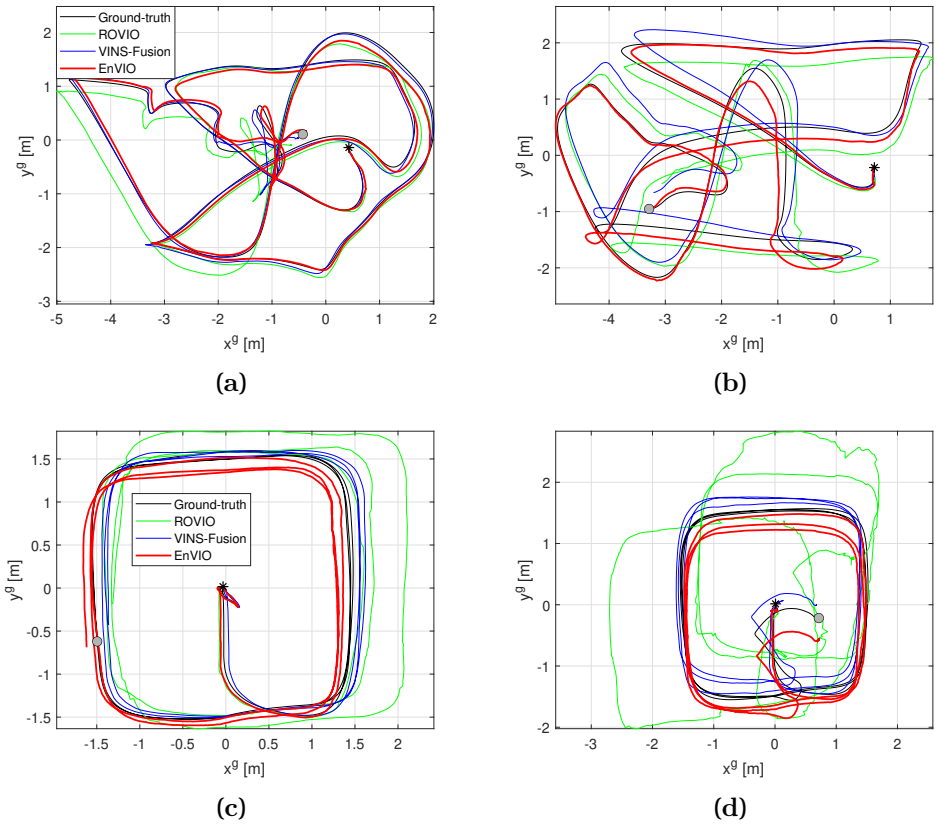


Figure 3.11: The ground-truth and estimated trajectories in the flight tests in which flight distances are (a) #1 flight, 49.3m; (b) #2 flight, 44.7m; (c) #3 flight, 32.8m; (d) #4 flight, 37.7m.

was played on a laptop with Intel i7-7820 CPU at 2.90 GHz. New features are initialized if the current number of features falls below 250 ($n_{\min} = 250$) and the maximum number of iterations is 10 at the original resolution ($n = 10$). The filter iteration is stopped 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) [86], the first 100 estimated poses (5 seconds) are aligned to their corresponding ground-truth poses. TABLE 3.3 summarizes ATEs and an average computation time in the same CPU per frame for ROVIO, VINS-Fusion, and EnVIO. Note that this evaluation uses open-source packages of ROVIO and VINS-Fusion (without loop-closure), and IMU noise parameters are tuned according to the sensor to compare them as fairly as possible.

ROVIO vs. 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. 3.10a. 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 3.3. In contrast, EnVIO also utilizes pixels on the low-textured floor in Fig. 3.10c, and it contributes to the more accurate pose estimation as shown in TABLE 3.3.

VINS-Fusion vs. EnVIO

VINS-Fusion extracts uniformly distributed Shi-Tomasi features tracked by the KLT tracker. A windowed bundle adjustment (BA) minimizes reprojection er-

Table 3.3: Absolute trajectory error and average computation time per frame in the flight test.

	ROVIO[1)] [49]			VINS-Fusion[1)] [20]			Iterated EKF			SG-iterated EKF[2)] (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

1) Stereo + IMU configuration is set for ROVIO and VINS-Fusion.
2) EnVIO reports the median value over 5 runs due to the randomness in ensemble sampling.

rors 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. 3.10b. Therefore, it cannot maintain long-baseline features across the whole image. It seems that this drawback leads to larger errors than the proposed approach. Also, note that the computation time, which is longer than EnVIO, only includes the BA thread.

In contrast, the proposed 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 non-negligible image gradient is given. As a result, EnVIO outputs lower pose errors than VINS-Fusion. Furthermore, the lightweight two-view tracking shows 36.3 ms per frame as in TABLE 3.3.

Iterated EKF vs. EnVIO

Iterated EKF is based on the proposed architecture without the stochastic gradient. Even if the estimator is initialized in a static condition with low motion uncertainty, the use of the stochastic gradient can further boost estimation accuracy. Since the proposed gradient has a strength that paves the way for the correct convergence direction when bootstrapped from the large initial state error, the accuracy gain would be significant if there is a large initial motion uncertainty. This stress case was thoroughly studied in the simulation environment in Section 3.4.1. It is noticeable that the computation of the stochastic gradient for each feature only adds 2.0 ms per frame on average.

Computation time

TABLE 3.4 summarizes an average computation time in the four flights with its standard deviation for each crucial step in EnVIO. At the implementation, the measurement Jacobian matrix is divided into sub-block matrices since it has a

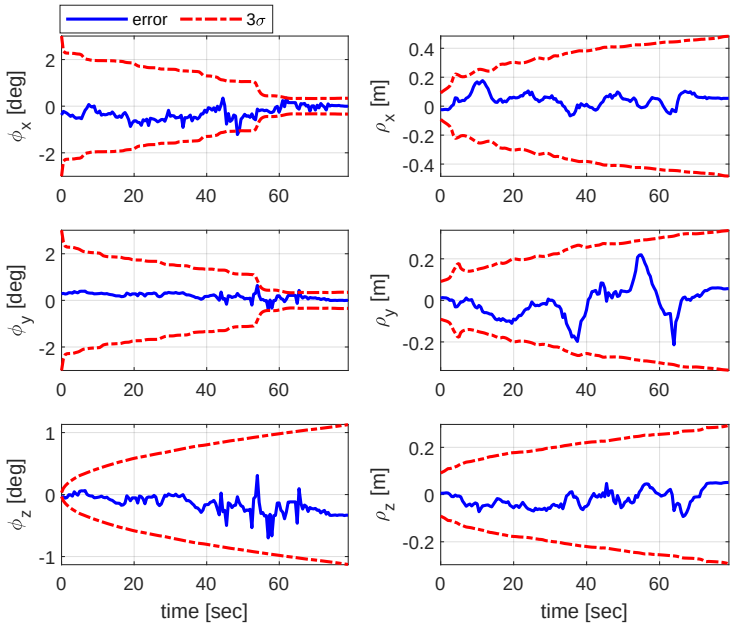
Table 3.4: Timing statistics per frame of EnVIO

	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

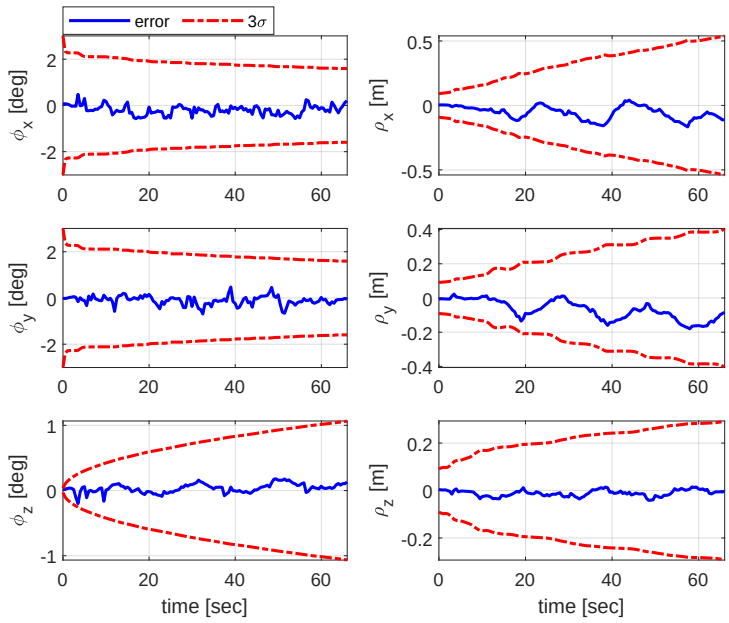
sparse structure for efficient matrix multiplication. The most time-consuming part is the filter update due to matrix inversion for the Kalman gain at each iteration. The proposed method can run at most 27 fps in terms of the mean computation time, but it can be increased with further optimization.

Filter consistency

Fig. 3.12 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. 3.12b 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.



(a)



(b)

Figure 3.12: Attitude and position error with their $\pm 3\sigma$ bounds in (a) #1 flight and (b) #3 flight.

3.5 Conclusion

This chapter has proposed *ensemble visual-inertial odometry* (EnVIO), a framework of photometric VIO coupled with the *stochastic gradient* using uncertainty-aware ensembles. Specifically, this study formulated the brightness consistency and derived the filter iteration step on matrix Lie groups. As the key contribution, an optimal image gradient termed the stochastic gradient is derived 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 the method was highlighted in the flight test, where most of the scenes are composed of the visually low-textured floor. Since the proposed 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. Future work also includes a visual-inertial mapping module to bound error drift and build a globally consistent map.

Chapter 4

Object SLAM with Improved Consistency

This chapter contains the contents of the following conference publication:

J. H. Jung and C. G. Park,
"**Object-based Visual-Inertial Navigation System on Matrix Lie Group,**"
in IEEE International Conference on Robotics and Automation, 2022,
pp. 9499–9505, doi: 10.1109/ICRA46639.2022.9812443.

This chapter proposes a novel object-based visual-inertial navigation system fully embedded in a matrix Lie group and built upon the invariant Kalman filtering theory. Specifically, relative pose measurements of objects are considered and an error equation is derived at the associated tangent space. It is proved that the observability property does not suffer from the filter inconsistency and nonlinear error terms are identically zero at the object initialization. A thorough Monte-Carlo simulation reveals that the proposed approach yields consistent estimates and is very robust to a large initial state uncertainty. Furthermore, this study demonstrates a real-world application to the KITTI dataset with a deep neural network-based 3D object detector. Experimental results report that noises on pose measurements follow a Gaussian-like density matching the assumption. The proposed method improves the localization

and object global mapping accuracy by probabilistically accounting for inertial readings and object pose uncertainties at multiple views.

4.1 Introduction

The invariant extended Kalman filter (IEKF) has been developed based on an invariant dynamic system to the group action. It was introduced by S. Bonnabel [87], where the observer is invariant under the left group action. A. Barrau and S. Bonnabel [12] broadened a considered system that is called the *group affine system* for more practical applications. The IEKF guarantees local stability for a certain set of nonlinear systems. Recently, the same authors [78] introduced a mathematical technique to bear a high-precision inertial navigation system (INS) in the group affine system to account for the Earth's rotation.

In a deterministic sense, the log-linearity [12] is a powerful property to express a nonlinear system in Lie groups by linearized error dynamics on the tangent space. This contributes to the filter stability in part by making the Kalman gain independent to a trajectory. On the other hand, the banana-shaped distribution due to sensor noises is known to be well-represented by the exponential coordinate in robot navigation as a stochastic point of view [70, 78, 88]. These are the main reasons why the IEKF outputs superior navigation results in terms of the estimator accuracy and consistency when compared to the conventional EKF [12, 89–91]. Especially in a simultaneous localization and mapping (SLAM) problem, the Lie group structure captures the correct dimension of the unobservable subspace without any artificial remedies [90, 92].

However, most of the previous work remained on the vector space in their output model, such as a relative point measurement (SLAM), the gravity or magnetic vector (attitude heading reference system). This chapter extends a

measurement model on the vector space to the matrix Lie group $SE(3)$ that represents a rigid body pose to imbue the IEKF into an object-based visual-inertial navigation system (VINS). In a strict sense, the Kalman gain of the proposed model depends on a vehicle trajectory, but the derived model does obey the true dimension of the unobservable space yielding consistent estimates.

On the other hand, in contrast to low-level visual features such as intensities, points, or lines, objects possess rich information to localize robots, and in turn, they can geometrically as well as semantically perceive the world around themselves. With prior knowledge of an object such as dense point clouds or CAD models [10, 93] or without prior information such as spheres, cuboids, or ellipsoids [57–59, 94], objects serve as visual landmarks to build semantically meaningful maps for high-level tasks. Despite impressive results on low-level feature-based VINS by virtue of a complementary characteristic of visual and inertial sensors [15, 16, 19, 22, 95], there are few works on object-level navigation in the line of visual-inertial fusion. Furthermore, most single image 3D object detectors concerns about a *local pose* of objects in which a measurement quality is hard to be guaranteed. However, a *global pose* is required to understand a global map of a scene, and the global mapping accuracy can be improved when fusing multiple local pose estimates. This motivated us to fuse visual and inertial measurements to build an object-based VINS.

To achieve consistency and robustness to a large initial error, a new EKF-based estimator is formulated that is fully embedded in the matrix Lie group. It is proved that the proposed method captures the true observability characteristic and the nonlinear error terms are identically zero at the object initialization. The object-based VINS outputs a 6-DOF vehicle pose with semantically labeled 6-DOF objects in a global map. The below summarizes key contributions as follows.

- Object-based visual-inertial fusion on the matrix Lie group with observability and nonlinear error analysis.

- A thorough validation in a Monte-Carlo simulation that reveals the consistency and robustness of the estimator.

- Real-world demonstration using object measurements from a deep neural network 3D object detector in the KITTI dataset [96] with a comparison to the state-of-the-art object SLAM methods.

4.2 Visual-Inertial Object SLAM Formulation

4.2.1 Problem Definition

The objective of this chapter is to estimate the current state $\mathcal{X}(t_k)$ with estimate confidence given the initial state $\mathcal{X}(t_0)$, noisy IMU measurements $\{a_m, \omega_m\}_{t_1:t_k}$ and relative pose measurements of objects $\{T_o^b\}_{t_1:t_k}$. The state includes the body attitude, position, velocity X_b^g which is defined in (3.2), accelerometer bias b_a, gyroscope bias b_g and object poses $T_{o_j}^g$,

$$\mathcal{X} = \begin{bmatrix} X_b^g & 0 & 0 & 0 & \cdot & 0 \\ 0 & B_a & 0 & 0 & \cdot & 0 \\ 0 & 0 & B_g & 0 & \cdot & 0 \\ 0 & 0 & 0 & T_{o_1}^g & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \ddots & \cdot \\ 0 & 0 & 0 & 0 & \cdot & T_{o_m}^g \end{bmatrix} \tag{4.1}$$

where IMU biases [89] are expressed as

$$B_a = \begin{bmatrix} Id & b_a \\ 0 & 1 \end{bmatrix}, \quad B_g = \begin{bmatrix} Id & b_g \\ 0 & 1 \end{bmatrix}. \tag{4.2}$$

The proposed state representation $\mathcal{X}$ lives in $SE_2(3) \times \mathbb{R}^3 \times \mathbb{R}^3 \times SE(3) \times \cdots \times SE(3)$.

By defining the right-invariant error $\exp(\delta x^\wedge) = \hat{\mathcal{X}}\mathcal{X}^{-1}$, vector elements in the associated Lie algebra are

$$\begin{aligned}
\delta x &= \begin{bmatrix} \zeta_b^T & \delta b_a^T & \delta b_g^T & \xi_1^T & \cdots & \xi_M^T \end{bmatrix}^T \\
\zeta_b &= \ln\left(\hat{X}_b X_b^{-1}\right)^\vee \\
\delta b_a &= \hat{b}_a - b_a \\
\delta b_g &= \hat{b}_g - b_g \\
\xi_j &= \ln\left(\hat{T}_j T_j^{-1}\right)^\vee
\end{aligned} \tag{4.3}$$

where the overhead hat $\hat{(\cdot)}$ is an estimate of the corresponding quantity. It is simplified that X_b^g as X_b and the jth object pose $T_{o_j}^g$ as T_j for readability.

4.2.2 Process Model

The current state is propagated by an inertial navigation system f in the continuous-time,

$$\frac{d}{dt}X_b(t) = f_{a_m,\omega_m}\left(X_b(t),\ B_a(t),\ B_g(t)\right) \tag{4.4}$$

where f is a group affine system when eliminating the IMU biases [12]. The linearized model at the current estimate is derived as

$$\frac{d}{dt}\begin{bmatrix} \zeta_b(t) \\ \delta b_a(t) \\ \delta b_g(t) \end{bmatrix} \approx F(t)\begin{bmatrix} \zeta_b(t) \\ \delta b_a(t) \\ \delta b_g(t) \end{bmatrix} + G(t)n(t) \tag{4.5}$$

where $n(t)$ follows a Gaussian process, $GP\left(0,\ Q\delta(t-\tau)\right)$ with a power spectral density matrix Q. $F(t)$ and $G(t)$ are Jacobian matrices of the error and noise vector as detailed in Appendix B. It is remarkable to note that the linearized

system (4.5) is *perfect* when IMU biases and noises are zero [12].

It is assumed that objects are static in $\{g\}$ such that $d\xi_j/dt = 0$. At the implementation, (4.4) and (4.5) are discretized to propagate the expectation of the state $\hat{X}_b$ and filter covariance $E[\delta x\, \delta x^T]$.

4.2.3 Measurement Model on SE(3)

Previous work on the IEKF was constrained to a limited set of measurement models on the vector space to make a linearized system independent of the vehicle trajectory [12, 87]. To deal with the relative pose observation, a measurement model is extended to the matrix Lie group $SE(3)$. In a strict sense, the measurement model is not invariant observation in the line of [12], but it will be shown that its unobservable subspace does not depend on the vehicle trajectory in 4.2.5.

Objects are expressed as a 6-DOF rigid body pose in the global frame $\{g\}$, and their poses are observed in the body frame $\{b\}$. Denoting $Y_j \in SE(3)$ as a relative pose measurement of the jth object pose in the body frame, a measurement equation at time t_k is

$$\begin{aligned} Y_j(t_k) &= h(\mathcal{X}_b(t_k),\, n_j(t_k)) \\ &= T_b(t_k)^{-1} T_j(t_k) \exp(n_j(t_k)^\wedge) \end{aligned} \tag{4.6}$$

where T_b is the upper left 4×4 matrix of X_b, and n_j is a zero-mean white Gaussian noise uncorrelated to the system noise in the tangent plane. Y_j is measured by a 3D object detector, and a deep learning-based method [97] is adopted in 4.3.2. Time notation t_k is omitted for clarity.

Using the Baker-Campbell-Hausdorff (BCH) formula and defining the right-

invariant error $\exp(\epsilon_j^\wedge) = \hat{Y}_j Y_j^{-1}$, a linearized model is derived

$$\begin{aligned}\exp(\epsilon_j^\wedge) \approx\ & \exp(\mathrm{Ad}_{\hat{T}_b^{-1}}(\xi_j - \xi_b - \mathrm{Ad}_{\hat{T}_j} n_j)^\wedge \\ & + O\left(\|\xi_b\|\,\|\xi_j\|\right)\,)\end{aligned} \tag{4.7}$$

where ξ_b is the upper 6×1 vector of ζ_b in (4.3). In the tangent space with a sufficiently small error, Jacobian matrices are obtained as

$$\epsilon_j \approx \mathrm{Ad}_{\hat{T}_b^{-1}}(\xi_j - \xi_b) - \mathrm{Ad}_{\hat{T}_b^{-1}}\mathrm{Ad}_{\hat{T}_j} n_j. \tag{4.8}$$

The filter innovation is defined as

$$r = \begin{bmatrix} \ln\left(\hat{Y}_1 Y_1^{-1}\right)^{\vee\,T} & \cdots & \ln\left(\hat{Y}_M Y_M^{-1}\right)^{\vee\,T} \end{bmatrix}^T \tag{4.9}$$

and a posteriori $\mathcal{X}^+$ is updated from a priori $\mathcal{X}^-$ as

$$\hat{\mathcal{X}}^+ = \exp\left(-(Kr)^\wedge\right)\hat{\mathcal{X}}^-. \tag{4.10}$$

The Kalman gain K and posterior covariance are computed using the prior covariance and the measurement Jacobian matrices (4.8).

4.2.4 Object Initialization

An object is initialized in the filter state at the first detection with the mean as $\hat{T}_j = \hat{T}_b Y_j$. A new object has the following relationship,

$$\xi_j = \xi_b + \mathrm{Ad}_{\hat{T}_j} n_j + O\left(\|\xi_b\|\,\|n_j\|\right). \tag{4.11}$$

Since ξ_b and n_j are statistically independent, the initial covariance matrix is initialized as

$$E[\xi_j \xi_j^T] = E[\xi_b \xi_b^T] + \mathrm{Ad}_{\hat{T}_j} E[n_j n_j^T] \mathrm{Ad}_{\hat{T}_j}^T. \tag{4.12}$$

The cross-correlation terms are initialized based on (4.11) and objects are marginalized if they are not visible for 5 seconds from the latest detection.

The higher order term in (4.7) that depends on the filter state ξ_b, ξ_j is further investigated. Using the BCH formula it is derived as

$$\begin{aligned} O\left(\|\xi_b\| \, \|\xi_j\|\right) = & \hat{T}_b^{-1} \Bigg(-\frac{1}{2}\left[\xi_b^\wedge, \xi_j^\wedge\right] - \frac{1}{12}\left[\xi_b^\wedge, \left[\xi_b^\wedge, \xi_j^\wedge\right]\right] \\ & + \frac{1}{12}\left[\xi_j^\wedge, \left[\xi_b^\wedge, \xi_j^\wedge\right]\right] + \cdots \Bigg) \hat{T}_b \end{aligned} \tag{4.13}$$

where $\left[\xi_b^\wedge, \xi_j^\wedge\right] = \xi_b^\wedge \xi_j^\wedge - \xi_j^\wedge \xi_b^\wedge$ is the Lie bracket. It is remarkable to note that (4.13) is *identically zero* just after the object initialization in a deterministic sense. That is $\xi_j = \xi_b$ from (4.11). This makes the proposed estimator more robust to initial errors since the measurement model is *perfectly* represented by its linearized equation. In contrast, $SO(3)$-EKF parameterized by (4.15) cannot enjoy this property.

4.2.5 Unobservable Subspace

In a linear discrete system, the observability matrix in $t \in [t_0,\ t_k]$ is defined as

$$\mathcal{O} = \begin{bmatrix} H(t_0) \\ H(t_1)\Phi(t_1, t_0) \\ \vdots \\ H(t_k)\Phi(t_k, t_{k-1}) \cdots \Phi(t_1, t_0) \end{bmatrix} \tag{4.14}$$

where H is a measurement matrix, and Φ is a state-transition matrix. Without loss of generality, sensor biases are excluded and only the single jth object is considered for simplification. One can straightforwardly derive the observability matrix including sensor biases and multiple objects.

