A Comprehensive Introduction of Visual-Inertial Navigation

Yangyang NING

March 2022

Abstract

In this article, a tutorial introduction to visual-inertial navigation(VIN) is presented. Visual and inertial perception are two complementary sensing modalities. Cameras and inertial measurement units (IMU) are the corresponding sensors for these two modalities. The low cost and light weight of camera-IMU sensor combinations make them ubiquitous in robotic navigation. Visual-inertial Navigation is a state estimation problem, that estimates the ego-motion and local environment of the sensor platform. This paper presents visual-inertial navigation in the classical state estimation framework, first illustrating the estimation problem in terms of state variables and system models, including related quantities representations (Parameterizations), IMU dynamic and camera measurement models, and corresponding general probabilistic graphical models (Factor Graph). Secondly, we investigate the existing model-based estimation methodologies, these involve filter-based and optimization-based frameworks and related on-manifold operations. We also discuss the calibration of some relevant parameters, also initialization of state of interest in optimization-based frameworks. Then the evaluation and improvement of VIN in terms of accuracy, efficiency, and robustness are discussed. Finally, we briefly mention the recent development of learning-based methods that may become alternatives to traditional model-based methods.

1 Introduction

Environment and ego-motion perception is critically important for navigation. Motion perception in biology offers a more intuitive perspective, human use various sensory modalities to feel self-motion and surroundings, especially combining motion and balance from the inner ear, joint position, and visual information from the eyes to obtain a virtual sense of movement, called kinesthesia[1]. Instead, motion perception of robots is provided through the use of navigation and positioning techniques including dead reckoning methods (using inertial sensors, pedometer, wheel encoder, magnetometer, and gyrocompass, etc.) and position fixing methods (using global navigation satellite system (GNSS), ultra-wideband (UWB), acoustic ranging, lidar, and cameras, etc.)[2]. In this work, we focus on two types of sensory modalities: visual and inertial. Visual-inertial navigation is a typical state estimation problem, that estimates ego-motion and local map given measurement data from the camera and IMU. Ego-motion can be quantitatively described as robot states normally involving translation, rotation, and velocity, evenly higher time derivatives (acceleration, etc.) in certain reference coordinates over time[3]. In terms of local map representation, VIN usually quantifies the geometric information of local map in the form of 3D point clouds, meshes, voxels, or signed distance functions (SDF)[4] with different capabilities and efficiencies.

For robot navigation, VIN provides both robot's current state and local map, which are essential for path planning, obstacle avoidance, and real-time control. As a popular navigation technique, VIN greatly contributes to the general guidance, navigation, and control (GNC) systems shown in Figure 1. In particular, since VIN uses onboard sensors, such sensory modalities are critically essential in the absence of external positioning systems like GPS, UWB, and visual motion capture systems. As such, VIN is widely used for indoor localization, underwater exploration, city reconstruction, and search-and-rescue.

Compared to some GPS-free navigation system that uses only cameras and lidars, visual-inertial fusion improves accuracy, robustness, and efficiency of estimation. Visual-only systems like visual simultaneous localization and mapping(V-SLAM) require static scenes, sufficient illumination, Lambertian textured surface (no transparent or reflective), rich texture, and scene overlap. However, as a proprioceptive sensor, IMU is not limited by the above conditions by its internal mechanism that reads local linear acceleration and angular velocity. Micro-electro-mechanical-system (MEMS)