SO(3)-EKF

Error vectors of the vehicle state ϕ_b, δp_b, δv_b and the jth object ϕ_j, δp_j are defined as

$$\begin{aligned}
\phi_b &= \ln\left(\hat{R}_b R_b^T\right)^\vee \\
\delta p_b &= \hat{p}_b - p_b \\
\delta v_b &= \hat{v}_b - v_b \\
\phi_j &= \ln\left(\hat{R}_j R_j^T\right)^\vee \\
\delta p_j &= \hat{p}_j - p_j
\end{aligned} \tag{4.15}$$

that contradicts the nonlinear errors, ζ_b and ξ_j in (4.3). Consider a time step t_l between t_0 and t_k, then the lth observability matrix block is

$$\begin{aligned}
\mathcal{O}_l^{\text{EKF}} &= H(t_l)\Phi(t_l, t_{l-1})\cdots\Phi(t_1, t_0) \\
&= \begin{bmatrix} R_{b_l}^T & 0 \\ 0 & R_{b_l}^T \end{bmatrix} \times \\
&\begin{bmatrix} -Id & 0 & 0 & Id & 0 \\ \left(p_j - p_{b_l} + \int_{t_0}^{t_l}\int_{t_0}^{t} R_{b_\tau} a(\tau) d\tau dt\right)^\wedge & -Id & -\Delta t_l Id & 0 & Id \end{bmatrix}
\end{aligned} \tag{4.16}$$

where p_j is the translational part of T_j, a is the specific force measured by an accelerometer, and $\Delta t_l = t_l - t_0$. If $\mathcal{O}_l^{\text{EKF}}$ is evaluated at the true state, the nullspace of $\mathcal{O}^{\text{EKF}}$ is

$$\begin{aligned}
N_1 &= \begin{bmatrix} 0 & Id & 0 & 0 & Id \end{bmatrix}^T \\
N_2 &= \begin{bmatrix} g^T & -(p_{b_0}^\wedge g)^T & -(v_{b_0}^\wedge g)^T & g^T & -(p_j^\wedge g)^T \end{bmatrix}^T .
\end{aligned} \tag{4.17}$$

In this expression, g is the gravity vector in $\{g\}$. However, N_2 no longer exists when $\mathcal{O}^{\text{EKF}}$ is evaluated at the linearization point as analogous to the point

model [92]. Please refer to the detailed derivation in Appendix A.

The proposed formulation

Again consider the lth observability matrix block with the error definition in (4.3). $\mathcal{O}_l^{\text{IEKF}}$ is derived using Φ and H in (4.5) and (4.8), respectively.

$$\mathcal{O}_l^{\text{IEKF}} = \text{Ad}_{T_b^{-1}} \begin{bmatrix} -Id & 0 & 0 & Id & 0 \\ -\frac{\Delta t_l^2}{2} g^\wedge & -Id & \Delta t_l Id & 0 & Id \end{bmatrix} \tag{4.18}$$

It turns out that the nullspace bases of $\mathcal{O}^{\text{IEKF}}$ is

$$\begin{aligned} \mathcal{N}_1 &= \begin{bmatrix} 0 & Id & 0 & 0 & Id \end{bmatrix}^T \\ \mathcal{N}_2 &= \begin{bmatrix} g^T & 0 & 0 & g^T & 0 \end{bmatrix}^T \end{aligned} \tag{4.19}$$

where $\mathcal{N}_1$ and $\mathcal{N}_2$ correspond to the global translation and rotation around the gravity direction, respectively. Remarkably, the nullspace does not depend on the linearization point. This solves the filter inconsistency problem that gains spurious information along the unobservable direction due to incorrect linearization. Please refer to the detailed derivation in Appendix B.

4.3 Experiments

4.3.1 Monte-Carlo Simulation

To validate the proposed method, 50 runs of a Monte-Carlo simulation sampling initial state error, IMU and object measurement noises are conducted. The IMU performance is based on the inertial sensor equipped in OXTS RT3003 [96]. A virtual camera has a field of view of 81 deg and 29 deg for horizontal and vertical views, respectively. Object measurement noises are based on the noise analysis in 4.3.2: $\sigma_{p_j} = 3\,\mathrm{m}$ and $\sigma_{R_j} = 8\,\mathrm{deg}$. A constant speed circular vehicle trajectory with a radius of 16 m and 12 static objects is generated, as shown in Fig. 4.1.

The objective of this experiment is to show that 1) the proposed method, *Proposed filter*, does not gain fictitious information along the unobservable bases by inspecting the averaged normalized estimation error squared (ANEES) and 2) *Proposed filter* is robust to an initial attitude error when compared to a conventional $SO(3)$-parametrized EKF ($SO(3)$-*EKF*). The ANEES at a certain instance is defined as

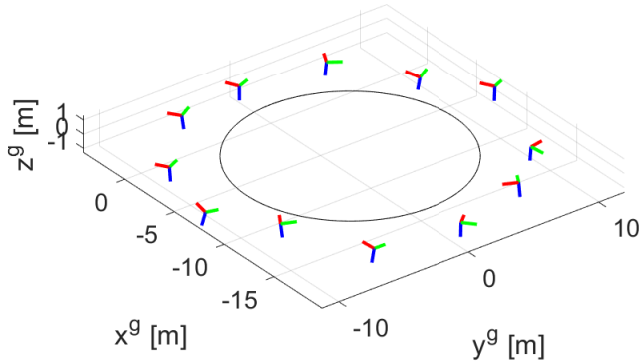
$$\mathrm{ANEES} = \frac{1}{MN} \sum_{i=1}^{M} e_i^T P_i^{-1} e_i \tag{4.20}$$

where M is a number of Monte-Carlo runs, N is a dimension of the state, and e, P are an estimation error and its predicted covariance matrix, respectively. Note that the ANEES should be 1 in a consistent estimator.

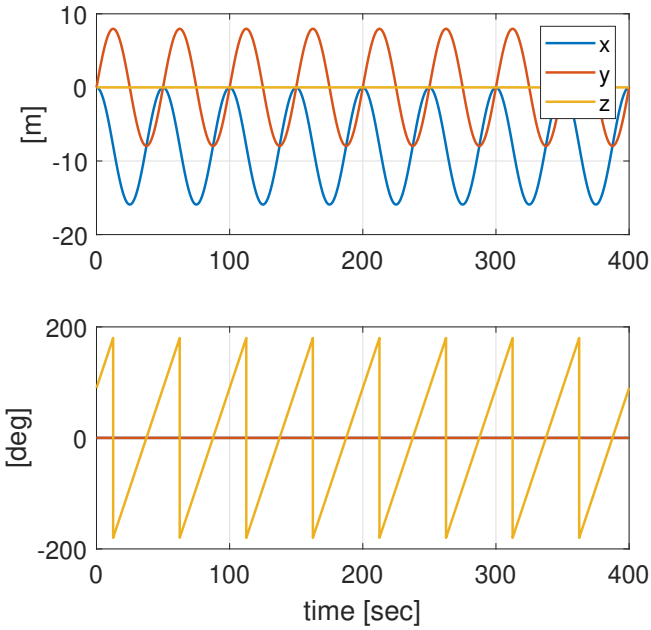
At the first setting with a small attitude uncertainty, 0.001 deg of a vehicle, *Proposed filter* outputs consistent estimates giving ANEES close to 1 as shown in 4.2b. This improves the localization accuracy as shown in 4.2a. In contrast, $SO(3)$-*EKF* gains spurious information along the rotation about the gravity direction as shown in 4.2b, and this leads to the degradation on the

localization accuracy. This confirms the theoretical result in (4.19) that *Proposed filter* obeys the true observability property, but $SO(3)$-*EKF* suffers from the underestimation.

At the second setting with the increasing initial attitude uncertainty $\sigma_{R_b} = \{0.001,\ 1,\ 2,\ 3,\ 4,\ 5\}$ deg, an estimated Gaussian distribution of $SO(3)$-*EKF* almost fails to capture the true density due to the large initial uncertainty showing worse accuracy and consistency as shown in Fig. 4.2c and Fig. 4.2d. However, *Proposed filter* is robust to the initial uncertainty since it *perfectly* represents the nonlinear system as a linear system in a deterministic sense. It is originated from the fact that the $SE_2(3)$ structure in (4.5) is the proper group representation and the higher order terms (4.13) are identically zero at the object initialization. In terms of computational complexity, *Proposed filter* only adds additional computing cost for the matrix exponential on $SE_2(3)$ and logarithm on $SE(3)$ compared to $SO(3)$-*EKF*. These are effectively computed using a closed-form expression.



(a)



(b)

Figure 4.1: (a) Virtual circular trajectory with 12 objects, (b) vehicle's true position and attitude profile.

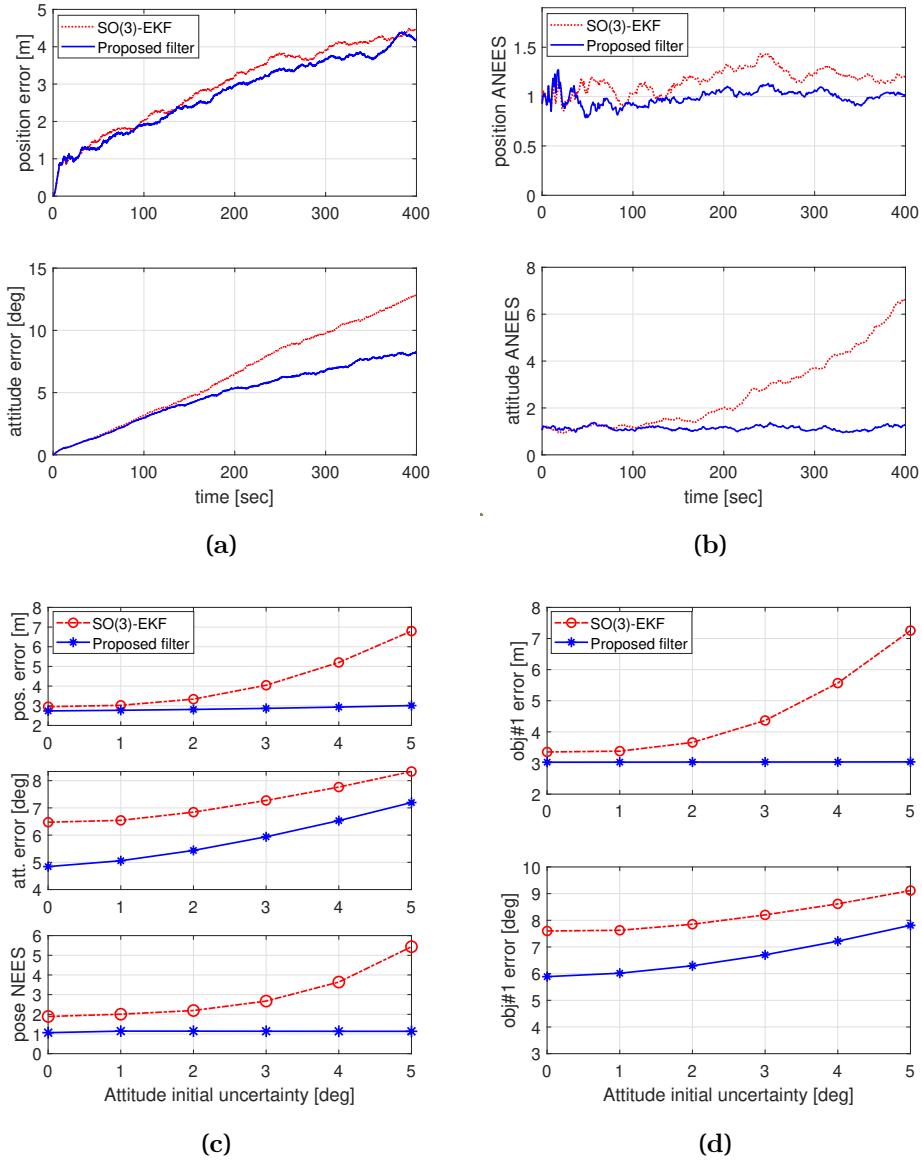


Figure 4.2: (a) Pose root mean square error (RMSE) and (b) averaged normalized estimation error squared (ANEES) in the 50 Monte-Carlo runs. The proposed method outputs accurate and consistent estimates giving ANEES near 1. (c) Pose RMSE and (d) object mapping accuracy with the increasing attitude initial uncertainty $\sigma_{R_b} = \{0.001,\ 1,\ 2,\ 3,\ 4,\ 5\}$ deg. The proposed method is robust to the initial attitude error in terms of localization and mapping accuracy and consistency.

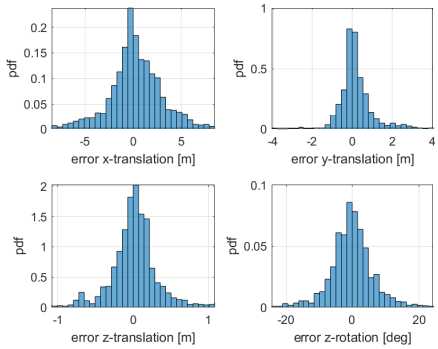
4.3.2 Driving Datasets

The KITTI dataset [96] is chosen to demonstrate the feasibility of the proposed method since the dataset provides raw IMU measurements as well as the ground-truth vehicle poses and 3D bounding boxes for quantitative mapping performance. As a 3D object detector, the Mousavian *et al.*'s method [97] with YOLOv3 [98] is adopted. The network was trained on the KITTI 3D object benchmark and outputs relative position, yaw angle in $\{c\}$, and cuboid lengths of objects. Snapshots of well and bad-fitted cuboids are shown in Fig. 4.3a. All these local 3D object measurements are fused in a fully probabilistic fashion to obtain better localization and global mapping result. Note that any 3D detectors can be employed if it provides a relative pose measurement.

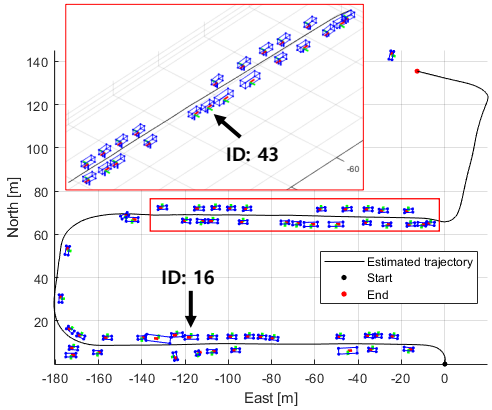
Most research on 3D detectors reports the performance as a 3D intersection over union (IoU). However, in terms of a filter measurement, a pose measurement error before filtering on Lie algebra $\xi_j = \ln(\hat{T}_j T_j^{-1})^\vee$ should be analyzed. Therefore, a noise distribution of the pose measurements is investigated. Note that incoming object measurements are matched to the closest ground-truth cuboid to obtain errors. Error histograms exhibit Gaussian-like densities matching the assumption in Fig. 4.3b. The position error along the depth (the x-axis of $\{b\}$) shows the largest standard deviation as 2.82 m due to a limitation of monocular vision, while the relative yaw reports the deviation as 8.09 deg. These values are set as a measurement uncertainty in the Monte-Carlo simulation in 4.3.1 and in Table 4.1 for the rest of the sequences.

The objective of this experiment is to show that 1) the fusion of inertial and deep object measurement increase not only the localization but also the objects' global mapping accuracy, 2) the proposed method shows comparable localization accuracy to the state-of-the-art methods [59, 99] using only object

(a)



(b)



(c)

Figure 4.3: The representative result on KITTI *2011_09_26_0022* sequence with (a) well and bad-fitted cuboid measurement by the Mousavian's method from which the proposed method is updated, (b) noise statistics of relative pose measurements, and (c) qualitative localization and mapping results of the proposed method.

measurements and the nonholonomic constraint. For the former, the localization accuracy by the absolute trajectory error (ATE) and the mapping accuracy by objects' global pose error when marginalized out (OBJ RMSE) are reported. For the latter, the relative pose error (RPE) is evaluated using the evaluation toolkit provided in [96].

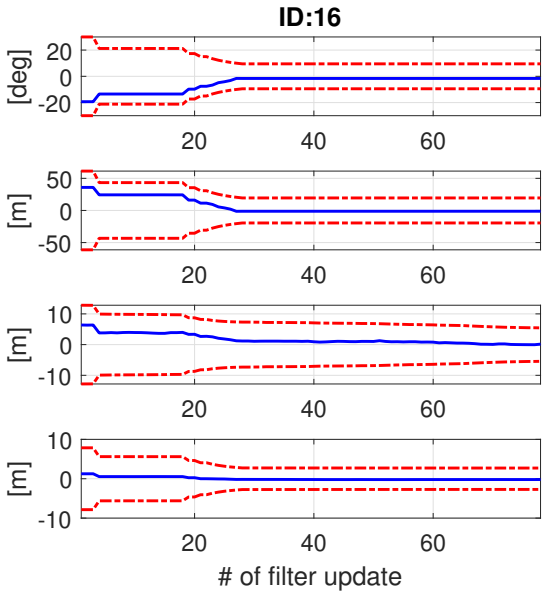
Table 4.1 reports the evaluation result, and the followings are the four key interpretations. First, while the nonholonomic constraint (*NH*) reduces much of a position error in an inertial navigation system (*INS*), *Proposed filter* further decreases the position error by fusing object and inertial measurements. Second, the rotational error of *Proposed filter* is slightly worse than *INS* in the order of millidegree. This indicates that the visual measurement is not as precise as the gyroscope, which has a 0.01 deg/s 1σ bias. Third, *Proposed filter* reduces object mapping error from 7.31 to 5.51 m and from 13.93 to 9.07 deg by probabilistically accounting for measurement noises at multiple views. Lastly, RPE of the proposed approach is comparable to the state-of-the-art methods. Although it is assumed that the data association is solved using the closest reference cuboid, it is worth mentioning that the proposed method only utilizes object measurements and the nonholonomic constraint with a forward speed, while other methods additionally include corners [59] or semantic keypoints [99].

Fig. 4.3c illustrates the estimated localization and object mapping results where parked cars are well-aligned to the vehicle trajectory qualitatively. An estimation history of the highlighted objects is shown in Fig. 4.4 with their actual errors and predicted standard deviations. It is remarkable to note that objects are well-converged from their large initial uncertainties (4.12).

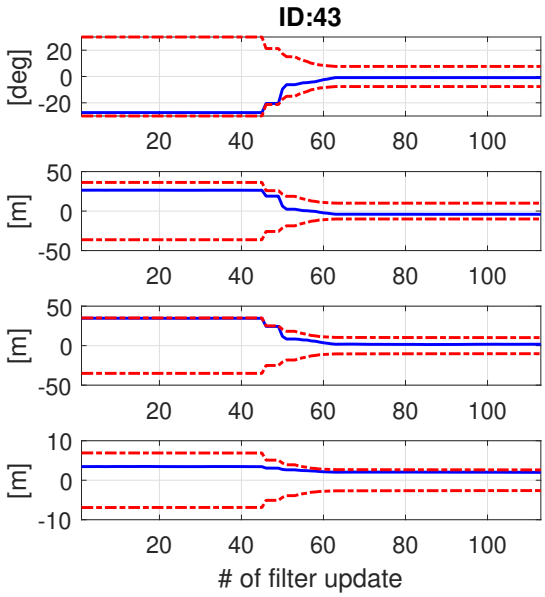
Table 4.1: Localization accuracy (ATE, RPE) and object mapping accuracy (OBJ RMSE) on KITTI 2011_09_26_00XX raw sequences

Metric	Method	0022	0023	0036	0039	0061	0064	0095	0096	Mean
ATE [%]	*INS*	4.58	3.52	8.31	**0.73**	9.19	2.70	3.61	3.00	4.46
	INS+NH	1.00	0.72	1.79	0.89	5.97	0.90	1.65	1.53	1.81
	Proposed[1)]	**0.98**	**0.38**	**1.73**	0.86	**2.10**	**0.72**	**0.56**	**1.13**	**1.06**
ATE [deg/km]	*INS*	1.36	**1.84**	0.81	3.86	**1.69**	**2.88**	2.43	1.47	**2.04**
	INS + NH	**1.32**	**1.84**	**0.79**	3.86	2.34	2.93	2.40	**1.36**	2.11
	Proposed	1.34	1.86	**0.79**	**3.82**	2.30	2.96	**2.39**	**1.36**	2.10
OBJ RMSE[2)] $[m]$	*INS+NH*	5.14	3.76	7.38	4.15	21.45	6.85	**3.47**	6.27	7.31
	Proposed	**5.01**	**3.56**	**6.83**	**3.51**	**9.41**	**6.48**	3.58	**5.67**	**5.51**
OBJ RMSE[2)] $[deg]$	*INS+NH*	10.89	22.86	17.25	10.54	11.22	14.91	9.66	14.08	13.93
	Proposed	**6.32**	**16.85**	**9.70**	**6.43**	**7.01**	**11.18**	**5.34**	**9.72**	**9.07**
RPE [%]	CubeSLAM [59]	1.68	1.72	2.93	1.61	**1.24**	**0.93**	1.49	1.81	1.68
	OrcVIO [99]	**1.64**	2.51	2.11	**1.03**	3.11	2.48	**1.05**	1.40	1.92
	Proposed+SPD[3)]	1.69	**1.19**	**1.07**	1.35	1.62	1.00	1.72	**1.14**	**1.35**

1) *INS* + nonholonomic (*NH*) + deep object measurements
2) Objects' global pose RMSE when marginalized out from the filter state
3) Forward speed updates the filter as well as a nonholonomic constraint as in OrcVIO



(a)



(b)

Figure 4.4: Selected object mapping errors (blue) in the KITTI *0022* sequence versus a number of filter update with their $\pm 3\,\sigma$ confidence (red). Note that the corresponding objects are drawn in Fig. 4.3c.

4.4 Conclusion

In this chapter, a consistent and robust object-based VINS has been proposed that is closely related to the invariant Kalman filtering. The state-space was fully embedded in the matrix Lie group to formulate the 3D object observation. The proposed approach solves the filter inconsistency and is very robust to the initial state uncertainty. Furthermore, it is demonstrated that the fusion of visual and inertial measurement can improve localization and mapping accuracy. In future research, object size information and low-level features will be incorporated to decrease the estimation error further. The object data association will be solved using geometric and appearance-based constraints and the static object assumption will be also relieved.

Chapter 5

Object SLAM with Pose Ambiguity

This chapter contains the contents of the following journal publication:

J. H. Jung and C. G. Park, "**Gaussian Mixture Midway-Merge for Object SLAM With Pose Ambiguity,**"
IEEE Robotics and Automation Letters, vol. 8, no. 1, pp. 400–407, Jan. 2023, doi: 10.1109/LRA.2022.3224665.

This chapter proposes a novel method to merge a Gaussian mixture on matrix Lie groups and present its application for a simultaneous localization and mapping problem with symmetric objects. The key idea is to predetermine the weighted mean called a *midway point* and merge Gaussian mixture components at the associated tangent space. Through this rule, the covariance matrix captures the original density more accurately, and the need for the back-projection is spared when compared to the conventional merge. The strength of the midway-merge is highlighted by numerically evaluating dissimilarity metrics of density functions before and after the merge on the rotational group. Furthermore, it is experimentally discovered that the rotational error of symmetric objects follows heavy-tailed behavior. Then, the Gaussian sum filter is formulated to model it by a Gaussian mixture noise. The effectiveness of the proposed approach is validated through virtual and real-world datasets.