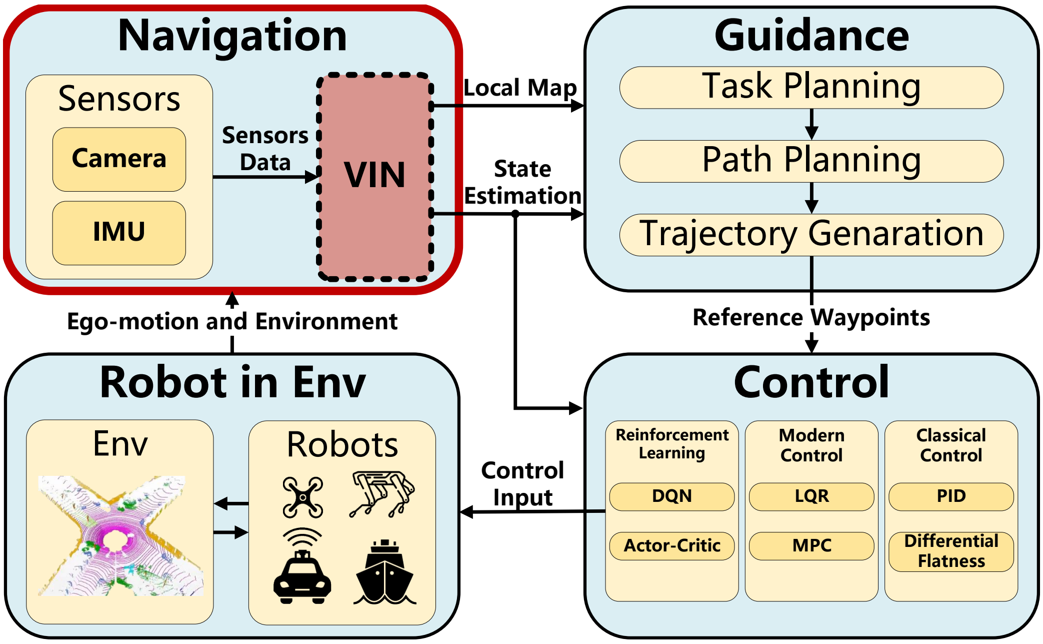


Figure 1: VIN in general guidance, navigation, and control (GNC) framework

IMU, in particular, has become ubiquitous in many robotics applications such as micro aerial vehicles (MAVs)[5][6], autonomous underwater vehicles (AUVs)[7], autonomous ground vehicles (AGVs)[8], quadruped robot[9], spacecraft[10], and mobile devices[11] due to its low cost and lightweight. However, IMU uses the dead reckoning technique for estimation which leads to drift over time caused by integrating the bias and noisy measurements. Two types of sensory modalities compensate for each other, IMU won't affect by visual extreme conditions including motion blur, low texture, and illumination change, and provides high rate measurements, whereas cameras are more accurate at slow motion with no drift in static scenes.

2 States and Models in VIN

As a typical state estimation problem, VIN is to estimate the robot motion and local map given IMU kinematics model and camera measurement model. The robot's current ego-motion can be characterized as the pose at the current moment and its time derivatives (velocity and acceleration). The metric information of a local map can be represented as 3D points, lines, or surfaces distributed in 3D space, where semantic information acted as "category signatures" tagged on these metric representations. In this work, we focus on metric information and explore works that use 3D points as local map landmarks. Poses and 3D landmarks are normally of interest to VIN state estimation. In VIN, some supporting parameters are also needed for estimating poses and 3D landmarks, including some time-varying quantities like velocity, IMU bias, gravity direction in the local frame, and some time-invariant quantities like camera intrinsic, and camera-IMU extrinsic, etc. A typical state estimation model is shown in Equation 1a and 1b as dynamic and measurement models respectively. In many VIN works[12][13][14], systems are modeled as hybrid systems[15] with continuous-time dynamic and discrete-time measurements.

$$\textbf{Continuous-time Dynamic Model:} \qquad \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \tilde{\mathbf{u}}, \mathbf{w}) \tag{1a}$$

$$\textbf{Discrete-time Measurement Model:} \qquad \tilde{\mathbf{z}}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v} \tag{1b}$$

In VIN, poses, landmarks, time-varying quantities(velocity, IMU bias, etc.), and sometimes time-invariant quantities (online calibration) are contained in state vector $\mathbf{x}$, the dynamic function $\mathbf{f}(\cdot)$ is represented by IMU kinematics with linear acceleration and angular velocity measurement as control input $\tilde{\mathbf{u}}$. The measurement model $\mathbf{h}(\cdot)$ is normally a camera measurement model with 2D image features as output $\tilde{\mathbf{z}}_k$. $\mathbf{w}$ and $\mathbf{v}$ are process noise (noises from IMU measurements) and measurement noise (noises from camera measurements) respectively. In this section, we first define states and reference frames in VIN problem and discuss the parametrization of quantities involved in VIN. Lastly, we briefly derive IMU kinematics model and camera measurement model and relate them as factors in the factor graph for estimation methods clarification in Section 3.

2.1 Reference Frames and States in VIN

Geometric quantities such as poses and landmarks are relative quantities that depend on the frame of reference (coordinate). It is critical to firmly identify different frames. These include a *fixed* world frame $\boldsymbol{\mathcal{F}_W}$ (global frame, map frame, or inertial frame), a *moving* body frame $\boldsymbol{\mathcal{F}_B}$ (robot frame or vehicle frame), some sensors frames (rigidly installed on the robot) like camera frames $\boldsymbol{\mathcal{F}_C}$ and IMU frames $\boldsymbol{\mathcal{F}_I}$. In many VIN cases, the IMU frame is coincident with the body frame as shown in Figure 2, where X, Y, and Z axis are in red, green, and blue respectively.