5.1 Introduction

A Gaussian mixture (GM) is prevalent in engineering problems such as state estimation and target tracking due to its ability to model multiple hypotheses. A radar altimeter exhibits a GM noise characteristic in vegetated areas by multiple returns from the ground and vegetation canopy [100, 101]. The posterior intensity is represented by a GM in the probability hypothesis density filter [102] for the efficient update recursion. Especially in robotics research, GM is a versatile tool to perceive the surrounding real-world environment. In simultaneous localization and mapping (SLAM) problems, multiple loop closure and data association hypotheses can be encoded in GM distributions [65, 103]. It is common to model the map as a mixture of local Gaussian distributions for better registration [104, 105]. Multiple pose hypotheses of symmetric objects are modeled in a GM distribution [64, 66].

The Bayes rule plays a central role in estimation problems, and the Bayesian filter is solved analytically in a linear Gaussian system. However, once a GM is introduced in either process or measurement distributions, the Bayesian recursion exponentially increases the number of hypotheses [72], [106]. To maintain a tractable size of a state dimension, Gaussian components should be merged or pruned with the minimum information loss. On the other hand, in the aspect of modeling a state-space representation, a matrix Lie group is a natural tool to deal with the underlying geometry, for instance, describing the three-dimensional position and attitude of a rigid body [11, 78]. Filtering on matrix Lie groups [12, 107] has shown promising consistency and accuracy over the conventional parameterization in a vector space.

However, dealing with a GM on matrix Lie groups is not straightforward. A question arises in how to define GMs and a reduction procedure on matrix

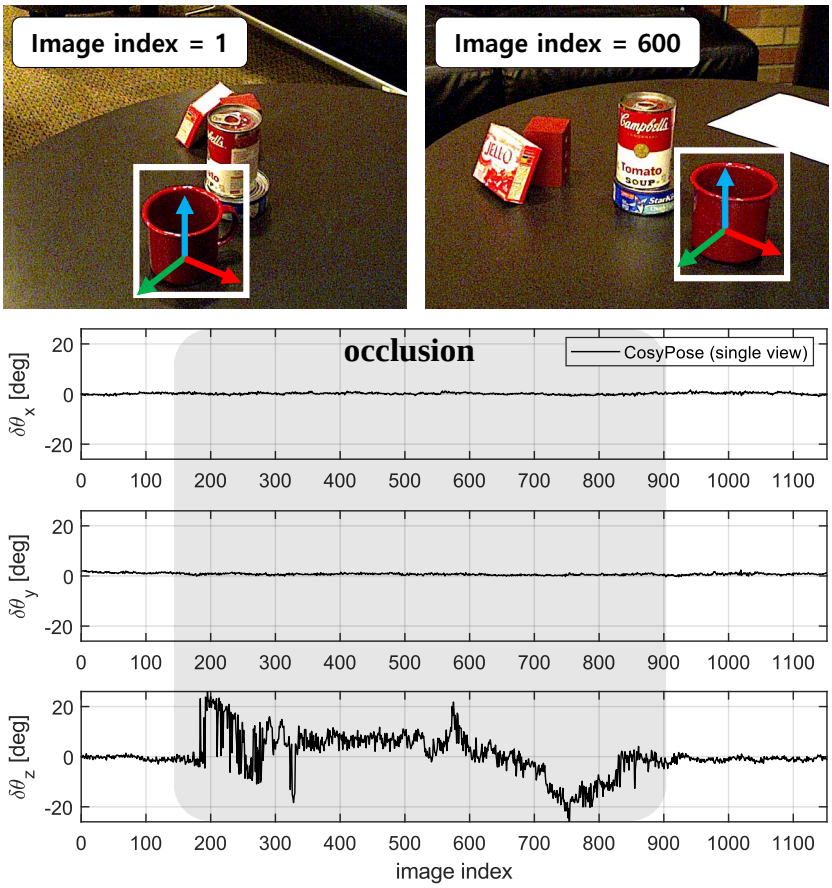


Figure 5.1: Rotational error estimated by a 6 DOF pose detector (CosyPose) of a mug in the YCB-Video dataset. The symmetric z-axis exhibits heavy-tailed noise distribution due to self-occlusion.

Lie groups. To this end, this study proposes to merge a GM at a common tangent space that is called a *midway point*, a weighted matrix of mean matrices. Through this merge rule, it is proved that the approximation error is less than the conventional merge by Ćesić *et al.* [69] when deriving the merged covariance matrix. Furthermore, the proposed approach eliminates the need for the back-projection leading to a lighter computational burden over the conventional merge.

A promising application example in object SLAM with pose ambiguity is presented. Since a pioneering work of object-based SLAM [10], these systems have shown promising results over conventional low-level SLAM by capturing semantically as well as geometrically meaningful information. However, estimating the 6 degrees of freedom (DOF) pose of a symmetric object with occlusion is still very challenging. For instance, it is experimentally discovered that the rotational error along the symmetric axis of a mug behaves as a heavy-tailed distribution, as shown in Fig. 5.1 and Fig. 5.4. A standard Gaussian distribution cannot capture this behavior and properly weight these outliers. To tackle this, the noise distribution is fitted by a GM and a Kalman filter with the proposed GM merge method is formulated using a 6 DOF pose detector as a sensor. The main contribution of this chapter is as follows.

- A novel GM merge on matrix Lie groups called *Gaussian mixture midway-merge* is proposed, where probability density functions (PDF) are transformed at the tangent space of the predetermined midway point, then merged.

- The strength of the merge method is validated through the numerical evaluation of PDF dissimilarity metrics on the special orthogonal group $SO(3)$.

- Kalman filter with the proposed merge is formulated for object SLAM with pose ambiguity and experimental tests demonstrate its effectiveness in a Monte-Carlo simulation, photo-realistic simulator, and real-world dataset. The implementation[1] is open-sourced for the benefit of the community.

5.2 Gaussian Mixture Merge

This section reviews the uncertainty transformation in [69] and reveals a limitation that would yield information loss with increasing distance between means of GM components. Then, a *Gaussian mixture midway-merge* method is introduced to mitigate this.

5.2.1 Uncertainty at Transformed Mean

To merge a GM on matrix Lie groups, it is straightforward to merge them at the same mean. Assume that it is required to express $N_G(\hat{X}_i, P_i)$ at some common point $\hat{X}_c$. Using notations in Section 2.2, the ith random vector is $\xi_i = \ln(\hat{X}_i X^{-1})^\vee$. Substituting $X = \exp(-\xi_c^\wedge)\hat{X}_c$,

$$\begin{aligned}\xi_i &= \ln\left(\hat{X}_i \hat{X}_c^{-1} \exp(\xi_c^\wedge)\right)^\vee = \ln\left(\exp(\Delta x_i^\wedge)\exp(\xi_c^\wedge)\right)^\vee \\ &\overset{(2.34)}{\approx} J_l(\Delta x_i)(\xi_c + \Delta x_i) \end{aligned} \tag{5.1}$$

if Δx_i is small where $\exp(\Delta x_i^\wedge) := \hat{X}_i \hat{X}_c^{-1}$. Then, substituting (5.1) to the first line of (2.38), the below is obtained

$$\begin{aligned} 1 \approx \int_{\mathbb{R}^N} &(2\pi)^{-\frac{N}{2}} \left|\bar{P}_i\right|^{-\frac{1}{2}} \times \\ &\exp\left(-\frac{1}{2}(\xi_c + \Delta x_i)^T \bar{P}_i^{-1}(\xi_c + \Delta x_i)\right) d\,\xi_c \end{aligned} \tag{5.2}$$

[1] https://github.com/lastflowers/midway

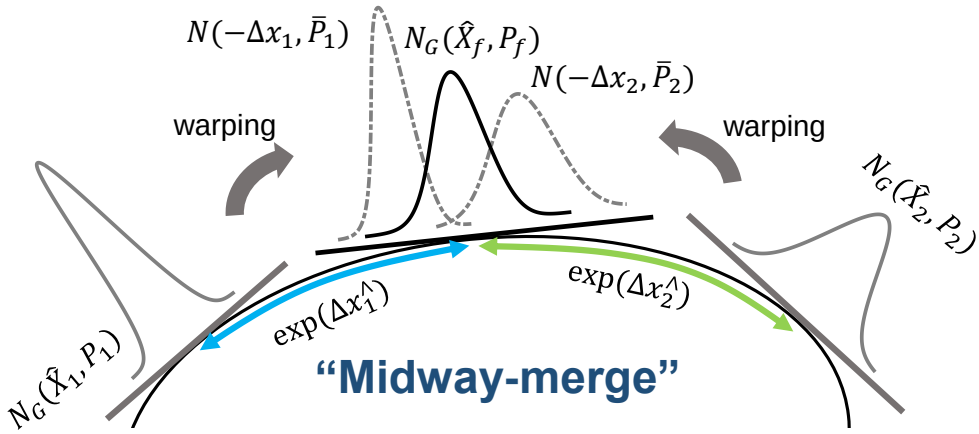


Figure 5.2: Schematic illustration of the proposed merge with the corresponding densities at each step.

where $d\,\xi_i = |J_l(\Delta x_i)| d\,\xi_c$ and the new covariance is $\bar{P}_i = J_l^{-1}(\Delta x_i) P_i J_l^{-T}(\Delta x_i)$. Therefore, the random vector at the new mean follows, $\xi_c \sim N(-\Delta x_i, \bar{P}_i)$ [69].

It is clear that as the distance between mean matrices $\|\Delta x_i\|$ increases, (5.2) is no longer a valid PDF, hence $\bar{P}_i$ cannot capture the correct uncertainty. Also, being a valid density in the line of (2.38) requires back-projection to make a zero-mean. To mitigate the assumption and remove the additional computation, GM components are merged at a *midway point* in the following section.

5.2.2 Midway-Merge

Given two components in a GM on matrix Lie groups,

$$w_1 N_G(\hat{X}_1, P_1) + w_2 N_G(\hat{X}_2, P_2) \tag{5.3}$$

the key idea is to merge them at the fused mean,

$$\hat{X}_f = \left(\hat{X}_2 \hat{X}_1^{-1}\right)^{w_2^*} \hat{X}_1 \tag{5.4}$$

where $w_2^* = w_2/(w_1 + w_2)$ is a normalized weight. (5.4) is analogous to linear interpolation in a Lie algebra [11]. $\hat{X}_f$ is termed as a *midway point* as it is placed between $\hat{X}_1$ and $\hat{X}_2$ as schematically seen from Fig. 5.2. If $N_G(\hat{X}_1, P_1)$ is expressed at the midway point using (5.2),

$$\begin{aligned} \xi_f &\sim N\left(-\Delta x_1, \bar{P}_1\right) \\ \Delta x_1 &= \ln\left(\hat{X}_1 \hat{X}_f^{-1}\right)^\vee \\ \bar{P}_1 &= J_l^{-1}(\Delta x_1) P_1 J_l^{-T}(\Delta x_1) \end{aligned} \tag{5.5}$$

where it is assumed that $O(\|\Delta x_1\|^2) = 0$. Likewise, the second component is transformed to the midway point,

$$\begin{aligned} \xi_f &\sim N\left(-\Delta x_2, \bar{P}_2\right) \\ \Delta x_2 &= \ln\left(\hat{X}_2 \hat{X}_f^{-1}\right)^\vee \\ \bar{P}_2 &= J_l^{-1}(\Delta x_2) P_2 J_l^{-T}(\Delta x_2). \end{aligned} \tag{5.6}$$

Again, it is assumed that $O(\|\Delta x_2\|^2) = 0$. The transformed distributions in (5.5) and (5.6) are merged at the tangent space on $\hat{X}_f$ by preserving moments as introduced in Section 2.6,

$$\begin{aligned} \Delta x_f &= -w_1^* \Delta x_1 - w_2^* \Delta x_2 \\ &= -w_1^* \ln\left(\hat{X}_1 \hat{X}_f^{-1}\right)^\vee - w_2^* \ln\left(\hat{X}_2 \hat{X}_f^{-1}\right)^\vee. \end{aligned} \tag{5.7}$$

The distance between $\hat{X}_1$ and $\hat{X}_2$ are defined as

$$\exp(\Delta x^\wedge) := \hat{X}_2 \hat{X}_1^{-1}. \tag{5.8}$$

Substituting (5.4) and (5.8) to (5.7) yields

$$\begin{aligned}\Delta x_f &= -w_1^* \ln\left(\exp\left(-w_2^* \Delta x^\wedge\right)\right)^\vee - w_2^* \ln\left(\exp\left(\Delta x^\wedge\right) \exp\left(-w_2^* \Delta x^\wedge\right)\right)^\vee \\ &= w_1^* w_2^* \Delta x - w_2^*(1 - w_2^*)\Delta x = 0. \end{aligned} \tag{5.9}$$

This implies that the merged distribution at the tangent space of $\hat{X}_f$ is a valid distribution as a line of (2.38). Using (2.41), the merged covariance is

$$\begin{aligned}P_f &= w_1^* \bar{P}_1 + w_2^* \bar{P}_2 + w_1^* w_2^* (\Delta x_2 - \Delta x_1)(\Delta x_2 - \Delta x_1)^T \\ &= w_1^* \bar{P}_1 + w_2^* \bar{P}_2 + w_1^* w_2^* \Delta x \Delta x^T, \end{aligned} \tag{5.10}$$

and the weight is

$$w_f = w_1 + w_2. \tag{5.11}$$

Fig. 5.2 illustrates the overall procedure of the *Gaussian mixture midway-merge* in which each component is transformed to the same mean and merged in a vector space. It is remarkable to note that the proposed approach does not require any back-projection to matrix Lie groups by virtue of (5.9) that was needed in [69]. This spares computing the adjoint when merging the covariance.

5.2.3 Approximated Error Analysis

It has been assumed that both $\|\Delta x_1\|^2$ and $\|\Delta x_2\|^2$ are zero when deriving the merge method. This approximation is actually less significant than the assumption ($\|\Delta x\|^2 = 0$) in the previous method [69].

Theorem 1. *Given* $\|\Delta x_1\|$, $\|\Delta x_2\|$, *and* $\|\Delta x\|$ *in* (*5.5*), (*5.6*), *and* (*5.8*), *respectively, then,* $\|\Delta x_1\|^2 + \|\Delta x_2\|^2 \leq \|\Delta x\|^2$

Proof. From the definition it can be seen that the distance between $\hat{X}_1$ and $\hat{X}_f$ is

$$\Delta x_1 = \ln\left(\exp(-w_2^* \Delta x^\wedge)\right)^\vee = -w_2^* \Delta x. \tag{5.12}$$

Likewise, the distance between $\hat{X}_2$ and $\hat{X}_f$ is

$$\Delta x_2 = \ln\left(\exp((1-w_2^*)\Delta x^\wedge)\right)^\vee = (1-w_2^*)\,\Delta x. \tag{5.13}$$

Therefore,

$$\begin{aligned}\|\Delta x_1\|^2 + \|\Delta x_2\|^2 &= (w_1^{*2} + w_2^{*2})\|\Delta x\|^2 \\ &\leq \|\Delta x\|^2\end{aligned} \tag{5.14}$$

since $0 \leq w_1^* \leq 1$, $0 \leq w_2^* \leq 1$ and $w_1^* + w_2^* = 1$. ■

It is remarkable to note that the approximation error in the Gaussian mixture midway-merge is always less than the conventional method [69] except when $w_1^* = 1$ or $w_2^* = 1$. In these extreme cases, a merge is not required. Based on (5.14), the proposed approach can estimate the merged covariance matrix more accurately when merging Gaussian distributions on matrix Lie groups.

5.3 Gaussian Merge on SO(3)

The objective of this test is to investigate dissimilarity between densities before $p(X)$ and after the merge $q(X)$ with increasing distance of mean matrices. Suppose that densities are given on $SO(3)$ such that

$$\begin{aligned}p(X) = &\underbrace{0.5}_{w_1^*} N_G(\underbrace{\exp(\begin{bmatrix}0,0,0\end{bmatrix}^{T^\wedge})}_{\hat{X}_1}, \underbrace{5^2 I_3}_{P_1}) \\ &+ \underbrace{0.5}_{w_2^*} N_G(\underbrace{\exp(\begin{bmatrix}\phi,-\phi,\phi\end{bmatrix}^{T^\wedge})}_{\hat{X}_2}, \underbrace{10^2 I_3}_{P_2})\end{aligned} \tag{5.15}$$

$$q(X) = N_G\left(\hat{X}_f, P_f\right) \tag{5.16}$$

where the angle, ϕ is increased from 5 to 60 deg with the interval of 5 deg. The KLD and ISD are defined as follows

$$
\begin{aligned}
D_{KL}(p \,||\, q_\star) &= \int_{SO(3)} p(X) \ln\left(\frac{p(X)}{q_\star(X)}\right) dX \\
ISD(p,\, q_\star) &= \int_{SO(3)} \left(p(X) - q_\star(X)\right)^2 dX
\end{aligned} \tag{5.17}
$$

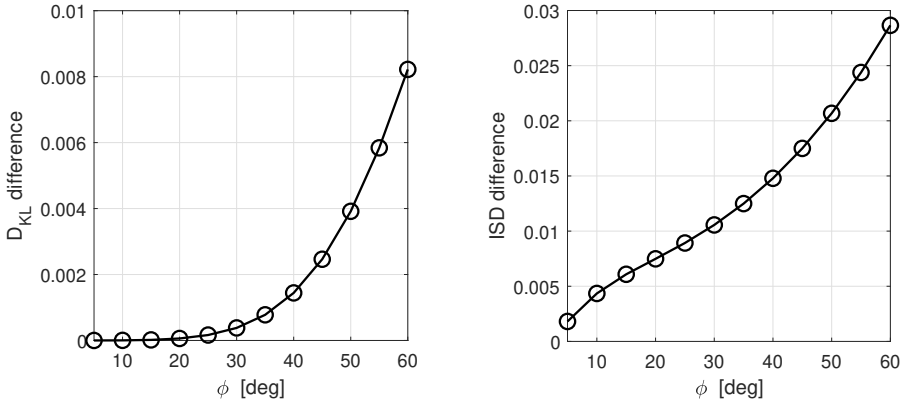
where $p(X)$ is defined in (5.15), and $q_\star(X)$ either could be the proposed merge $q_1(X)$ or the previous method $q_2(X)$ [69].

Since a direct numerical integration on $SO(3)$ is intractable, the KLD and ISD are numerically integrated on $\mathfrak{so}(3)$ with a 0.3 deg interval using the BCH formula up to the 5th order terms. By the definition of KLD,

$$
\begin{aligned}
&D_{KL}(p \,||\, q) = \\
&\int_{\mathfrak{so}(3)} \left(\eta_1 \exp\left(-\frac{1}{2}\xi^T P_1^{-1}\xi\right) + \eta_2 \exp\left(-\frac{1}{2}\xi_2^T P_2^{-1}\xi_2\right)\right) \\
&\times \ln\left(\frac{\eta_1 \exp\left(-\frac{1}{2}\xi^T P_1^{-1}\xi\right) + \eta_2 \exp\left(-\frac{1}{2}\xi_2^T P_2^{-1}\xi_2\right)}{\eta_f \exp\left(-\frac{1}{2}\xi_f^T P_f^{-1}\xi_f\right)}\right) d\xi
\end{aligned} \tag{5.18}
$$

where ξ, ξ_2, and ξ_f are random vectors on $\hat{X}_1$, $\hat{X}_2$, and $\hat{X}_f$, respectively. The normalizers are

$$
\begin{aligned}
\eta_1 &= w_1^*(2\pi)^{-\frac{3}{2}}|P_1|^{-\frac{1}{2}}, \\
\eta_2 &= w_2^*(2\pi)^{-\frac{3}{2}}|P_2|^{-\frac{1}{2}}\frac{|J_l(\xi)|}{|J_l(\xi_2)|}, \\
\eta_f &= (2\pi)^{-\frac{3}{2}}|P_f|^{-\frac{1}{2}}\frac{|J_l(\xi)|}{|J_l(\xi_f)|}.
\end{aligned} \tag{5.19}
$$



(a) **(b)**

Figure 5.3: The differences of the (a) Kullback-Leibler divergence and (b) integral square distance between the proposed and the Ćesić's method.

Using the BCH formula ξ_2 is approximated as follows

$$\begin{aligned}\xi_2 &= \ln\left(\hat{X}_2 X^{-1}\right)^{\vee} = \ln\left(\hat{X}_2 \hat{X}_1^{-1} \hat{X}_1 X^{-1}\right)^{\vee} \\ &= \ln\left(\exp(\Delta x^{\wedge})\exp(\xi^{\wedge})\right)^{\vee} \\ &\approx J_l(\xi)^{-1}\Delta x + \xi + \frac{1}{12}\Delta x^{\wedge}\Delta x^{\wedge}\xi - \frac{1}{24}\xi^{\wedge}\Delta x^{\wedge}\Delta x^{\wedge}\xi \\ &+ \frac{1}{120}\Delta x^{\wedge}\xi^{\wedge}\Delta x^{\wedge}\xi^{\wedge}\Delta x + \frac{1}{120}\xi^{\wedge}\Delta x^{\wedge}\xi^{\wedge}\Delta x^{\wedge}\xi. \end{aligned} \tag{5.20}$$

Likewise, ξ_f is approximated as analogous to ξ_2. (5.18) is solved by the Euler method with $\Delta\xi_i = 0.3\,\mathrm{deg}$ interval,

$$\sum_{i=1}^{1200^3} p(\xi_i)\ln\left(\frac{p(\xi_i)}{q(\xi_i)}\right)\Delta\xi_i. \tag{5.21}$$

Fig. 5.3 shows the differences of each dissimilarity measure: $D_{KL}(p\,||\,q_2) - D_{KL}(p\,||\,q_1)$ and $ISD(p,\,q_2) - ISD(p,\,q_1)$. Since the differences are larger than zero at all ϕ, the proposed method more accurately captures the original density $p(X)$ than the conventional method. As expected from the error analysis in

Section 5.2.3, the differences become larger when ϕ increases. This is originated from the violation of the assumption that $\|\Delta x\|^2 \approx 0$.

5.4 Object SLAM formulation

A SLAM problem with a symmetric object is considered as a promising example of the midway-merge. Specifically, this chapter focuses on jointly estimating robot and object poses parameterized by

$$X = (T_R, T_1, \ldots, T_M) \tag{5.22}$$

where T_R represents a robot pose and $\{T_j\}_{j=1}^{M}$ are object poses in $SE(3)$. The robot pose is driven by the odometry model,

$$\frac{d}{dt} T_R(t) = T_R(t)\,(u(t) + n_w(t))^\wedge \tag{5.23}$$

where $u(t) \in \mathbb{R}^6$ is the true body velocity, and $n_w(t) \in \mathbb{R}^6$ is a zero-mean white Gaussian noise with $E[n_w(t)\, n_w(\tau)^T] = Q\,\delta(t-\tau)$. It is assumed that objects are static $d/dt\, T_j = 0$. An onboard sensor (RGB sensor with a deep neural network) measures a relative pose of the jth object at a discrete time,

$$Y_j(t_k) = T_R(t_k)^{-1}\, T_j(t_k)\, \exp(n_j(t_k)^\wedge). \tag{5.24}$$

The measurement noise is a white GM that is uncorrelated to n_w such that

$$n_j(t_k) \sim \alpha_1\, N(0, R_1) + \alpha_2\, N(0, R_2). \tag{5.25}$$

Around 10k images of a mug in the YCB-Video dataset [108] are investigated when its handle is self-occluded. The rotational error histogram by CosyPose [109] with a single view is shown in Fig. 5.4. It is clearly seen that the symmetric z-axis exhibits heavy-tailed behavior. This is encoded by two GM components

in n_j for a simple but effective representation.

The Gaussian sum filter is implemented where each hypothesis is an extended Kalman filter parameterized by the right-invariant error based on the object-based SLAM which is developed in Chapter 4. The weights of each hypothesis are recursively updated according to the Bayes rule [72]. By introducing the multimodal noise, it is evident that the number of hypotheses exponentially increases. Unless otherwise noted, two Gaussian components are sequentially merged, where they have the largest and the smallest weight, until the number of hypotheses reaches the predefined threshold N_h. Also, the largest and second-largest weights of Gaussian components are merged to summarize the estimated quantity. Algorithm 2 shows the Gaussian mixture invariant extended Kalman filter (*GM-IEKF*) with the midway-merge. It is implemented in MATLAB with Intel i5-7600 CPU at 3.50 GHz.

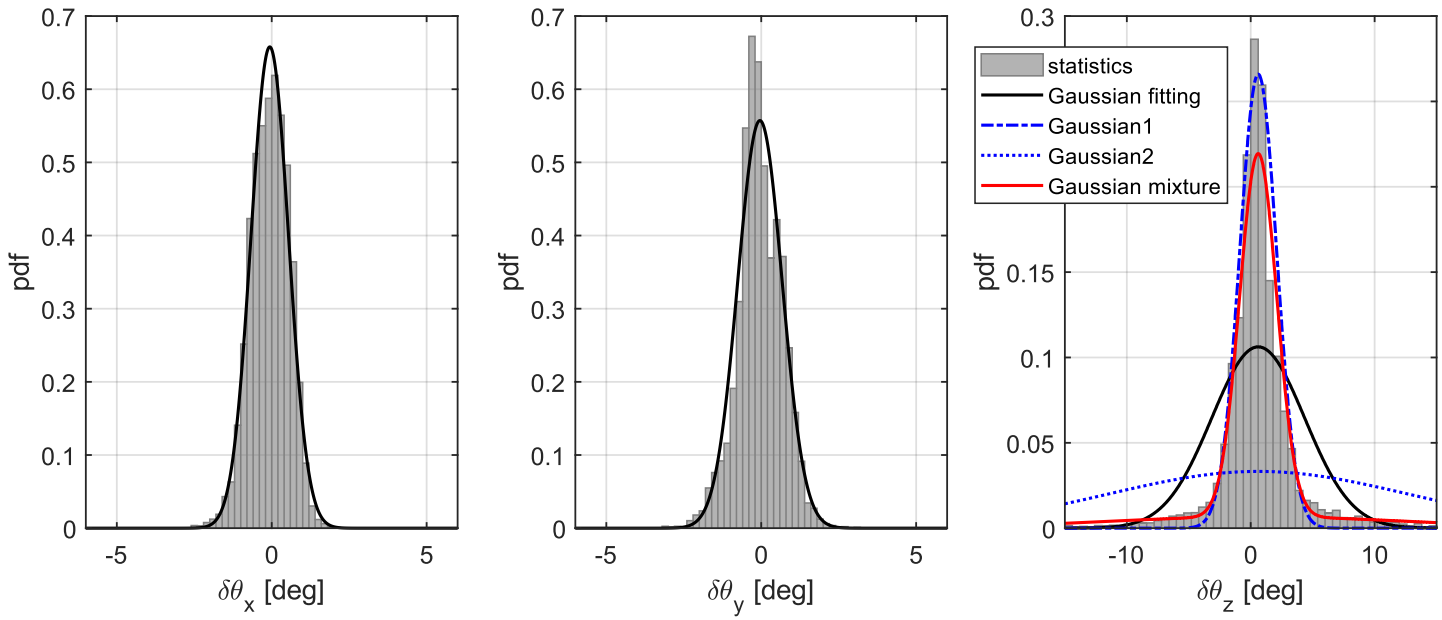


Figure 5.4: Rotational error histogram of the self-occluded mug estimated by CosyPose in the YCB-Video dataset. The Gaussian mixture captures the heavy-tailed density, while the Gaussian fitting fails to account for it.

Algorithm 2 GM-IEKF with midway-merge

Input: $\{w^{(h)},\ X^{(h)},\ P^{(h)}\}_{h=1}^{N_h^-},\ \{Y_j\}_{j=1}^{M},\ u_m$

Output: $\{w^{(h)},\ X^{(h)},\ P^{(h)}\}_{h=1}^{N_h}$

for $h = 1$ to N_h^- **do** // prediction for every hypothesis

  $(X^{(h)},\ P^{(h)}) \leftarrow$ `Prediction` $(X^{(h)},\ P^{(h)},\ u_m)$

end for

for $h = 1$ to N_h^- **do** // update for every hypothesis

  for $i = 1$ to 2 **do**

    $w^{(h+)} \leftarrow$ `weightUpdate` $(w^{(h)},\ X^{(h)},\ P^{(h)},\ R_i)$

    $(X^{(h+)},\ P^{(h+)}) \leftarrow$ `Update` $(X^{(h)},\ P^{(h)},\ \{Y_j\}_{j=1}^{M},\ R_i)$

  end for

end for

$N_h^+ = 2N_h^-$

while $N_h^+ \geq N_h$ **do**

  `midwayMerge` $(w^{(h)},\ X^{(h)},\ P^{(h)}) \cdots$ from (5.4), (5.10), (5.11)

end while

5.5 Experiments

Throughout this section, GM-IEKF with the proposed midway-merge is designated as GM(*midway*), while GM-IEKF with the conventional merge anchored at the tangent space of the larger weight [69] is designated as GM($\mathcal{T}_L$). Pose error is defined as

$$\epsilon = \ln\left(\hat{T}\,T^{-1}\right)^{\vee} \tag{5.26}$$

by the convention in (2.37). In the simulation part, the averaged normalized estimation error squared (ANEES) measures estimator consistency. At a specific time, ANEES is defined as

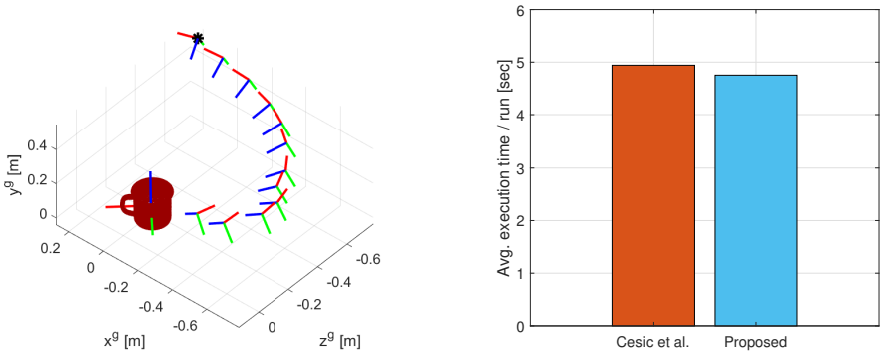
$$\text{ANEES} = \frac{1}{N\,N_s}\sum_{i=1}^{N_s}\epsilon_i^T P_i^{-1}\epsilon_i \tag{5.27}$$

where $N = 6 + 6M$ is a dimension of the state, $N_s = 100$ is the number of simulative runs, and P is the error covariance. Therefore, ANEES should be the unity if the covariance fully explains the actual error.

5.5.1 Monte-Carlo Simulation

A virtual trajectory shown in Fig. 5.5a, odometry readings, and pose detection are generated based on a sensor specification. The odometry noises are set as $0.026\,\text{deg}/s/\sqrt{Hz}$ and $0.002\,m/s/\sqrt{Hz}$, respectively. A virtual detector outputs a 6 DOF pose of the mug where only the rotation z-axis gives the GM noise, that is

$$\begin{aligned}
\alpha_1 &= 0.7,\\
\sqrt{R_1} &= \text{diag}(4^\circ,\ 4^\circ,\ 4^\circ,\ 0.01\text{m},\ 0.01\text{m},\ 0.01\text{m}),\\
\alpha_2 &= 0.3,\\
\sqrt{R_2} &= \text{diag}(4^\circ,\ 4^\circ,\ 12^\circ,\ 0.01\text{m},\ 0.01\text{m},\ 0.01\text{m}).
\end{aligned} \tag{5.28}$$



(a) **(b)**

Figure 5.5: (a) Virtual trajectory with a mug where the asterisk marks the first camera pose. (b) Average execution time of a single run in 100 Monte-Carlo simulations.

To expose merge methods in an extreme case, the most distant components after the filter update are merged. Given a single object, $N_h = 4$ is set in the simulation.

TABLE 5.1 reports averaged rotational root mean square error (RMSE) and ANEES in 100 runs. The process noise deviation $\sqrt{Q}$ is increased to simulate low-grade sensors. The proposed merge decreases the estimation error further and outputs more consistent estimates as $\sqrt{Q}$ increases. The more uncertain the process model is, the more opportunities multimodal a posteriori occurs. That is, $\|\Delta x\|^2$ in (5.8) becomes larger. Furthermore, as the proposed method directly merges distributions at the predetermined midway point, it spares computing the adjoint to back-project the merged distribution. This reduces execution time, as shown in Fig. 5.5b.

5.5.2 Photo-realistic Simulation

The objective of this simulation is to quantitatively show the effectiveness of the Gaussian midway-merge method in terms of localization and object map-

Table 5.1: Rotational root mean square error and averaged normalized estimation error squared in 100 Monte-Carlo runs

Noise level	Method	Rotation robot [deg]	Rotation object [deg]	ANEES
$\sqrt{10^0\,Q}$	GM ($\mathcal{T}_L$)	0.919	0.885	1.023
	GM (*midway*)	0.919	0.885	1.023
$\sqrt{10^1\,Q}$	GM ($\mathcal{T}_L$)	1.540	1.246	1.026
	GM (*midway*)	1.540	1.246	1.026
$\sqrt{10^2\,Q}$	GM ($\mathcal{T}_L$)	2.677	2.021	1.036
	GM (*midway*)	**2.676**	**2.020**	1.036
$\sqrt{10^3\,Q}$	GM ($\mathcal{T}_L$)	4.564	3.312	1.083
	GM (*midway*)	**4.561**	**3.309**	**1.082**
$\sqrt{10^4\,Q}$	GM ($\mathcal{T}_L$)	7.413	5.096	1.449
	GM (*midway*)	**7.400**	**5.079**	**1.443**
$\sqrt{10^5\,Q}$	GM ($\mathcal{T}_L$)	10.646	6.502	1.890
	GM (*midway*)	**10.582**	**6.396**	**1.863**

Table 5.2: IMU specification in the virtual environment

Specifications	Gyroscope	Accelerometer
Sampling rate	100 Hz	100 Hz
Random walk	0.3 deg/$\sqrt{\text{hr}}$	0.14 m/s/$\sqrt{\text{hr}}$
Bias stability	4.6 deg/hr	0.036 mg

Table 5.3: Drone localization and object mapping error in pose

Methods	Robot error [m]	Robot error [°]	Object error [m]*	Object error [°]*
Only propagation	0.69	13.62	-	-
GM ($\mathcal{T}_L$)	0.25	8.12	0.10	6.35
GM (*midway*)	**0.23**	**7.68**	**0.08**	**3.03**

*Mean error of all estimated objects

ping accuracy. To achieve this, a photo-realistic virtual environment based on AirSim [83] and YCB objects [108,110] in a room-scale environment is designed as shown in Fig. 5.6. A drone flies following the trajectory shown in Fig. 5.6 capturing stereo images and IMU measurements where TABLE 5.2 shows specifications of the IMU. A virtual stereo camera has 12cm of a baseline with 960×600 resolution images in 20 fps.

The main difference to the previous simulation is that the merge distance increases due to the larger scale scenario. Fig. 5.7 and Fig. 5.8 highlight the successful case of the proposed merge method on matrix Lie groups. As expected from the approximation error analysis in Section 5.2.3 and the Monte-Carlo simulation, GM(*midway*) improves the estimation accuracy when compared to GM($\mathcal{T}_L$). The quantitative results are shown in Table 5.3.

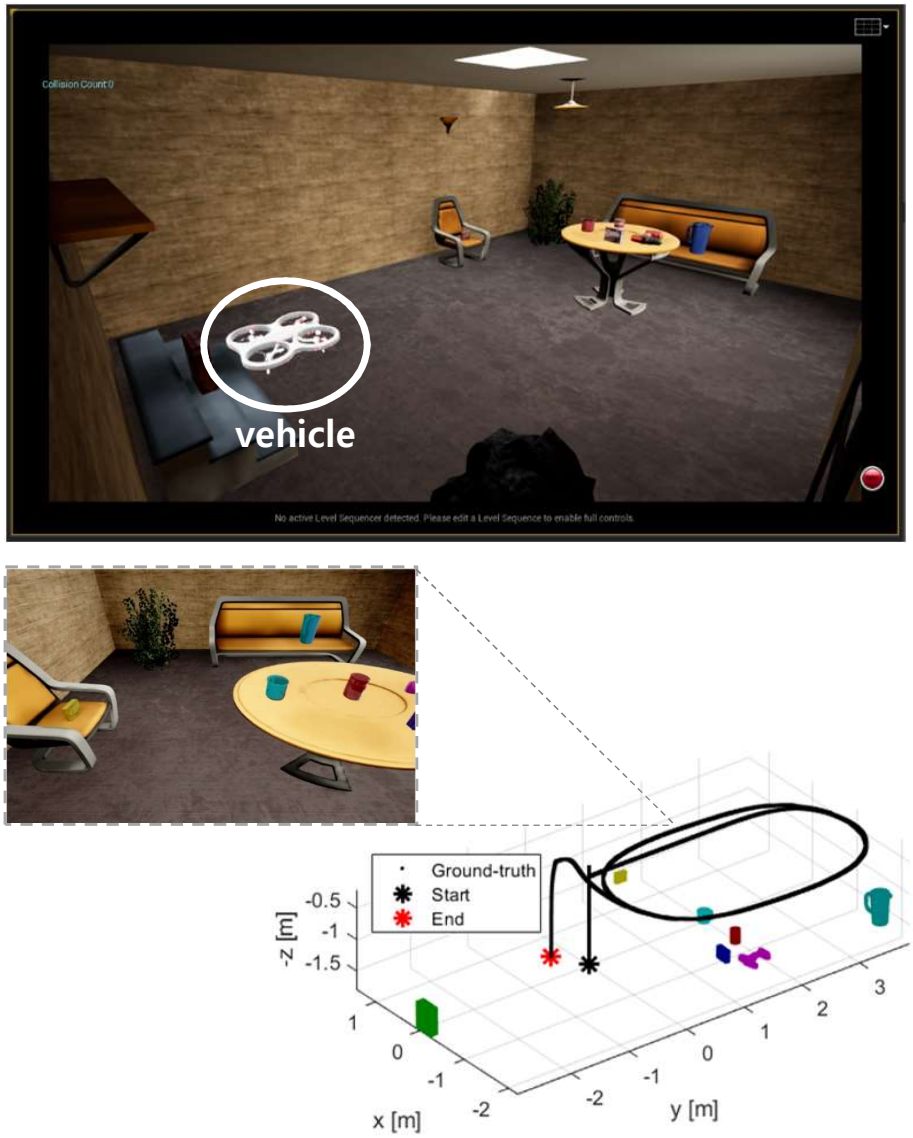


Figure 5.6: A room-scale virtual environment with YCB objects in the scene and the ground-truth trajectory of a drone with an onboard sample image

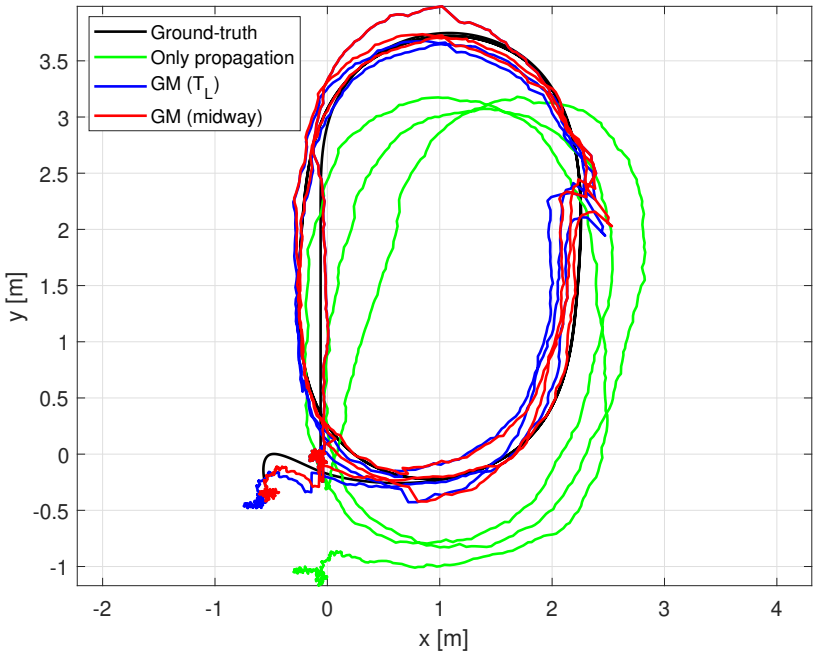


Figure 5.7: The ground-truth and estimated trajectory in the virtual environment. Only propagation: no object measurements; GM-IEKF with the previous merge GM ($\mathcal{T}_L$) and the proposed merge method GM (*midway*).

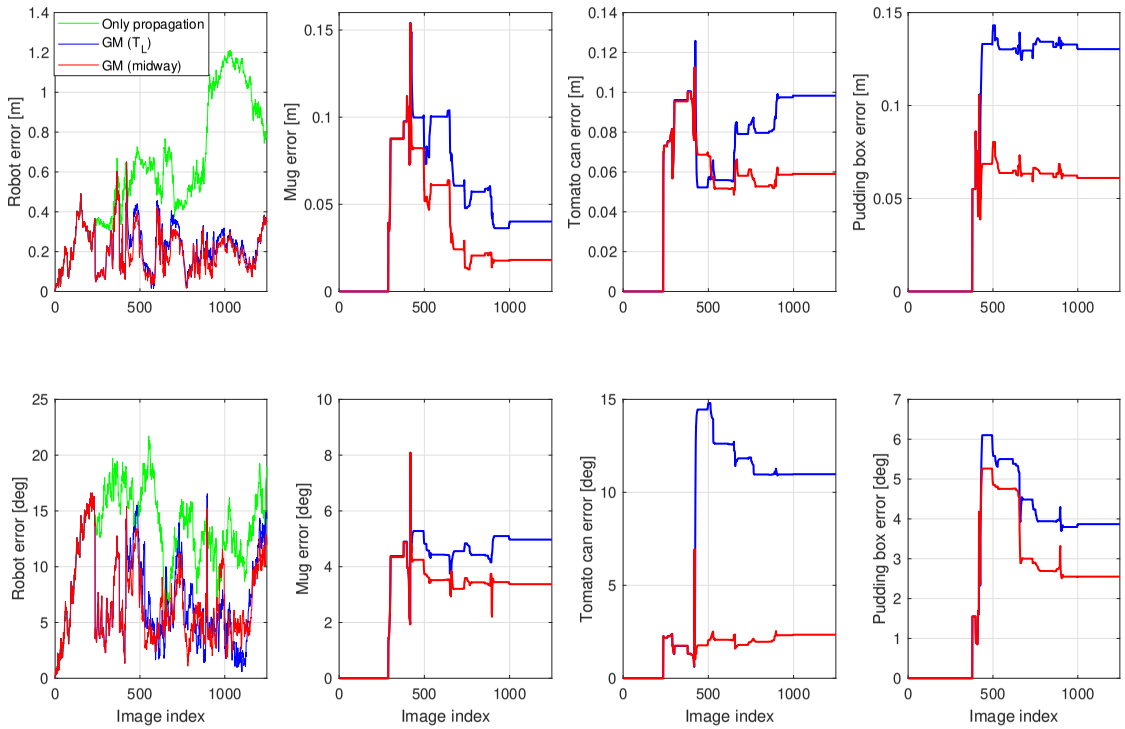


Figure 5.8: Position and rotation error of the drone and object mapping error in position and rotation for Mug, Tomato can, and Pudding box.

5.5.3 Real-world Datasets

In the YCB-Video dataset [108], this section validates the object-based SLAM formulated by GM-IEKF with the midway-merge method. The goal of this test is to investigate the efficacy of the midway-merge and how estimation error decreases with respect to the state-of-the-art 6 DOF pose detector, object SLAM (RIEKF [61]) and low-level SLAM (ORB-SLAM3 [22] with RGB-D). Among possible candidates such as CosyPose [109] and PoseRBPF [62], CosyPose with a single view is adopted as a sensor trained by the authors because of its superior performance. Data sequences with a mug are selected where the mug possesses heavy-tailed noise distribution along the symmetric axis as analyzed in Fig. 5.4. The noise characteristic of CosyPose is predictable since the authors introduce *the symmetric distance* in the training loss function. This allows the network to train symmetric objects, but at the same time, this admits estimation errors due to the symmetry. This error is modeled by a GM distribution in the filter.

Objects other than the mug in scenes are also modeled as heavy-tailed distribution for a generalization of the proposed approach. Given multiple objects, N_h is set as 12. Please note that objects that output very unstable estimates are excluded, such as bowls. Since odometry measurements are unavailable in the dataset, a constant velocity model is assumed.

TABLE 5.5 reports RMSE of the robot (camera) and mug pose, and robot-mug relative pose. GM-IEKF with two merge methods outputs almost identical estimation error. This is due to the low noise level Q (slow and smooth motion) as analyzed in TABLE 5.1. However, the midway-merge consistently reduces execution time with respect to the conventional merge [69] as summarized in TABLE 5.4 by sparing the back-projection procedure. The execution time for GM(*midway*) corresponds to the time to process the 14 line in Algorithm 2

per frame, while the midway-merge is replaced by the conventional merge in GM($\mathcal{T}_L$). The back-projection involves two dense matrix multiplication in a state dimension N. In the YCB-Video dataset, $N = 36$ depending on the number of objects, and TABLE 5.4 indicates that 13 times GM merge on average per frame is not trivial. Note that the Gaussian mixture model sacrifices computational burden due to the multi-hypothesis modeling when compared to the single hypothesis [61]. However, multiple filter updates are parallelizable, and in terms of *Update per filter*, which means execution time per N_h, the time budget is almost identical when compared to the single hypothesis case. Therefore, if accuracy is prioritized over the computational burden, the proposed approach would be a proper choice since it effectively mitigates the large rotational error.

In contrast, filtered pose by the proposed approach GM(*midway*) reduces 49.5% of relative rotation error when compared to CosyPose. The significant rotational error reduction is observed in the sequence 0022 where the mug handle is occluded in most scenes. Fig. 5.9 and Fig. 5.11 highlight that the pose detector suffers from the large deviation when the handle is occluded, while the proposed approach mitigates this by perceiving the prior pose. Fig. 5.10 also shows other examples of ambiguous objects. Please see the supplementary video for further visualization.[2] RIEKF [61], fed by the identical 6 DOF pose as GM methods, also improves pose error when compared to CosyPose thanks to the prior information. However, the method cannot explicitly address the noise due to symmetry as opposed to the proposed method, which leads to a large rotation error as highlighted in Fig. 5.9 and Fig. 5.10. Lastly, GM(*midway*) improves the robot localization accuracy with respect to ORB-SLAM3 (RGB-D) in most sequences, even without using depth measurements. This implies that objects possess rich information for localization and mapping.

[2] https://youtu.be/EvtW-mb8YK8

Table 5.4: Average execution time in millisecond per frame

Sequence	Method	Update per filter	Merge
0000	RIEKF [61]	0.28	-
	GM ($\mathcal{T}_L$)	0.29	3.10
	GM (*midway*)	0.28	2.92
0007	RIEKF	0.28	-
	GM ($\mathcal{T}_L$)	0.39	3.95
	GM (*midway*)	0.40	3.52
0022	RIEKF	0.36	-
	GM ($\mathcal{T}_L$)	0.49	4.93
	GM (*midway*)	0.48	4.28
0027	RIEKF	0.28	-
	GM ($\mathcal{T}_L$)	0.40	3.97
	GM (*midway*)	0.39	3.48
0033	RIEKF	0.48	-
	GM ($\mathcal{T}_L$)	0.65	6.70
	GM (*midway*)	0.65	5.90
0039	RIEKF	0.47	-
	GM ($\mathcal{T}_L$)	0.64	6.72
	GM (*midway*)	0.64	5.91
Mean	RIEKF	**0.36**	-
	GM ($\mathcal{T}_L$)	0.48	4.90
	GM (*midway*)	0.47	**4.34**

Table 5.5: Root mean square error of the robot and mug pose in the YCB-Video dataset

Sequence	Method	Robot position [cm]	Robot rotation [deg]	Object position [cm]	Object rotation [deg]	Relative position [cm]	Relative rotation [deg]
0000	ORB-SLAM3 [22]	1.46	1.52	-	-	-	-
	CosyPose [109]	-	-	-	-	0.77	2.89
	RIEKF [61]	1.69	2.15	1.83	1.64	0.81	1.96
	GM ($\mathcal{T}_L$)	1.62	2.03	1.77	1.80	0.82	1.84
	GM (*midway*)	1.62	2.03	1.75	1.80	0.82	1.84
0007	ORB-SLAM3	3.70	2.63	-	-	-	-
	CosyPose	-	-	-	-	0.93	4.30
	RIEKF	3.57	3.15	3.52	2.07	1.06	2.43
	GM ($\mathcal{T}_L$)	3.53	2.26	3.48	1.87	0.98	1.67
	GM (*midway*)	3.50	2.25	3.46	1.86	0.98	1.67
0022	ORB-SLAM3	3.11	3.63	-	-	-	-
	CosyPose	-	-	-	-	0.90	7.75
	RIEKF	2.77	3.90	2.25	3.24	0.90	3.48
	GM ($\mathcal{T}_L$)	2.73	1.80	2.30	2.64	0.83	2.68
	GM (*midway*)	2.82	1.82	2.36	2.65	0.83	2.69
0027	ORB-SLAM3	1.61	2.94	-	-	-	-
	CosyPose	-	-	-	-	1.15	2.86
	RIEKF	1.66	2.97	1.86	1.09	1.14	2.28
	GM ($\mathcal{T}_L$)	1.51	2.23	1.72	1.10	1.12	2.38
	GM (*midway*)	1.53	2.23	1.74	1.10	1.12	2.38
0033	ORB-SLAM3	1.37	1.95	-	-	-	-
	CosyPose	-	-	-	-	1.02	4.01
	RIEKF	1.35	2.79	1.75	3.61	1.03	2.38
	GM ($\mathcal{T}_L$)	1.52	1.63	1.85	2.53	0.95	2.49
	GM (*midway*)	1.43	1.63	1.78	2.53	0.95	2.49
0039	ORB-SLAM3	2.01	1.99	-	-	-	-
	CosyPose	-	-	-	-	0.83	1.44
	RIEKF	1.82	1.46	1.48	1.38	0.84	0.82
	GM ($\mathcal{T}_L$)	1.93	1.03	1.67	0.93	0.83	0.67
	GM (*midway*)	2.02	1.08	1.64	0.99	0.83	0.67
Mean	ORB-SLAM3	2.21	2.44	-	-	-	-
	CosyPose	-	-	-	-	0.93	3.88
	RIEKF	2.14	2.74	2.12	2.17	0.96	2.23
	GM ($\mathcal{T}_L$)	2.14	1.83	2.13	1.81	0.92	1.96
	GM (*midway*)	2.15	1.84	2.12	1.82	0.92	1.96

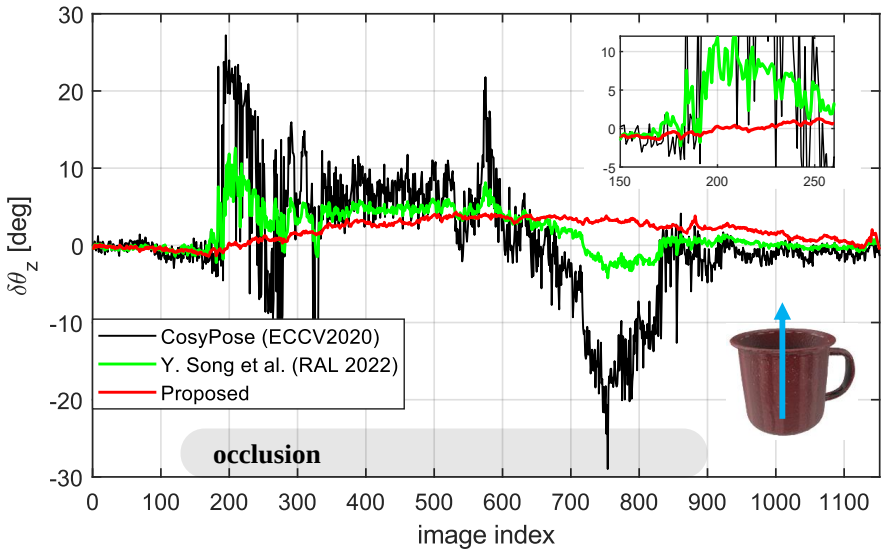
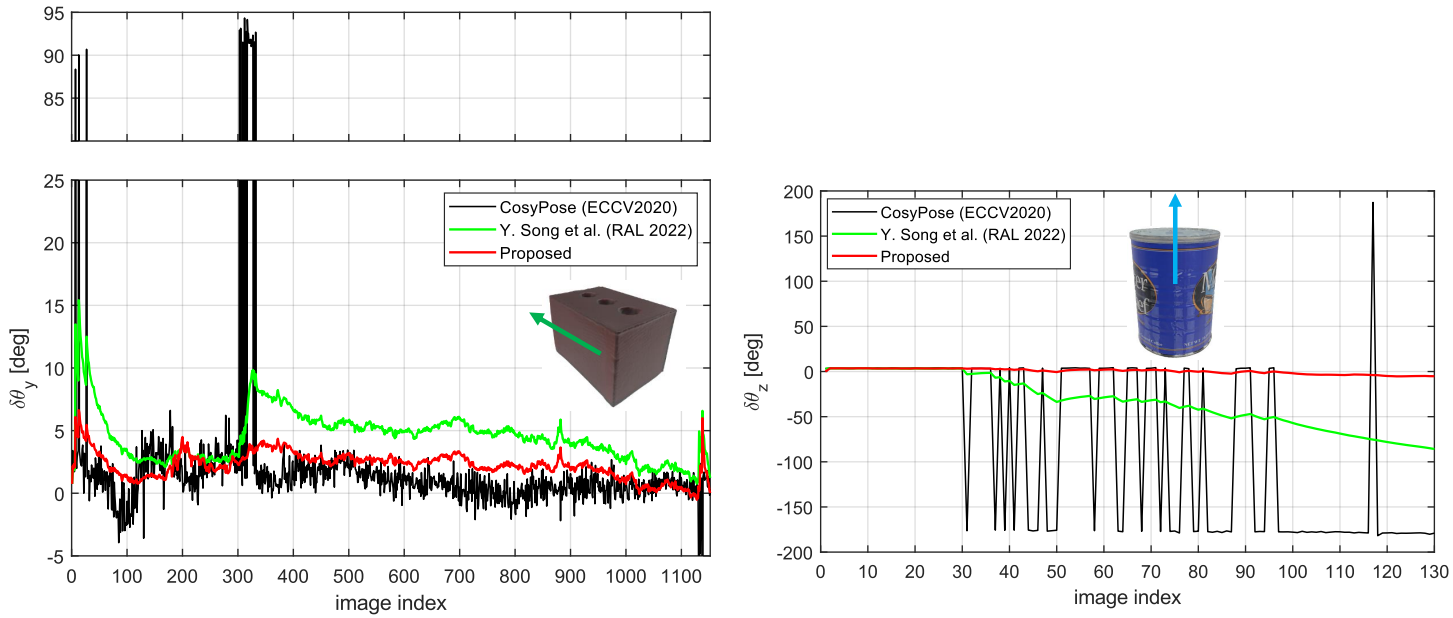


Figure 5.9: The rotational error of the symmetric axis of the mug in the 0022 sequence. The filtering method successfully mitigates large errors by virtue of prior information and the GM noise modeling.



(a) **(b)**

Figure 5.10: The rotational error of the ambiguous axis of the (a) brick and (b) can.

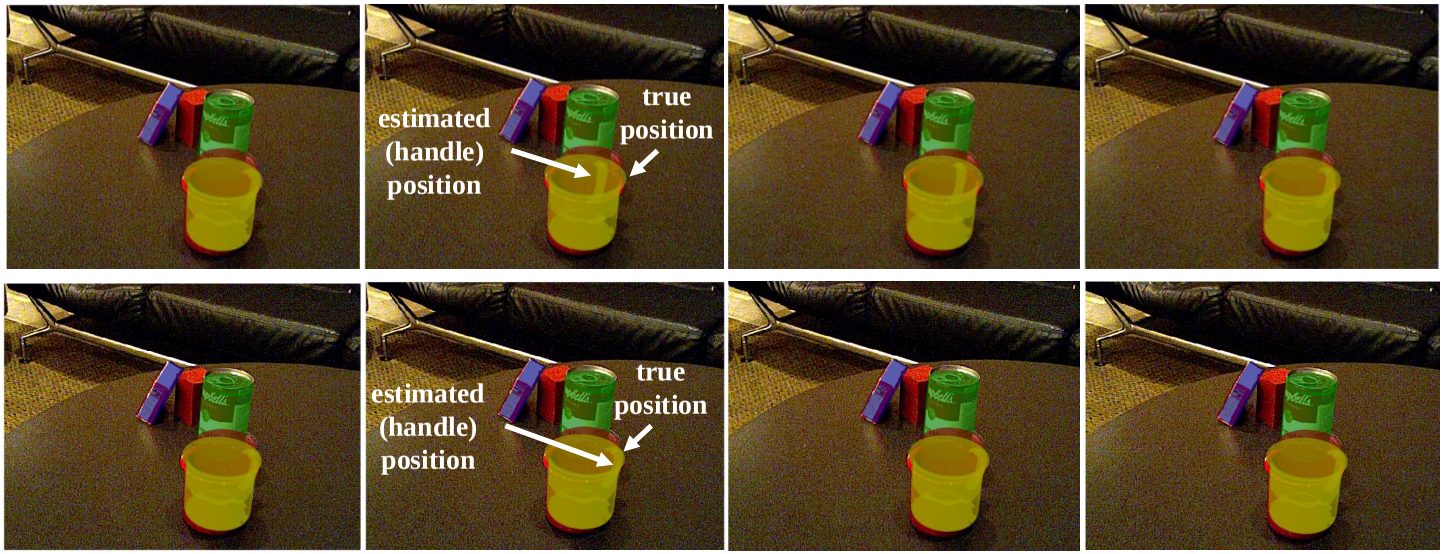


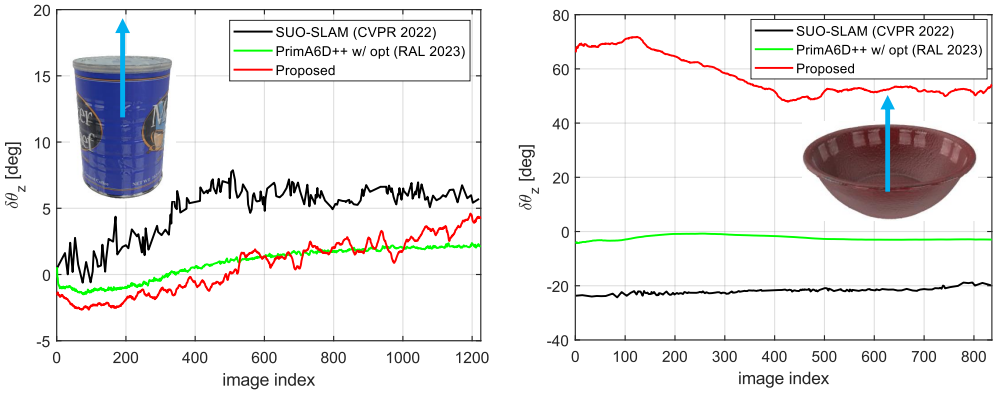
Figure 5.11: Temporally consecutive images (image index: from 206 to 209) that include the occluded mug in the 0022 sequence. Measurements (top row) and filtered pose (bottom row) are rendered on top of the image. The measurement clearly exhibits a large error along the symmetric axis.

5.6 Disccusion on different types of symmetric objects

As defined in Lee *et al.* [111] symmetric objects possesses either discrete (e.g., a rectangular table) or continuous ambiguity (e.g., a round table). This section investigates the pose estimation accuracy of discrete and continuous symmetric objects in the proposed approach along with the recent uncertainty-aware object SLAM methods: SUO-SLAM [67] and PrimA6D++ [112], and discusses the strengths and limitations of the proposed approach with promising future research direction.

Table 5.6 shows the relative camera-object pose error in the 0056 sequence of the YCB-Video dataset. Results other than the proposed method are obtained based on the corresponding open-source codes. Note that SUO-SLAM is in *slam mode* and PrimA6D++ is with the graph optimization using only RGB for fair comparison. Fig. 5.12a highlights the rotation error of *master chef can* where the object has discrete ambiguity along the ambiguous axis with $\pm 180°$. This shows the strength of the proposed filtering method: ambiguous observations by discrete symmetry can be successfully mitigated only with a single pass and single update iteration leading to comparable accuracy to the state of the art.

On the other hand, Table 5.7 shows results in the 0053 sequence where the scene includes a bowl having continuous symmetry. Fig. 5.12b represents the relative rotation error along the ambiguous axis where the proposed method shows the least accurate estimation. Since the proposed approach does not directly utilize image intensities but depends on a pose detector, a consistent bias included in pose detection is not observable in the estimator. Therefore, incorporating contour or intensity matching between a prediction and observation would be a desirable improvement direction to cope with continuous symmetric objects such as a bowl.



(a) **(b)**

Figure 5.12: The rotational error along the ambiguous axis of the (a) *master chef can* and (b) *bowl*

Table 5.6: Time-averaged RMSE of camera-object position [cm] / rotation [deg] error in the 0056 sequence of YCB-Video

Methods	SUO-SLAM [67]	PrimA6D++ w/ opt [112]	Proposed
master chef can	1.64 / 5.55	0.76 / **1.69**	**0.49** / 2.80
pitcher base	3.28 / 2.77	2.70 / 1.57	**2.43** / **1.26**
power drill	3.10 / 9.43	2.50 / **1.92**	**2.26** / 2.38

Table 5.7: Time-averaged RMSE of camera-object position [cm] / rotation [deg] error in the 0053 sequence of YCB-Video

Methods	SUO-SLAM [67]	PrimA6D++ w/ opt [112]	Proposed
tomato can	**1.24** / **1.53**	1.68 / 2.78	1.59 / 3.21
potted meat can	1.10 / 3.01	0.92 / **1.27**	**0.70** / 4.00
bowl	1.50 / 22.1	**0.52** / **2.78**	1.24 / 57.7

5.7 Conclusion

This chapter has proposed the *Gaussian mixture midway-merge* that merges Gaussian distributions on matrix Lie groups. Specifically, the proposed approach computes the merged mean and transforms the covariance matrices at the corresponding tangent space. This simple but powerful technique decreases information loss when the distance between mean matrices increases and computation time by sparing the adjoint. As a promising example, GM-IEKF, the Gaussian sum filter with the proposed merge for a symmetric object SLAM problem is formulated. A thorough Monte-Carlo simulation and demonstration on real-world as well as synthetic datasets reveal that the midway-merge has a lighter computational burden and a nice property when the noise level increases when compared to the conventional merge. The proposed method has great potential in state estimation on matrix Lie groups that deals with Gaussian mixtures.

Future work includes evaluating the presented approach with diverse symmetric objects in a long-range scenario. Generalizing the GM noise model to explain a symmetric object such as a bowl without any distinguished textures is also a part of future work.

Chapter 6

Visual-Inertial Object SLAM System Integration

This chapter contains the contents of the following conference publication:

J. H. Jung and C. G. Park, "**A Framework for Visual-Inertial Object-Level Simultaneous Localization and Mapping,**" *in IEEE/ION Position, Location and Navigation Symposium,* 2023, pp. 1335–1340, doi: 10.1109/PLANS53410.2023.10140108

This chapter presents a framework of simultaneous localization and mapping (SLAM) by combining the modular visual-inertial odometry (VIO) and object SLAM estimator. Semantic objects are known to possess rich localization information, such as scale and orientation. However, how to tightly couple these object measurements to an inertial sensor is not straightforward. To answer this, local object poses from a deep neural network are fused to build a globally consistent object map under precise prior estimates from the VIO module. The contribution of this work is the representation of the object map with six-dimensional poses that enables a robot to exploit orientational, as well as positional information in the filtering formulation. The proposed method can output cm-level accuracy localization and mapping in a room-scale environment in a photo-realistic virtual environment.

6.1 Introduction

Visual-inertial fusion for autonomous navigation has been tremendously studied in the last two decades, and the estimator architecture and a method for processing visual information have been established [74]. Images capture rich visual textures in a scene for visual navigation, but it does not contain the absolute scale and are sensitive to motion blur due to fast motion. On the other hand, an inertial measurement unit (IMU) outputs angular velocity and specific force in a much higher sampling frequency and contains the absolute scale for estimating a position of a vehicle. Therefore, an IMU can bridge the gap between captured images, while the error accumulation can be mitigated by visual information.

However, most of the previous research focused on the so-called low-level visual features such as corners and lines [16, 113]. This allows a vehicle to perceive the metric information, for instance, the vehicle' s position is described by the cartesian coordinate. In contrast, high-level visual features such as semantic objects in an object-level approach have not only geometrically but also semantically valuable information to localize a moving vehicle and perceive its surrounding [10, 56]. In a semantic sense, a semantically labeled map can be generated, while rotation or scale information can be exploited in a geometric sense. The main objective of this chapter is to localize a moving platform in three-dimensional space while building globally as well as locally consistent object-level map parameterized by a six-dimensional object pose in a stream of visual and inertial measurements.

On the other hand, six-dimensional object pose detection from images has been a challenging task in computer vision. Many previous works have shown remarkable estimation accuracy in benchmark datasets [62, 63, 109]. However,

multi-view constraints were not probabilistically exploited and their estimates contain large errors when facing symmetric objects. More importantly, most of these approaches did not address globally consistent object mapping while mainly focusing on the local poses of objects. A global understanding of environments is necessary when an intelligent agent performs a semantic task, such as "fetch me a clamp on the table.".

To tackle this, this chapter proposes a framework, as shown in Fig. 6.1 of visual-inertial object-level simultaneous localization and mapping (SLAM) that combines modular visual-inertial odometry (VIO) in Chapter 3 and object SLAM in Chapter 5 by jointly estimating the robot pose, as well as the objects' poses. The contribution of this chapter lies in the representation of the object map that enables a robot to process orientational, as well as positional localization information in the filtering formulation. Furthermore, it is shown that the proposed method can effectively mitigate pose errors from a deep neural network in a fully probabilistic way, generating a cm-level accuracy object map by fusing visual and inertial measurements in a simulation environment.

6.2 The System Overview

The presented system is built upon the previous chapters: the ensemble visual-inertial odometry and object SLAM with pose ambiguity. Fig. 6.1 shows the overall block diagram of the system. First, the VIO module minimizes the image intensity difference of sequential images using iterated extended Kalman filter, then estimates pose increment from the previous time step. After updating the estimator by the kth image, the delta pose increment ΔT is passed to the object SLAM module,

$$\Delta T = \left(T_{b_r}^{g}\right)^{-1} T_{b_k}^{g} \tag{6.1}$$

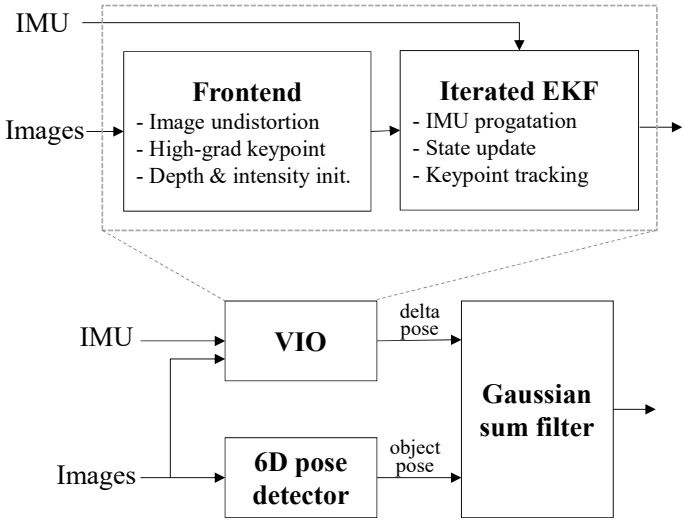


Figure 6.1: Overview block diagram of the presented system

where

$$T_{b_i}^g = \begin{bmatrix} R_{b_i}^g & p_{b_i}^g \\ 0 & 1 \end{bmatrix}, \ i \in \{r, k\}. \tag{6.2}$$

Second, a deep neural network-based six-dimensional pose detector estimates the relative robot-object pose. Lastly, Gaussian sum filter-based object SLAM estimates objects, as well as the robot poses in the global frame.

6.3 Simulation Results

The objective of this simulation is to quantitatively evaluate localization and mapping error decreases when using objects as measurements. The simulation environment developed in Section 5.5.2 is used. For implementation details, locally high-gradient visual keypoints are extracted in the left image with a maximum number of 200 in the VIO module. Then the depth is initialized based on the stereo baseline. Bad-conditioned keypoints are detected based on normalized cross-correlation between consecutive two views. For the ob-

ject SLAM, pre-trained CosyPose with a single view [109] is adopted and the maximum number of hypotheses is assigned as 12. This means that Gaussian distributions are merged iteratively until reaching the predefined number after updating Gaussian sum filter. Also, filter estimates are reported as the merged mean and covariance matrix of the largest and the second-largest weighted distributions.

6.3.1 VIO error statistics

The actual delta pose error histogram obtained from the EnVIO is presented. Fig. 6.2 shows a six-dimensional pose error histogram that has been obtained through the virtual environment. Qualitatively, the histogram follows Gaussian-like distributions that match the assumption (if ergodic). The noise standard deviations for object SLAM are set based on the error statistics, $\sigma_R = 0.054\ \mathrm{deg}/\sqrt{\mathrm{s}}$ and $\sigma_p = 0.009\ \mathrm{m}/\sqrt{\mathrm{s}}$. These parameters would be the starting point of filter tunning in different environments for generalization.

6.3.2 Pose detector error analysis

In the implementation, CosyPose with a single view pre-trained by the authors [109] is used as a perception sensor. Fig. 6.3 clearly shows that the deep neural network inherently possesses measurement noise. In other words, symmetric objects such as a mug suffer from large yaw errors due to the ambiguous shape. It is clearly seen in the overlaid image by the estimated mask in Fig. 6.3 as the handle is not consistent with the true position. This motivated us to model the object SLAM problem by Gaussian sum filter to account for multi-hypothesis

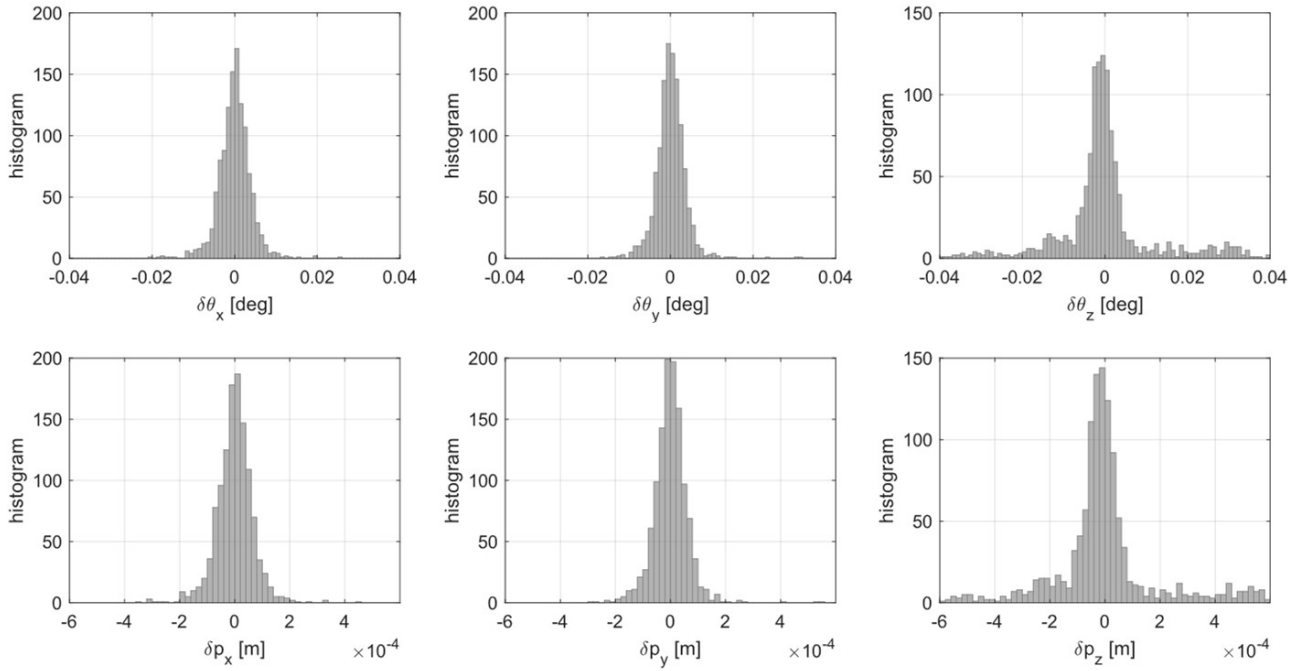


Figure 6.2: The visual-inertial odometry noise statistics for all six-dimensional axes: orientation error at the top and position error at the bottom.

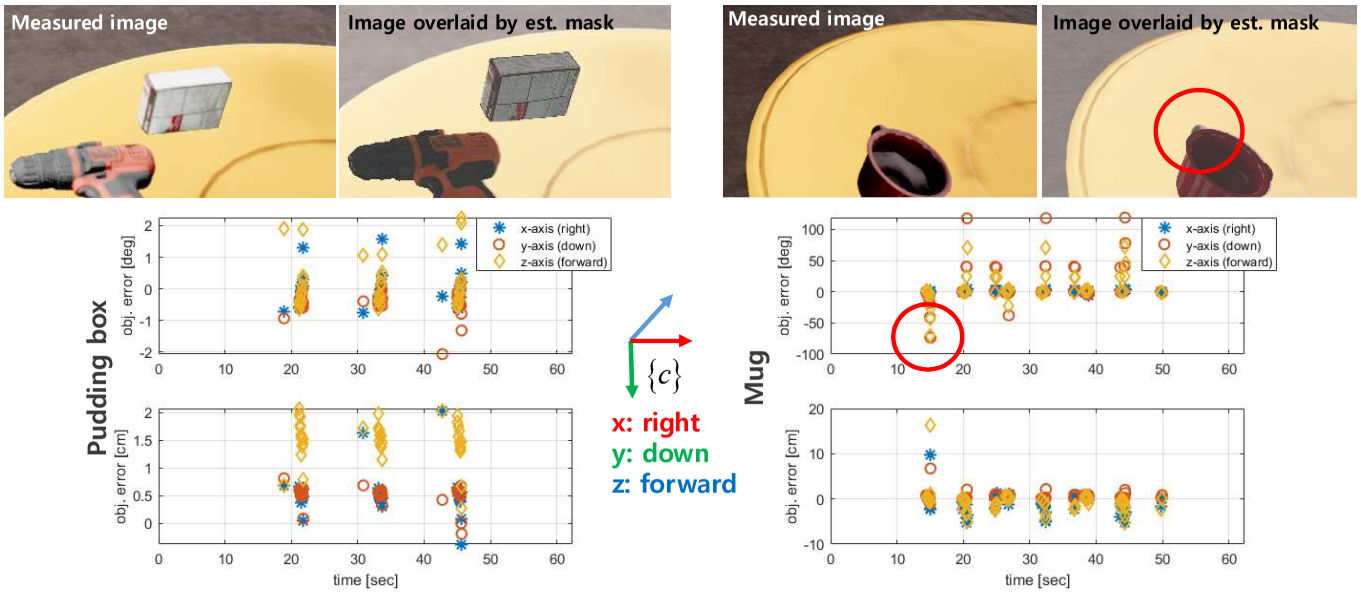


Figure 6.3: Six-dimensional pose detector (CosyPose with a single view) error analysis for the pudding box and the mug in the virtual environment: (Top) measured image from the drone and the same image overlaid by pose estimates from the detector; (Bottom) the corresponding robot-object relative pose error a long time where the large orientation error along the symmetrical axis of the mug is highlighted.

measurement error. In this simulation, the noise vector n_j in (5.25) is

$$\begin{aligned} n_j \sim\, & 0.7\,\mathcal{N}\left(0, \mathrm{diag}(4^\circ, 4^\circ, 4^\circ, 5\,\mathrm{mm}, 5\,\mathrm{mm}, 5\,\mathrm{mm})\right) \\ & + 0.3\,\mathcal{N}\left(0, \mathrm{diag}(4^\circ, 4^\circ, 8^\circ, 5\,\mathrm{mm}, 5\,\mathrm{mm}, 5\,\mathrm{mm})\right) \end{aligned} \tag{6.3}$$

where the larger noise 8° along the z-axis addresses the object symmetry.

6.3.3 Robot Localization

The effectiveness of object measurements is investigated when fused with the EnVIO. Fig. 6.4 shows the orientation and position errors of the drone at the global frame $\{g\}$. Here, *INS* is an inertial navigation system without any measurements for baseline, *OpenVINS* [15] is a state-of-the-art method for comparison, *EnVIO* is a VIO module that builds the framework, and *OBJ VI-SLAM* is the proposed framework that fuses the pose increment and local object poses from a pose detector. Along with the quantitative summary in TABLE 6.1 for the corresponding errors in Fig. 6.4, it is seen that the VIO module, *EnVIO* clearly mitigates large errors of *INS* and shows comparable results to the state-of-the-art method, *OpenVINS*. Given the relatively lower orientation precision of the pose detector than the IMU, the orientation error of the drone in *OBJ VI-SLAM* has not been improved. However, the position error reflects the effectiveness of object measurements improving position accuracy with large margins when compared to *EnVIO*.

Also, the vibrating behavior of the yaw error in Fig. 6.4 is observed when the drone turns. This originated from the Euler integration in the implementation, and more sophisticated integration such as the trapezoidal method can mitigate this issue.

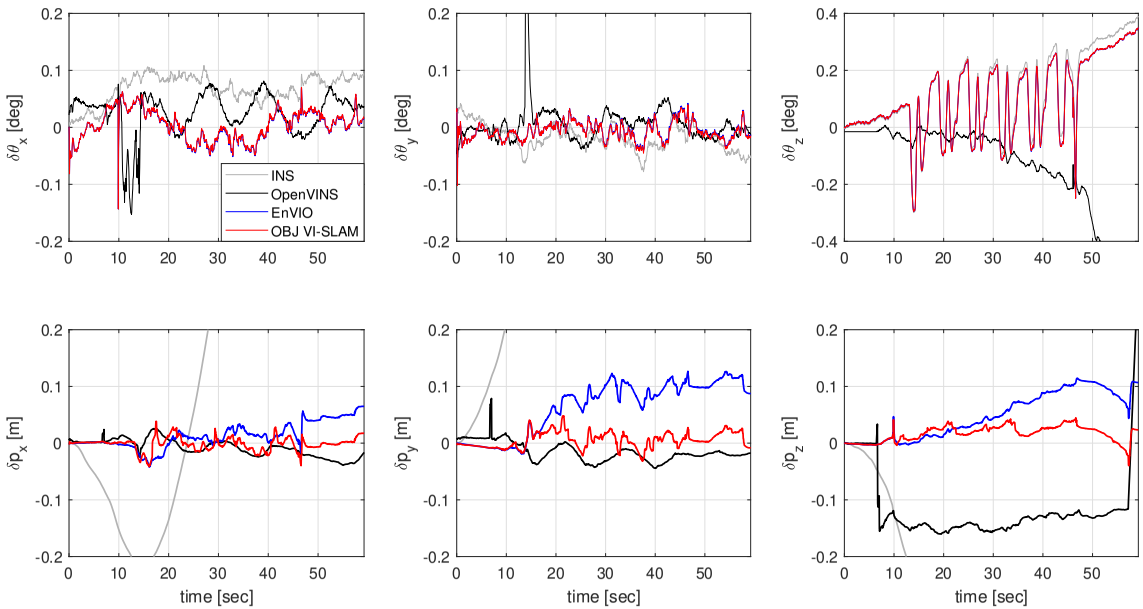


Figure 6.4: Orientation and position error referenced at the gravity-aligned (z-axis) global frame $\{g\}$ of the robot along time in the simulation.

Table 6.1: Orientation and position root mean square error along time for each axis in the simulation

Estimators	$\delta\theta_x$ [deg]	$\delta\theta_y$ [deg]	$\delta\theta_z$ [deg]	δp_x [cm]	δp_y [cm]	δp_z [cm]
INS	0.072	0.029	0.187	131	811	171
OpenVINS [15]	0.045	0.050	0.245	1.69	2.26	13.0
EnVIO	0.027	**0.017**	0.168	2.64	7.93	6.34
OBJ VI-SLAM	**0.026**	0.018	**0.165**	**1.15**	**1.47**	**2.12**

6.3.4 Object mapping

Lastly, this section investigates the effectiveness of object pose filtering by comparing the raw measurements from the pose detector and filtered object pose in the framework. Fig. 6.5 draws object mapping errors in terms of the orientation and position errors of the two objects. It is evident that the pose network inherently possesses estimation error due to sensor noise and the discrepancy between training and test data. Especially, along the symmetric z-axis, the network suffers from large errors due to shape symmetry. The proposed approach successfully mitigates this thanks to the prior information from the VIO module and multi-hypothesis noise modeling in Gaussian sum filter adoption.

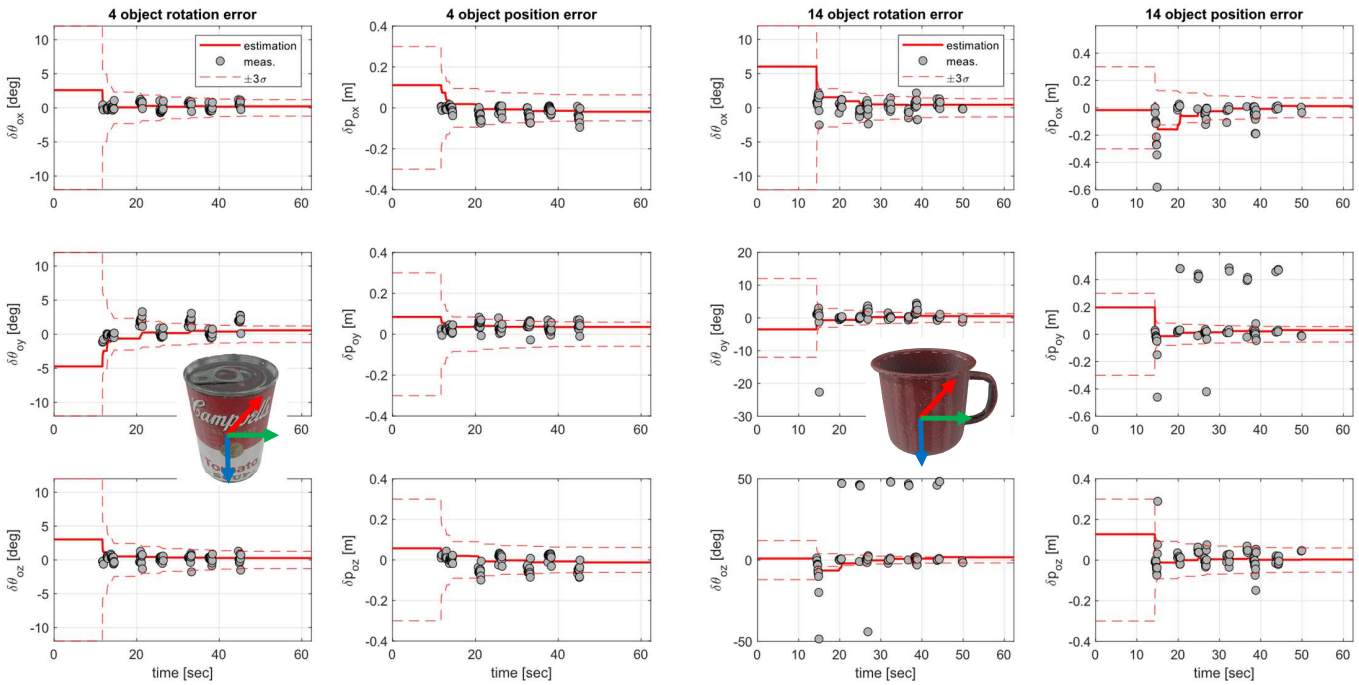


Figure 6.5: Raw pose measurement errors from the pose detector (green dots) and estimation errors (solid red lines) with $\pm 3\sigma$ bounds (dashed red lines) in the estimator of two YCB objects: tomato can and mug in the simulation.

6.4 Conclusion

This chapter has proposed a framework of visual-inertial object-level SLAM that includes a VIO module and joint estimation of objects, as well as robot poses. To tackle the measurement ambiguity of artificial objects, a multi-hypothesis noise vector is encoded in Gaussian sum filter. In the photo-realistic simulation, test results have shown the effectiveness of using object measurements in the context of visual-inertial SLAM.

Nonetheless, the current study has some limitations that motivate future work. First, it has been assumed that objects are static. It is desirable to release this assumption by studying various aspects of the target tracking literature. Second, this study heuristically sets the noise vector in the measurement model. Future work includes assigning noises based on the Bayesian neural network for the principled uncertainty.

Chapter 7

Conclusion

7.1 Concluding Remarks

Toward a robust visual-inertial navigation system, this study has made theoretical contributions built on matrix Lie groups, yet practical implementations of Bayesian filtering that tracks the underlying posterior distribution. Visual-inertial state estimation has been tremendously studied and implemented in commercial products in the past two decades. However, it is still not possible to cope with every failure case due to degeneracy in sensors. To close the gap between the research objective and the state of the art, this dissertation focused on estimation uncertainty in dealing with the photometric and pose measurements. Specifically, this study addressed robustness to large initial uncertainty and robustness to ambiguous sensor measurements. Through this research, it is possible to resolve such conditions. Starting with the basic idea followed by a very fundamental simulation study, the effectiveness of the proposed approaches is validated through intensive real-world datasets and experiments.

First, this study proposed the *optimal image gradient* that minimizes the expectation of the linearization error squared. This is a generalization from a deterministic to a stochastic system by accounting for the projective uncertainty in an image domain. The proposed approach was implemented in visual-inertial odometry (VIO), which directly minimizes the photometric er-

ror, and the optimal gradient was realized by sampling ensembles according to the state uncertainty. To deal with the nonlinearity in image intensities, this study adopted the iterated EKF to propagate the mean and covariance matrix, and this photometric formulation spares the explicit data association among keypoints. The strength of the method lies in a nice property that enlarges the convergence basin for iterative estimators, leading to high robustness to the initial uncertainty. The ablation study showed the effectiveness of the gradient in terms of estimation accuracy and consistency with increasing initial velocity error. In conjunction with the image pyramid, a standard technique to flatten local minima, the proposed method exhibited successful convergence up to 3 m/s velocity error that can possibly occur during inflight initialization. Furthermore, this study developed a real-time state estimator and demonstrated it in a flying robot experiment. The real-world drone flight tests revealed accurate (18 cm, 0.60°) and real-time performance (36 ms) in a laptop CPU, in which state-of-the-art estimators struggled to achieve this accuracy.

Second, this study proposed to formulate an object-based simultaneous localization and mapping (SLAM) problem in the invariant EKF. The conventional EKF-SLAM suffers from inconsistency since the unobservable bases depend on the linearization point. This previous finding was expanded to a pose measurement on matrix Lie groups. Specifically, this study derived that the unobservable directions are independent of the linearization point, leading to accurate and consistent estimation performance. This is a generalization of the classical keypoint-based to object-based SLAM. Through a Monte-Carlo simulation, the proposed method achieved consistency giving an average normalized estimation error squared as 1.07, while the conventional EKF gains spurious information along the yaw direction. Furthermore, it was shown that the presented approach can be realized in a real-world driving scenario. This study

adopted an off-the-shelf deep neural network as a perception sensor to detect object poses and analyzed the six-dimensional detection error to properly tune its measurement uncertainty in filtering. In an open-source driving dataset, the experimental results showed the validity of object measurements, and the proposed method achieved comparable estimation accuracy (relative pose error: 1.35 %) using only object measurements when compared to state-of-the-art object-based SLAM methods.

Third, this study proposed the *Gaussian mixture midway-merge* method to merge a pair of Gaussian distributions on matrix Lie groups. The key idea was to determine the common tangent space first, then warped distributions were merged at the associated space. It was proved that this rule yields less approximation error than the conventional merge method. This theoretical contribution was confirmed by investigating the dissimilarity between densities before and after merging. As a promising application, this study tackled a challenge in object-based SLAM with a symmetric shape, where the number of hypotheses exponentially increases over time in a naive approach. To be specific, the Gaussian sum filter was formulated to address the multiple hypotheses where the second work of this dissertation implements each filter. The increasing hypotheses are bounded by the proposed merge method. Through a Monte-Carlo simulation and photo-realistic virtual environment, the strength of the presented merge was highlighted: The longer the distance between distributions, the less information loss when distributions are merged. This study focused on mug pose estimation, which exhibits heavy-tailed distribution due to occlusion, as a real example. The large rotation error of the state-of-the-art pose detector along the symmetric axis was mitigated. In a room-scale environment, the filtering method reduced the rotation error by 49.5% ($3.99^\circ \rightarrow 1.96^\circ$) on average over more than 10k images compared to the pose detector.

Lastly, this study presented a SLAM system combining VIO and ambiguity-aware SLAM developed in this dissertation. In this framework, the odometry module estimates a pose increment between images, while a pose detector outputs the six-dimensional robot-object pose. Gaussian sum filter embraces all the measurements and jointly estimates robot and object poses. In a photo-realistic simulator, this study showcased that the presented system can output cm-level localization and mapping accuracy in a room-scale environment.

7.2 Future Works

This dissertation has addressed challenges in visual-inertial state estimation. Nonetheless, there are still promising future works that further advance the estimation reliability toward a fail-safe and fail-aware system.

Resilient state estimation

Resilience means an ability to recover from a failure, while robustness is an ability to resist degraded conditions. This dissertation focused on the robustness to the large initial error and ambiguous measurements, but it is very challenging to proactively consider all extreme conditions that can occur in real applications. Therefore, the estimator should be resilient meaning it is aware of failures (*i.e.* estimation error diverges) and restarts its process with reconfigured parameters (*i.e.* more proper tuning parameters) or a set of sensors (*i.e.* selecting a lidar in a low-light condition) automatically. An intelligent agent would monitor estimation uncertainties to detect which sensor is failed and reload predefined parameters. More general system-level resiliency with minimum tuning factors will be an interesting research direction in this topic.

End-to-end learning with physics-based knowledge

It is evident that there are elements better dealt with a model-free approach in state estimation with perception sensors. This example includes perceiving salient visual features and semantic objects in a camera and finding point correspondences in a lidar. However, model-free approaches have a fundamental limitation in generalization to unseen novel data. On the other hand, model-based methods rely on the physical law to describe the relationship between sensor measurements and kinematics, but it is vulnerable to unmodeled errors. This complementary characteristic has led to physics-informed machine learning, and it is a powerful tool to balance between the model and the data. In this context, this dissertation suggests a fusion of the learning-based pose detector and traditional probabilistic estimator. Yet, the presented approach as well as the state of the art treat model-based and model-free methods independently, causing suboptimal training for a given task. A tight fusion of a deep neural network and differentiable optimizer in an end-to-end manner would be a promising future research direction for fail-critical systems.

Appendix A

Derivation of unobservable subspace in SO(3)-EKF

Starting with the derivation of error equations, this appendix analytically derives the unobservable subspace for an object-based SLAM problem. Note that "≈" is used when higher-order terms are neglected or an assumption is made. Also, note that "=" in the error equation holds up to higher-order terms. The interested state space is

$$\mathcal{X} = (R_b,\, p_b,\, v_b,\, R_o,\, p_o) \tag{A.1}$$

where R_o, p_o is the rotation and position of an object. Then, the underlying kinematics is

$$\begin{aligned}
\dot{R}_b &= R_b(\omega_m - b_g - n_g)^\wedge \\
\dot{p}_b &= v_b \\
\dot{v}_b &= R_b(a_m - b_a - n_a) + g \\
\dot{b}_a &= n_{wa} \\
\dot{b}_g &= n_{wg} \\
\dot{R}_o &= 0 \\
\dot{p}_o &= 0
\end{aligned} \tag{A.2}$$

where notations are defined in (3.8).

Jacobian matrix in continuous time

By perturbing (A.2) with the error definition made in (4.15), the error dynamics for rotation is

$$\begin{aligned}
\exp(-\phi_b^\wedge)(-\dot{\phi}_b^\wedge)\hat{R}_b + \exp(-\phi_b^\wedge)\dot{\hat{R}} &= \exp(-\phi_b^\wedge)\hat{R}_b\left(\omega_m - (\hat{b}_g - \delta b_g) - n_g\right)^\wedge \\
(-\dot{\phi}_b^\wedge)\hat{R}_b + \hat{R}_b\left(\omega_m - \hat{b}_g\right)^\wedge &\approx \hat{R}_b\left(\omega_m - (\hat{b}_g - \delta b_g) - n_g\right)^\wedge \\
-\dot{\phi}_b^\wedge &= \left(\hat{R}_b(\delta b_g - n_g)\right)^\wedge \\
\dot{\phi}_b &= \hat{R}_b(-\delta b_g + n_g).
\end{aligned} \tag{A.3}$$

For position,

$$\delta\dot{p}_b = \delta\dot{v}_b. \tag{A.4}$$

For velocity,

$$\begin{aligned}
\dot{\hat{v}}_b - \delta\dot{v}_b &= \exp(-\phi_b^\wedge)\hat{R}_b(a_m - (\hat{b}_a - \delta b_a) - n_a) + g \\
\hat{R}_b(a_m - \hat{b}_a) - \delta\dot{v}_b &\approx (I - \phi_b^\wedge)\hat{R}_b(a_m - (\hat{b}_a - \delta b_a) - n_a) \\
\delta\dot{v}_b &\approx -\left(\hat{R}_b(a_m - \hat{b}_a)\right)^\wedge \phi_b + \hat{R}_b(-\delta b_a + n_a).
\end{aligned} \tag{A.5}$$

For biases,

$$\begin{aligned}
\delta\dot{b}_a &= n_{wa}, \\
\delta\dot{b}_g &= n_{wg}.
\end{aligned} \tag{A.6}$$

For the object,

$$\begin{aligned}
\dot{\phi}_o &= 0, \\
\delta\dot{p}_o &= 0.
\end{aligned} \tag{A.7}$$

By rewriting the derived equations, the error dynamics is expressed as follows,

$$\frac{d}{dt}\begin{bmatrix}\phi_b\\ \delta p_b\\ \delta v_b\\ \delta b_a\\ \delta b_g\\ \phi_o\\ \delta p_o\end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & -\hat{R}_b & 0 & 0\\ 0 & 0 & Id & 0 & 0 & 0 & 0\\ -\left(\hat{R}_b(a_m-\hat{b}_a)\right)^\wedge & 0 & 0 & -\hat{R}_b & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix}\phi_b\\ \delta p_b\\ \delta v_b\\ \delta b_a\\ \delta b_g\\ \phi_o\\ \delta p_o\end{bmatrix}$$

$$+\begin{bmatrix} 0 & \hat{R}_b & 0 & 0\\ 0 & 0 & 0 & 0\\ \hat{R}_b & 0 & 0 & 0\\ 0 & 0 & Id & 0\\ 0 & 0 & 0 & Id\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} n_a\\ n_g\\ n_{wa}\\ n_{wg}\end{bmatrix}. \tag{A.8}$$

State transition matrix in discrete time

The deterministic part without biases is the interested quantity in this analysis. At time $t \in [t_{k-1}, t_k]$, the velocity error is

$$\delta v_b(t) = \delta v(t_{k-1}) - \int_{t_{k-1}}^{t} \left(R_b(\tau)a_m(\tau)\right)^\wedge \phi_b(\tau)d\tau. \tag{A.9}$$

Error states are discretized at $\{t_{k-1}, t_k\}$. For rotation,

$$\delta\phi_b(t_k) = \delta\phi_b(t_{k-1}). \tag{A.10}$$

For position,

$$
\begin{aligned}
\delta p_b(t_k) &= \delta p_b(t_{k-1}) + \int_{t_{k-1}}^{t_k} \delta v(t) dt \\
&= \delta p_b(t_{k-1}) + \int_{t_{k-1}}^{t_k} \delta v(t_{k-1}) dt - \int_{t_{k-1}}^{t} \left(R_b(\tau) a_m(\tau)\right)^\wedge \phi_b(\tau) d\tau dt \\
&= \delta p_b(t_{k-1}) + \Delta t_k \delta v(t_{k-1}) - \int_{t_{k-1}}^{t_k} \int_{t_{k-1}}^{t} \left(R_b(\tau) a_m(\tau)\right)^\wedge d\tau dt \, \phi_b(t_{k-1}),
\end{aligned}
\tag{A.11}
$$

where $\Delta t_k = t_k - t_{k-1}$ and ϕ_b is actually constant at the given time. For velocity,

$$
\delta v(t_k) = \delta v(t_{k-1}) - \int_{t_{k-1}}^{t_k} \left(R_b(\tau) a_m(\tau)\right)^\wedge d\tau \, \phi_b(t_{k-1}). \tag{A.12}
$$

Therefore, the state-transition matrix for the error state is expressed as follows.

$$
\begin{bmatrix} \phi_b(t_k) \\ \delta p_b(t_k) \\ \delta v_b(t_k) \\ \phi_o(t_k) \\ \delta p_o(t_k) \end{bmatrix}
=
\underbrace{\begin{bmatrix}
Id & 0 & 0 & 0 & 0 \\
-\int_{t_{k-1}}^{t_k} \int_{t_{k-1}}^{t} \left(R_b(\tau) a_m(\tau)\right)^\wedge d\tau dt & Id & \Delta t_k Id & 0 & 0 \\
-\int_{t_{k-1}}^{t_k} \left(R_b(\tau) a_m(\tau)\right)^\wedge d\tau & 0 & Id & 0 & 0 \\
0 & 0 & 0 & Id & 0 \\
0 & 0 & 0 & 0 & Id
\end{bmatrix}}_{\Phi(t_{k-1}, t_k)}
\begin{bmatrix} \phi_b(t_{k-1}) \\ \delta p_b(t_{k-1}) \\ \delta v_b(t_{k-1}) \\ \phi_o(t_{k-1}) \\ \delta p_o(t_{k-1}) \end{bmatrix}
\tag{A.13}
$$

Measurement Jacobian

A robot observes the relative robot-object pose in $\{b\}$ at time t_l without loss of generality since the extrinsic parameter is calibrated,

$$
\begin{aligned}
\mathcal{R}_o^b &= R_b^T R_o \exp(n_R^\wedge) \\
\mathcal{P}_o^b &= R_b^T (p_o - p_b) + n_p
\end{aligned}
\tag{A.14}
$$

where the sensor measures the pose up to the white Gaussian noises, n_R and n_p. By perturbing (A.14) according to the error definition in SO(3)-EKF, the measurement Jacobian matrix is obtained. For rotation,

$$\begin{aligned}
\exp(-\phi_{bo}^{\wedge})\hat{R}_o^b &= \hat{R}_b^T \exp(\phi_b^{\wedge}) \exp(-\phi_o^{\wedge}) \hat{R}_o \exp(n_R^{\wedge}) \\
\exp(-\phi_{bo}^{\wedge}) &= \hat{R}_b^T \exp(\phi_b^{\wedge}) \exp(-\phi_o^{\wedge}) \exp\left((\hat{R}_o n_R)^{\wedge}\right) \hat{R}_b \\
&\overset{(2.32)}{\approx} \hat{R}_b^T \exp\left((\phi_b - \phi_o + \hat{R}_o n_R)^{\wedge}\right) \hat{R}_b \\
&= \exp\left((\hat{R}_b^T (\phi_b - \phi_o + \hat{R}_o n_R))^{\wedge}\right) \\
\phi_{bo} &= \hat{R}_b^T (-\phi_b + \phi_o - \hat{R}_o n_R).
\end{aligned} \tag{A.15}$$

For position,

$$\begin{aligned}
\hat{p}_o^b - \delta p_o^b &= \hat{R}_b^T \exp(\phi_b^{\wedge}) \left(\hat{p}_o - \delta p_o - (\hat{p}_b - \delta p_b)\right) + n_p \\
\delta p_o^b &\approx \hat{R}_b^T \left((\hat{p}_o - \hat{p}_b)^{\wedge} \phi_b - \delta p_b + \delta p_o\right) - n_p.
\end{aligned} \tag{A.16}$$

By rewriting error equations, the measurement Jacobian is expressed as follows.

$$\begin{bmatrix} \phi_{bo} \\ \delta p_o^b \end{bmatrix} = \underbrace{\begin{bmatrix} \hat{R}_b^T & 0 \\ 0 & \hat{R}_b^T \end{bmatrix} \begin{bmatrix} -Id & 0 & 0 & Id & 0 \\ (\hat{p}_o - \hat{p}_b)^{\wedge} & -Id & 0 & 0 & Id \end{bmatrix}}_{H(t_l)} \begin{bmatrix} \phi_b \\ \delta p_b \\ \delta v_b \\ \phi_o \\ \delta p_o \end{bmatrix} + \begin{bmatrix} -\hat{R}_b^T \hat{R}_o & 0 \\ 0 & -Id \end{bmatrix} \begin{bmatrix} n_R \\ n_p \end{bmatrix} \tag{A.17}$$

Observability matrix

Substituting $H(t_l)$ in (A.17) and $\Phi(t_l, t_0)$ in (A.13) into (4.14) yields the lth observability matrix block as follows.

$$
\begin{aligned}
\mathcal{O}_l^{\text{EKF}} &= \begin{bmatrix} R_b^T(t_l) & 0 \\ 0 & R_b^T(t_l) \end{bmatrix} \begin{bmatrix} -Id & 0 & 0 & Id & 0 \\ (p_o(t_l) - p_b(t_l))^\wedge & -Id & 0 & 0 & Id \end{bmatrix} \\
&\quad \times \begin{bmatrix} Id & 0 & 0 & 0 & 0 \\ -\int_{t_0}^{t_l}\int_{t_0}^{t} (R_b(\tau)a_m(\tau))^\wedge \, d\tau dt & Id & \Delta t_l Id & 0 & 0 \\ -\int_{t_0}^{t_l} (R_b(\tau)a_m(\tau))^\wedge \, d\tau & 0 & Id & 0 & 0 \\ 0 & 0 & 0 & Id & 0 \\ 0 & 0 & 0 & 0 & Id \end{bmatrix} \\
&= \begin{bmatrix} R_b^T(t_l) & 0 \\ 0 & R_b^T(t_l) \end{bmatrix} \\
&\quad \times \begin{bmatrix} -Id & 0 & 0 & Id & 0 \\ \left(p_o(t_l) - p_b(t_l) + \int_{t_0}^{t_l}\int_{t_0}^{t} R_b(\tau)a_m(\tau)d\tau dt\right)^\wedge & -Id & -\Delta t_l Id & 0 & Id \end{bmatrix}
\end{aligned}
\tag{A.18}
$$

where $\Delta t_l = t_l - t_0$ and it is assumed that the system is linearized at the true state. The right nullspace of the observability matrix is the unobservable subspace. Assume that the below bases constitute the unobservable subspace,

$$
N_1 = \begin{bmatrix} 0 \\ Id \\ 0 \\ 0 \\ Id \end{bmatrix}, \quad N_2 = \begin{bmatrix} g \\ -p_b(t_0)^\wedge g \\ -v_b(t_0)^\wedge g \\ g \\ -p_o(t_l)^\wedge g \end{bmatrix}. \tag{A.19}
$$

Then, it is clear to show that the global translation N_1 satisfies

$$\mathcal{O}_l^{\mathrm{EKF}} N_1 = 0. \tag{A.20}$$

For the global rotation about the gravity direction N_2,

$$\begin{aligned}
\mathcal{O}_l^{\mathrm{EKF}} N_2 &= \begin{bmatrix} R_b^T(t_l) & 0 \\ 0 & R_b^T(t_l) \end{bmatrix} \\
&\quad \times \begin{bmatrix} -g + g \\ \left(-p_b(t_l) + \int_{t_0}^{t_l} \int_{t_0}^{t} R_b(\tau) a_m(\tau) d\tau dt + p_b(t_0) + \Delta t_l v_b(t_0)\right)^{\wedge} g \end{bmatrix} \\
&= \begin{bmatrix} R_b^T(t_l) & 0 \\ 0 & R_b^T(t_l) \end{bmatrix} \begin{bmatrix} 0 \\ \frac{\Delta t_l^2}{2} g^{\wedge} g \end{bmatrix} \\
&= 0.
\end{aligned} \tag{A.21}$$

Therefore, N_1 and N_2 span the unobservable subspace.

Appendix B

Derivation of unobservable subspace in the proposed formulation

The interested state space is the same as in Appendix A, but it is expressed on matrix Lie groups. Specifically,

$$\mathcal{X} = (X_b, T_o)\,, \tag{B.1}$$

where

$$X_b = \begin{bmatrix} R_b & p_b & v_b \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad T_b = \begin{bmatrix} R_o & p_o \\ 0 & 1 \end{bmatrix}. \tag{B.2}$$

Then the underlying kinematics in a matrix form is

$$\begin{aligned} \dot{X}_b &= f_u(X_b) - X_b B - X_b N \\ \dot{T}_o &= 0, \end{aligned} \tag{B.3}$$

where

$$f_u(X_b) = \begin{bmatrix} R_b\,\omega_m^\wedge & v_b & R_b\,a_m + g \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$B = \begin{bmatrix} b_g^\wedge & 0 & b_a \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$N = \begin{bmatrix} n_g^\wedge & 0 & n_a \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{B.4}$$

Jacobian matrix in continuous time

For completeness, this appendix rephrases the error dynamics in [114] with the notations defined in this dissertation. Given the right invariant error, $\delta X_b = \hat{X}_b X_b^{-1}$, the error equation is derived as follows.

$$\begin{aligned} \delta\dot{X}_b &= \dot{\hat{X}}_b X_b^{-1} + \hat{X}_b(\dot{X_b^{-1}}) \\ &= \dot{\hat{X}}_b X_b^{-1} - \hat{X}_b X_b^{-1}\dot{X}_b X_b^{-1} \\ &\approx \left(f_u(\hat{X}_b) - \hat{X}_b\hat{B}\right) X_b^{-1} - \delta X_b\left(f_u(X_b) - X_b B - X_b N\right) X_b^{-1} \\ &= f_u(\delta X_b X_b)X_b^{-1} - \delta X_b f_u(X_b)X_b^{-1} - \hat{X}_b(\hat{B} - B)\hat{X}_b^{-1}\delta X_b + \hat{X}_b N \hat{X}_b^{-1}\delta X_b \end{aligned} \tag{B.5}$$

In the particular case, $X_b = Id$,

$$\delta\dot{X}_b = f_u(\delta X_b) - \delta X_b f_u(Id) - \hat{X}_b\delta B\hat{X}_b^{-1}\delta X_b + \hat{X}_b N\hat{X}_b^{-1}\delta X_b, \tag{B.6}$$

where $\delta B = \hat{B} - B$. Reminding the error state is

$$\begin{aligned}\delta X_b &= \exp(\zeta_b^\wedge) \approx Id + \zeta_b^\wedge \\ &= \begin{bmatrix} Id + \phi_b^\wedge & \rho_b & \nu_b \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},\end{aligned} \tag{B.7}$$

evaluating each term yields

$$f_u(\delta X_b) - \delta X_b f_u(Id) \approx \begin{bmatrix} 0 & \nu_b & g^\wedge \phi_b \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{B.8}$$

$$\hat{X}_b \delta B \hat{X}_b^{-1} \delta X_b \approx \begin{bmatrix} (\hat{R}_b \delta b_g)^\wedge & (\hat{p}_b)^\wedge \hat{R}_b \delta b_g & (\hat{v}_b)^\wedge \hat{R}_b \delta b_g + \hat{R}_b \delta b_a \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{B.9}$$

and

$$\hat{X}_b N \hat{X}_b^{-1} \delta X_b \approx \begin{bmatrix} (\hat{R}_b n_g)^\wedge & (\hat{p}_b)^\wedge \hat{R}_b n_g & (\hat{v}_b)^\wedge \hat{R}_b n_g + \hat{R}_b n_a \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{B.10}$$

Substituting these terms into (B.6) yields

$$\begin{aligned}\dot{\phi}_b &= -\hat{R}_b \delta b_g + \hat{R}_b n_g \\ \dot{\rho}_b &= \nu_b - (\hat{p}_b)^\wedge \hat{R}_b \delta b_g + (\hat{p}_b)^\wedge \hat{R}_b n_g \\ \dot{\nu}_b &= g^\wedge \phi_b - (\hat{v}_b)^\wedge \hat{R}_b \delta b_g + (\hat{v}_b)^\wedge \hat{R}_b n_g - \hat{R}_b \delta b_a + \hat{R}_b n_a.\end{aligned} \tag{B.11}$$

By rewriting the derived equations, the error dynamics is expressed as follows.

$$\frac{d}{dt}\begin{bmatrix}\phi_b\\ \rho_b\\ \nu_b\\ \delta b_a\\ \delta b_g\\ \phi_o\\ \rho_o\end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & -\hat{R}_b & 0 & 0\\ 0 & 0 & Id & 0 & -(\hat{p}_b)^\wedge \hat{R}_b & 0 & 0\\ g^\wedge & 0 & 0 & -\hat{R}_b & -(\hat{v}_b)^\wedge \hat{R}_b & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix}\phi_b\\ \rho_b\\ \nu_b\\ \delta b_a\\ \delta b_g\\ \phi_o\\ \rho_o\end{bmatrix}$$

$$+\begin{bmatrix} 0 & \hat{R}_b & 0 & 0\\ 0 & (\hat{p}_b)^\wedge \hat{R}_b & 0 & 0\\ \hat{R}_b & (\hat{v}_b)^\wedge \hat{R}_b & 0 & 0\\ 0 & 0 & Id & 0\\ 0 & 0 & 0 & Id\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} n_a\\ n_g\\ n_{wa}\\ n_{wg}\end{bmatrix}. \tag{B.12}$$

State transition matrix in discrete time

The deterministic part without biases is the interested quantity in this analysis. At time $t \in [t_{k-1}, t_k]$, the state-transition matrix is obtained in a closed form,

$$\begin{aligned}\Phi(t_k, t_{k-1})_b &= \exp\left(\begin{bmatrix} 0 & 0 & 0\\ 0 & 0 & Id\\ g^\wedge & 0 & 0\end{bmatrix}\Delta t_k\right)\\ &= \begin{bmatrix} Id & 0 & 0\\ \frac{\Delta t_k^2}{2} g^\wedge & Id & Id\Delta t_k\\ \Delta t_k g^\wedge & 0 & Id\end{bmatrix}\end{aligned} \tag{B.13}$$

where $\Delta t_k = t_k - t_{k-1}$. Therefore, the state-transition matrix for the error state is expressed as follows.

$$\begin{bmatrix} \phi_b(t_k) \\ \rho_b(t_k) \\ \nu_b(t_k) \\ \phi_o(t_k) \\ \rho_o(t_k) \end{bmatrix} = \underbrace{\begin{bmatrix} Id & 0 & 0 & 0 & 0 \\ \frac{\Delta t_k^2}{2} g^\wedge & Id & Id\Delta t_k & 0 & 0 \\ \Delta t_k g^\wedge & 0 & Id & 0 & 0 \\ 0 & 0 & 0 & Id & 0 \\ 0 & 0 & 0 & 0 & Id \end{bmatrix}}_{\Phi(t_{k-1}, t_k)} \begin{bmatrix} \phi_b(t_{k-1}) \\ \rho_b(t_{k-1}) \\ \nu_b(t_{k-1}) \\ \phi_o(t_{k-1}) \\ \rho_o(t_{k-1}) \end{bmatrix} \tag{B.14}$$

Measurement Jacobian

As in Appendix B, a robot measures the relative pose of an object at a discrete time t_l. Mathematically the relative pose measurement is,

$$Y = T_b^{-1} T_o \exp(n_o^\wedge) \tag{B.15}$$

where n_o is the white Gaussian noise. By perturbing (B.15) according to the right invariant error definition,

$$\begin{aligned} \exp(-\epsilon^\wedge) &= \hat{T}_b^{-1} \exp(\xi_b^\wedge) \exp(-\xi_o^\wedge) \hat{T}_o \exp(n_o^\wedge) \hat{T}_o^{-1} \hat{T}_b \\ &= \hat{T}_b^{-1} \exp(\xi_b^\wedge) \exp(-\xi_o^\wedge) \exp\left((\mathrm{Ad}_{\hat{T}_o} n_o)^\wedge \right) \hat{T}_b \\ &\overset{(2.32)}{\approx} \hat{T}_b^{-1} \exp\left((\xi_b - \xi_o + \mathrm{Ad}_{\hat{T}_o} n_o)^\wedge \right) \hat{T}_b \\ &= \exp\left(\left(\mathrm{Ad}_{\hat{T}_b^{-1}} (\xi_b - \xi_o + \mathrm{Ad}_{\hat{T}_o} n_o) \right)^\wedge \right) \\ \epsilon &= \mathrm{Ad}_{\hat{T}_b^{-1}} \left(-\xi_b + \xi_o - \mathrm{Ad}_{\hat{T}_o} n_o \right). \end{aligned} \tag{B.16}$$

In this expression, the adjoint is defined in (2.20). Rewriting with a matrix expression yields

$$\epsilon = \underbrace{\mathrm{Ad}_{\hat{T}_b^{-1}} \begin{bmatrix} -Id & 0 & 0 & Id & 0 \\ 0 & -Id & 0 & 0 & Id \end{bmatrix}}_{H(t_l)} \begin{bmatrix} \phi_b \\ \rho_b \\ \nu_b \\ \phi_o \\ \rho_o \end{bmatrix} - \mathrm{Ad}_{\hat{T}_b^{-1}} \mathrm{Ad}_{\hat{T}_o} n_o. \tag{B.17}$$

Observability matrix

This appendix derives a block of the observability matrix at time t_l by substituting Φ in (B.14) and H in (B.17) into (4.14).

$$\begin{aligned} \mathcal{O}_l^{\mathrm{IEKF}} &= \mathrm{Ad}_{\hat{T}_b^{-1}} \begin{bmatrix} -Id & 0 & 0 & Id & 0 \\ 0 & -Id & 0 & 0 & Id \end{bmatrix} \begin{bmatrix} Id & 0 & 0 & 0 & 0 \\ \frac{\Delta t_l^2}{2} g^\wedge & Id & Id\Delta t_l & 0 & 0 \\ \Delta t_l g^\wedge & 0 & Id & 0 & 0 \\ 0 & 0 & 0 & Id & 0 \\ 0 & 0 & 0 & 0 & Id \end{bmatrix} \\ &= \mathrm{Ad}_{\hat{T}_b^{-1}} \begin{bmatrix} -Id & 0 & 0 & Id & 0 \\ -\frac{\Delta t_l^2}{2} g^\wedge & -Id & -\Delta t_l Id & 0 & Id \end{bmatrix} \end{aligned} \tag{B.18}$$

Then, it is clear to show that the nullspace of $\mathcal{O}_l^{\mathrm{IEKF}}$ is

$$\mathcal{N}_1 = \begin{bmatrix} 0 \\ Id \\ 0 \\ 0 \\ Id \end{bmatrix}, \quad \mathcal{N}_1 = \begin{bmatrix} g \\ 0 \\ 0 \\ g \\ 0 \end{bmatrix}. \tag{B.19}$$

Bibliography

[1] D. Titterton, J. L. Weston, and J. Weston, *Strapdown inertial navigation technology.* IET, 2004.

[2] NASA. Mars Helicopter - NASA Mars. [Online]. Available: https://mars.nasa.gov/technology/helicopter/#Overview

[3] Dyson. The Dyson 360 Heurist robot | Dyson. [Online]. Available: https://www.dyson.co.uk/vacuum-cleaners/robot-vacuums/dyson-360-heurist/dyson-360-heurist-overview

[4] Skydio. Skydio 2+ | Skydio. [Online]. Available: https://www.skydio.com/skydio-2-plus

[5] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision.* Cambridge University Press, 2003.

[6] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 29, no. 6, pp. 1052–1067, Jun. 2007.

[7] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2007, pp. 3565–3572.

[8] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proceedings of IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2007, pp. 225–234.

[9] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[10] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, and A. J. Davison, "SLAM++: Simultaneous localisation and mapping at the level of objects," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1352–1359.

[11] T. D. Barfoot, *State estimation for robotics.* Cambridge, U.K.: Cambridge University Press, 2017.

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

[13] M. Li and A. I. Mourikis, "High-precision, consistent EKF-based visual-inertial odometry," *The International Journal of Robotics Research*, vol. 32, no. 6, pp. 690–711, May 2013.

[14] K. Sun, K. Mohta, B. Pfrommer, M. Watterson, S. Liu, Y. Mulgaonkar, C. J. Taylor, and V. Kumar, "Robust stereo visual inertial odometry for fast autonomous flight," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 965–972, Apr. 2018.

[15] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "OpenVINS: A research platform for visual-inertial estimation," in *Proceedings of the*

IEEE International Conference on Robotics and Automation, 2020, pp. 4666–4672.

[16] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *The International Journal of Robotics Research,* vol. 34, no. 3, pp. 314–334, Mar. 2015.

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

[18] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-Manifold Preintegration for Real-Time Visual–Inertial Odometry," *IEEE Transactions on Robotics,* vol. 33, no. 1, pp. 1–21, Feb. 2017.

[19] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics,* vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

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

[21] S. Cao, X. Lu, and S. Shen, "GVINS: Tightly coupled GNSS–visual–inertial fusion for smooth and consistent state estimation," *IEEE Transactions on Robotics,* vol. 38, no. 4, pp. 2004–2021, Aug. 2022.

[22] 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 Transactions on Robotics,* vol. 37, no. 6, pp. 1874–1890, Dec. 2021.

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

[24] ——, "Visual-inertial monocular SLAM with map reuse," *IEEE Robotics and Automation Letters,* vol. 2, no. 2, pp. 796–803, Apr. 2017.

[25] V. Usenko, N. Demmel, D. Schubert, J. Stückler, and D. Cremers, "Visual-inertial mapping with non-linear factor recovery," *IEEE Robotics and Automation Letters,* vol. 5, no. 2, pp. 422–429, Apr. 2020.

[26] S. Leutenegger, "OKVIS2: Realtime scalable visual-inertial slam with loop closure," *arXiv preprint arXiv:2202.09199,* 2022.

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

[28] L. Zhang, D. Wisth, M. Camurri, and M. Fallon, "Balancing the budget: Feature selection and tracking for multi-camera visual-inertial odometry," *IEEE Robotics and Automation Letters,* vol. 7, no. 2, pp. 1182–1189, Apr. 2022.

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

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

[31] M. Zhang, X. Zuo, Y. Chen, Y. Liu, and M. Li, "Pose Estimation for Ground Robots: On Manifold Representation, Integration, Reparameterization, and Optimization," *IEEE Transactions on Robotics,* vol. 37, no. 4, pp. 1081–1099, Aug. 2021.

[32] B. D. Lucas, T. Kanade *et al.*, "An iterative image registration technique with an application to stereo vision," in *Proceedings of International Joint Conference on Artificial Intelligence*, 1981, pp. 24–28.

[33] M. Hwangbo, J.-S. Kim, and T. Kanade, "Gyro-aided feature tracking for a moving camera: fusion, auto-calibration and GPU implementation," *The International Journal of Robotics Research,* vol. 30, no. 14, pp. 1755–1774, Dec. 2011.

[34] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *Proceedings of the IEEE International Conference on Computer Vision*, 2011, pp. 2320–2327.

[35] F. Steinbrücker, J. Sturm, and D. Cremers, "Real-time visual odometry from dense RGB-D images," in *IEEE International Conference on Computer Vision Workshops*, 2011, pp. 719–722.

[36] C. Kerl, J. Sturm, and D. Cremers, "Robust odometry estimation for RGB-D cameras," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2013, pp. 3748–3754.

[37] J. Engel, J. Sturm, and D. Cremers, "Semi-dense visual odometry for a monocular camera," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1449–1456.

[38] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *European Conference on Computer Vision*, 2014, pp. 834–849.

[39] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 40, no. 3, pp. 611–625, Mar. 2018.

[40] R. Wang, M. Schworer, and D. Cremers, "Stereo DSO: Large-scale direct sparse visual odometry with stereo cameras," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3903–3911.

[41] X. Gao, R. Wang, N. Demmel, and D. Cremers, "LDSO: Direct sparse odometry with loop closure," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018, pp. 2198–2204.

[42] L. Von Stumberg, V. Usenko, and D. Cremers, "Direct sparse visual-inertial odometry using dynamic marginalization," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2018, pp. 2510–2517.

[43] J. Zubizarreta, I. Aguinaga, and J. M. M. Montiel, "Direct sparse mapping," *IEEE Transactions on Robotics,* vol. 36, no. 4, pp. 1363–1370, Aug. 2020.

[44] L. Von Stumberg, P. Wenzel, Q. Khan, and D. Cremers, "GN-Net: The Gauss-Newton loss for multi-weather relocalization," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 890–897, Apr. 2020.

[45] L. Von Stumberg, P. Wenzel, N. Yang, and D. Cremers, "LM-Reloc: Levenberg-Marquardt based direct visual relocalization," in *International Conference on 3D Vision*, 2020, pp. 968–977.

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

[47] H. Luo, C. Pape, and E. Reithmeier, "Hybrid Monocular SLAM Using Double Window Optimization," *IEEE Robotics and Automation Letters*, Jul.

[48] S. H. Lee and J. Civera, "Loosely-coupled semi-direct monocular SLAM," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 399–406, Apr. 2019.

[49] M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Siegwart, "Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback," *The International Journal of Robotics Research*, vol. 36, no. 10, pp. 1053–1072, Sep. 2017.

[50] V. Usenko, J. Engel, J. Stückler, and D. Cremers, "Direct visual-inertial odometry with stereo cameras," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2016, pp. 1885–1892.

[51] T. Schneider, M. Dymczyk, M. Fehr, K. Egger, S. Lynen, I. Gilitschenski, and R. Siegwart, "maplab: An open framework for research in visual-inertial mapping and localization," *IEEE Robotics and Automation Letters,* vol. 3, no. 3, pp. 1418–1425, Jul. 2018.

[52] A. Cramariuc, L. Bernreiter, F. Tschopp, M. Fehr, V. Reijgwart, J. Nieto, R. Siegwart, and C. Cadena, "maplab 2.0–a modular and multi-modal mapping framework," *IEEE Robotics and Automation Letters,* vol. 8, no. 2, pp. 520–527, Feb. 2023.

[53] S. Khattak, H. Nguyen, F. Mascarich, T. Dang, and K. Alexis, "Complementary multi–modal sensor fusion for resilient robot pose estimation in subterranean environments," in *Proceedings of the International Conference on Unmanned Aircraft Systems*, 2020, pp. 1024–1029.

[54] J. McCormac, R. Clark, M. Bloesch, A. Davison, and S. Leutenegger, "Fusion++: Volumetric object-level SLAM," in *International Conference on 3D Vision*, 2018, pp. 32–41.

[55] B. Xu, W. Li, D. Tzoumanikas, M. Bloesch, A. Davison, and S. Leutenegger, "MID-Fusion: Octree-based Object-Level Multi-Instance Dynamic SLAM," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2019, pp. 5231–5237.

[56] Y. Ren, B. Xu, C. L. Choi, and S. Leutenegger, "Visual-Inertial Multi-Instance Dynamic SLAM with Object-level Relocalisation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2022, pp. 11 055–11 062.

[57] D. Frost, V. Prisacariu, and D. Murray, "Recovering Stable Scale in Monocular SLAM Using Object-Supplemented Bundle Adjustment," *IEEE Transactions on Robotics,* vol. 34, no. 3, pp. 736–747, Jun. 2018.

[58] L. Nicholson, M. Milford, and N. Sünderhauf, "QuadricSLAM: Dual Quadrics From Object Detections as Landmarks in Object-Oriented SLAM," *IEEE Robotics and Automation Letters,* vol. 4, no. 1, pp. 1–8, Jan. 2019.

[59] S. Yang and S. Scherer, "CubeSLAM: Monocular 3-D Object SLAM," *IEEE Transactions on Robotics,* vol. 35, no. 4, pp. 925–938, Aug. 2019.

[60] X. Liu, G. V. Nardari, F. C. Ojeda, Y. Tao, A. Zhou, T. Donnelly, C. Qu, S. W. Chen, R. A. Romero, C. J. Taylor, and V. Kumar, "Large-Scale Autonomous Flight With Real-Time Semantic SLAM Under Dense Forest Canopy," *IEEE Robotics and Automation Letters,* vol. 7, no. 2, pp. 5512–5519, Apr. 2022.

[61] Y. Song, Z. Zhang, J. Wu, Y. Wang, L. Zhao, and S. Huang, "A Right Invariant Extended Kalman Filter for Object Based SLAM," *IEEE Robotics and Automation Letters,* vol. 7, no. 2, pp. 1316–1323, Apr. 2022.

[62] X. Deng, A. Mousavian, Y. Xiang, F. Xia, T. Bretl, and D. Fox, "PoseRBPF: A Rao–Blackwellized Particle Filter for 6-D Object Pose Tracking," *IEEE Transactions on Robotics,* vol. 37, no. 5, pp. 1328–1342, Oct. 2021.

[63] M. Sundermeyer, Z.-C. Marton, M. Durner, M. Brucker, and R. Triebel, "Implicit 3D Orientation Learning for 6D Object Detection from RGB Images," in *European Conference on Computer Vision*, 2018, pp. 699–715.

[64] J. Fu, Q. Huang, K. Doherty, Y. Wang, and J. J. Leonard, "A Multi-Hypothesis Approach to Pose Ambiguity in Object-Based SLAM," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021, pp. 7639–7646.

[65] E. Olson and P. Agarwal, "Inference on networks of mixtures for robust robot mapping," *The International Journal of Robotics Research,* vol. 32, no. 7, pp. 826–840, Jun. 2013.

[66] Z. Lu, Q. Huang, K. Doherty, and J. J. Leonard, "Consensus-Informed Optimization Over Mixtures for Ambiguity-Aware Object SLAM," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021, pp. 5432–5439.

[67] N. Merrill, Y. Guo, X. Zuo, X. Huang, S. Leutenegger, X. Peng, L. Ren, and G. Huang, "Symmetry and Uncertainty-Aware Object SLAM for 6DoF Object Pose Estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14 901–14 910.

[68] J. Sola, J. Deray, and D. Atchuthan, "A micro lie theory for state estimation in robotics," *arXiv preprint arXiv:1812.01537,* 2018.

[69] J. Ćesić, I. Marković, and I. Petrović, "Mixture reduction on matrix Lie groups," *IEEE Signal Processing Letters,* vol. 24, no. 11, pp. 1719–1723, Nov. 2017.

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

[71] A. R. Runnalls, "Kullback-Leibler approach to Gaussian mixture reduction," *IEEE Transactions on Aerospace and Electronic Systems,* vol. 43, no. 3, pp. 989–999, Jul. 2007.

[72] D. Alspach and H. Sorenson, "Nonlinear Bayesian estimation using Gaussian sum approximations," *IEEE Transactions on Automatic Control,* vol. 17, no. 4, pp. 439–448, Aug. 1972.

[73] D. S. Bayard, D. T. Conway, R. Brockers, J. H. Delaune, L. H. Matthies, H. F. Grip, G. B. Merewether, T. L. Brown, and A. M. San Martin, "Vision-Based Navigation for the NASA Mars Helicopter," in *AIAA Scitech 2019 Forum*, 2019, pp. 1411–1432.

[74] G. Huang, "Visual-inertial navigation: A concise review," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2019, pp. 9572–9582.

[75] D. Scaramuzza and F. Fraundorfer, "Visual odometry [tutorial]," *IEEE Robotics & Automation Magazine,* vol. 18, no. 4, pp. 80–92, Dec. 2011.

[76] M. Irani and P. Anandan, "About direct methods," in *Proceedings of International Workshop on Vision Algorithms*, 1999, pp. 267–277.

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

[78] M. Brossard, A. Barrau, P. Chauchat, and S. Bonnabel, "Associating Uncertainty to Extended Poses for on Lie Group IMU Preintegration with Rotating Earth," *IEEE Transactions on Robotics,* vol. 38, no. 2, pp. 998–1015, Apr. 2022.

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

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

[81] D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches.* Hoboken, NJ, USA: John Wiley & Sons, 2006.

[82] 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 Robotics and Automation Letters,* vol. 6, no. 2, pp. 1343–1350, Apr. 2021.

[83] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and service robotics*, 2018, pp. 621–635.

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

[85] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 1280–1286.

[86] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry," in *Proceedings of the IEEE/RSJ*

International Conference on Intelligent Robots and Systems, 2018, pp. 7244–7251.

[87] S. Bonnabel, "Left-invariant extended Kalman filter and attitude estimation," in *Proceedings of the 46th IEEE Conference on Decision and Control*, 2007, pp. 1027–1032.

[88] A. W. Long, K. C. Wolfe, M. J. Mashner, and G. S. Chirikjian, "The Banana Distribution is Gaussian: A Localization Study with Exponential Coordinates," *Robotics: Science and Systems VIII,* vol. 265, 2013.

[89] S. Heo and C. G. Park, "Consistent EKF-Based Visual-Inertial Odometry on Matrix Lie Group," *IEEE Sensors Journal,* vol. 18, no. 9, pp. 3780–3788, May 2018.

[90] M. Brossard, A. Barrau, and S. Bonnabel, "Exploiting Symmetries to Design EKFs With Consistency Properties for Navigation and SLAM," *IEEE Sensors Journal,* vol. 19, no. 4, pp. 1572–1579, Feb. 2019.

[91] R. Hartley, M. Ghaffari, R. M. Eustice, and J. W. Grizzle, "Contact-aided invariant extended Kalman filtering for robot state estimation," *The International Journal of Robotics Research,* vol. 39, no. 4, pp. 402–430, Mar. 2020.

[92] A. Barrau and S. Bonnabel, "An EKF-SLAM algorithm with consistency properties," *arXiv preprint arXiv:1510.06263,* 2015.

[93] D. Gálvez-López, M. Salas, J. D. Tardós, and J. Montiel, "Real-time monocular object SLAM," *Robotics and Autonomous Systems,* vol. 75, pp. 435–449, Jan. 2016.

[94] S. Lin, J. Wang, M. Xu, H. Zhao, and Z. Chen, "Topology Aware Object-Level Semantic Mapping Towards More Robust Loop Closure," *IEEE Robotics and Automation Letters,* vol. 6, no. 4, pp. 7041–7048, Oct. 2021.

[95] J. H. Jung, J. Cha, J. Y. Chung, T. I. Kim, M. H. Seo, S. Y. Park, J. Y. Yeo, and C. G. Park, "Monocular Visual-Inertial-Wheel Odometry Using Low-Grade IMU in Urban Areas," *IEEE Transactions on Intelligent Transportation Systems,* vol. 23, no. 2, pp. 925–938, Feb. 2022.

[96] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *The International Journal of Robotics Research,* vol. 32, no. 11, pp. 1231–1237, Sep. 2013.

[97] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka, "3D Bounding Box Estimation Using Deep Learning and Geometry," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7074–7082.

[98] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv preprint arXiv:1804.02767,* 2018.

[99] M. Shan, Q. Feng, and N. Atanasov, "OrcVIO: Object residual constrained Visual-Inertial Odometry," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020, pp. 5104–5111.

[100] T. Schon, F. Gustafsson, and P.-J. Nordlund, "Marginalized particle filters for mixed linear/nonlinear state-space models," *IEEE Transactions on Signal Processing,* vol. 53, no. 7, pp. 2279–2289, Jul. 2005.

[101] J. Park, Y.-G. Park, and C. G. Park, "Parameter estimation of radar noise model for terrain referenced navigation using a new EM initialization method," *IEEE Transactions on Aerospace and Electronic Systems,* vol. 56, no. 1, pp. 107–112, Feb. 2020.

[102] B.-N. Vo and W.-K. Ma, "The Gaussian mixture probability hypothesis density filter," *IEEE Transactions on Signal Processing,* vol. 54, no. 11, pp. 4091–4104, Nov. 2006.

[103] K. J. Doherty, D. P. Baxter, E. Schneeweiss, and J. J. Leonard, "Probabilistic data association via mixture models for robust semantic SLAM," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2020, pp. 1098–1104.

[104] C. O' Meadhra, W. Tabib, and N. Michael, "Variable Resolution Occupancy Mapping Using Gaussian Mixture Models," *IEEE Robotics and Automation Letters,* vol. 4, no. 2, pp. 2015–2022, Apr. 2019.

[105] H. Huang, H. Ye, Y. Sun, and M. Liu, "GMMLoc: Structure Consistent Visual Localization With Gaussian Mixture Models," *IEEE Robotics and Automation Letters,* vol. 5, no. 4, pp. 5043–5050, Oct. 2020.

[106] C. Qian, C. Song, S. Li, Q. Chen, and J. Guo, "Algorithm of Gaussian Sum Filter Based on SGQF for Nonlinear Non-Gaussian Models," *International Journal of Control, Automation and Systems,* vol. 19, no. 8, pp. 2830–2841, Aug. 2021.

[107] A. Barrau and S. Bonnabel, "Invariant particle filtering with application to localization," in *Proceedings of the IEEE Conference on Decision and Control*, 2014, pp. 5599–5605.

[108] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes," *Robotics: Science and Systems*, 2018.

[109] Y. Labbé, J. Carpentier, M. Aubry, and J. Sivic, "CosyPose: Consistent multi-view multi-object 6D pose estimation," in *European Conference on Computer Vision*, 2020, pp. 574–591.

[110] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The YCB object and model set: Towards common benchmarks for manipulation research," in *Proceedings of the International Conference on Advanced Robotics*, 2015, pp. 510–517.

[111] T. Lee, Y. Jang, and H. J. Kim, "Object-based SLAM utilizing unambiguous pose parameters considering general symmetry types," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2023.

[112] M.-H. Jeon, J. Kim, J.-H. Ryu, and A. Kim, "Ambiguity-Aware Multi-Object Pose Optimization for Visually-Assisted Robot Manipulation," *IEEE Robotics and Automation Letters*, vol. 8, no. 1, pp. 137–144, 2022.

[113] J. Rydell, M. Tulldahl, E. Bilock, L. Axelsson, and P. Köhler, "Autonomous UAV-based forest mapping below the canopy," in *Proceedings of the IEEE/ION Position, Location and Navigation Symposium*, 2020, pp. 112–117.

[114] S. Heo, "EKF-based Visual-Inertial Navigation on Matrix Lie group with Improved Consistency," Ph.D. dissertation, Department of Aerospace and Mechanical Engineering, Seoul National University, Seoul, Republic of Korea, 2018.

국문초록

영상관성항법 시스템은 영상 및 관성 측정치를 기반으로 항체의 상태변수, 즉 위치, 자세 그리고 주변 지도를 추정한다. 안전한 자율 시스템을 위해서는 정확하고 실시간성이 보장되는 영상관성항법과 같은 상태변수 추정기가 필수적이다. 최근 많은 연구에도 불구하고 센서 성능이 저하되는 도전적인 상황에서 고장에 치명적인 시스템에 영상관성항법을 적용하기에 어려움이 있었다. 이를 위해서는 어떠한 환경에서도 안정적인 추정치를 출력할 수 있게 고장에 대해 강건해야 하며 고장이 발생하더라도 이를 자동으로 감지하여 회복할 수 있어야 한다.

본 논문에서는 강건한 영상관성항법 시스템을 설계하고자 세 가지의 연구 원칙을 제시한다. 첫번째로 해당 시스템은 상태변수에 대한 추정치 뿐만 아니라 센서의 불확실성을 고려하는 유효한 추정 신뢰도를 출력해야 한다. 추정된 불확실성은 그 자체로서 고장 감지를 위해 활용될 수 있고 경로 계획과 되먹임 제어와 같은 후속 작업에도 활용될 수 있다. 두번째로 상태변수 공간은 강체 운동의 불확실성을 전파하기에 가장 알맞은 리그룹에서 모델링되어야 한다. 마지막으로, 개발된 시스템은 겉보기 및 시점 변화에 강건한 의미론적 물체를 항법 측정치로서 활용할 수 있어야 한다.

앞서 제시한 연구 원칙에 입각하여 본 논문에서는 최적 이미지 기울기, 물체기반 동시적 위치 추정 및 지도 작성 (SLAM) 그리고 리-행렬군에서의 정규분포 융합 방법을 제시한다. 최적 이미지 기울기는 이미지 영역에서 투영 불확실성에 대하여 선형화 오차 제곱의 기대값을 최소화 하게 설계되었다. 물체기반 SLAM에서는 비가관측 공간이 선형화 지점에 의존적이지 않음을 해석적으로 증명하여 추정기가 유효한 불확실성을 출력함을 보였다. 또한, 리-행렬군 상의 정규분포를 융합하는 방법을 제시하고 이를 대칭성이 포함된 물체기반 SLAM에 적용하여

측정치의 모호성을 해결하였다. 최종적으로 본 논문에서 제안하는 방법들을 통합하여 물체기반 영상관성시스템을 제안한다.

본 논문에서는 시뮬레이션 및 실제 실험을 통해 제안한 방법들의 타당성을 검증하였다. 영상관성 오도메트리에서 기존 방법은 최대 3 m/s 속도 오차에서 수렴이 실패한 반면 제안한 최적 기울기를 적용할 경우 추정기가 성공적으로 수렴함을 보였다. 물체기반 SLAM에서 기존의 접근 방법은 추정치의 비일관성 문제 때문에 중력에 대한 회전 방향에 대해 유효한 신뢰도를 추정하지 못하였다. 하지만 본 연구에서는 물체기반 SLAM을 리그룹에서 모델링 하였고 이를 통해 유효한 추정 신뢰도를 추정할 수 있음을 보였다. 또한, 물체기반 SLAM의 대칭 모호성을 해결함으로써 10,000장의 이미지에 대해 6자유도 포즈 검출기보다 평균적으로 자세 오차를 49.5% 감소시켰다. 마지막으로 물체기반 영상관성항법 시스템을 제안하여 방 크기의 실내 환경에서 cm 수준의 항법 및 지도 작성이 가능함을 보였다.

주요어: 영상관성항법, 상태추정, 리-행렬군, 동시적 위치추정 및 지도작성
학번: 2019-32429