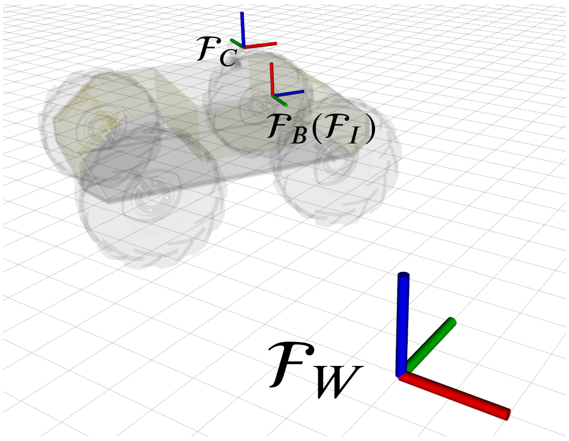


Figure 2: Fixed world frame and moving body frame with rigidly mounted sensor frames

Normally, the final result of VIN including a pose track and a local map is expressed in an arbitrary stationary world frame. The pose of a robot at each time-step is expressed in translation and rotation $({}_W\boldsymbol{t}_{WB}, \boldsymbol{R}_{WB})$ concerning the transformation between body frame and world frame. In world-centric framework, The 3D landmarks are also expressed in world frame. In many VIN practices, the world frame is set to the first camera frame's corresponding IMU frame $\boldsymbol{\mathcal{F}_{I_0}}$. This allows zero initial pose uncertainty, which reduces the level of uncertainty and increases the consistency of the estimate[16][17].

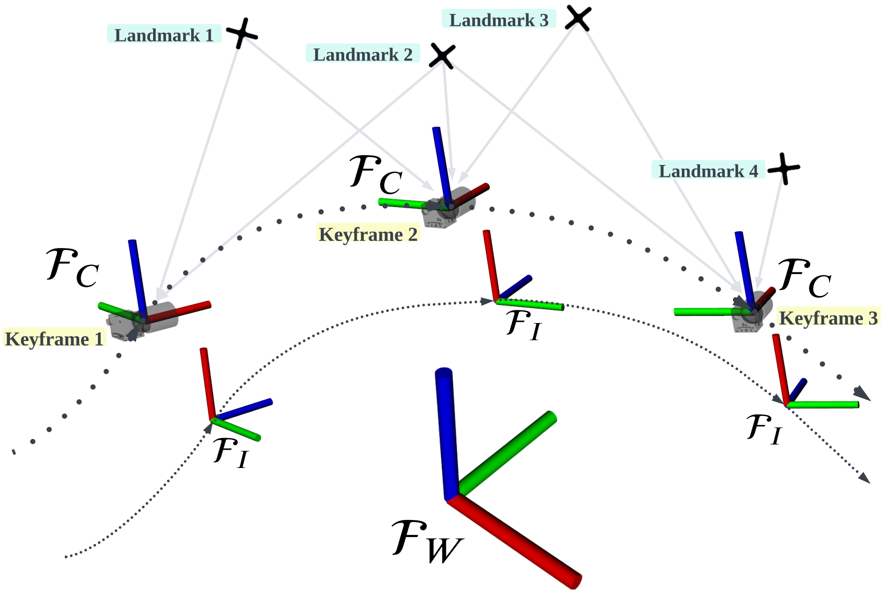
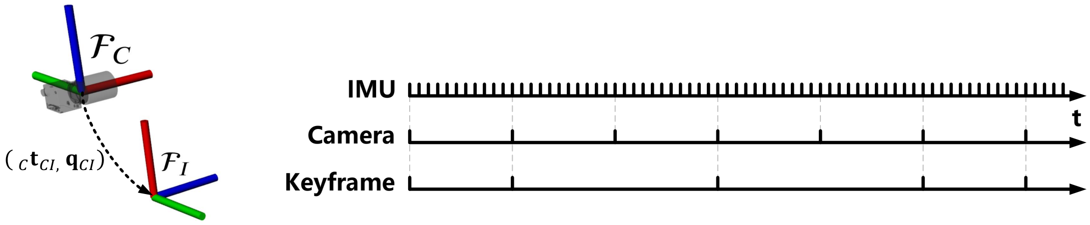


Figure 3: Landmarks and Camera-IMU poses along discrete measurement trajectory

The visual measurements of a moving camera are shown in a sequence of image frames that are expressed in a 2D image coordinate. This 2D image coordinate is parallel to the y-z plane of the 3D camera frame $\boldsymbol{\mathcal{F}_C}$, which is rigidly mounted on the robot, as does the IMU frame $\boldsymbol{\mathcal{F}_I}$. Figure 3 shows sparse landmark features, and discrete visual-inertial measurements along trajectories, whose sampling rate is indicated by the density of dashed lines. IMU and camera, as a multisensor system, exist temporal and spatial differences. The temporal difference is mainly reflected in their different measurement frequency and latency, and the spatial difference is caused by the fixed installation displacement of these two sensors (camera-IMU extrinsic as shown in Figure 4a). The measurement sampling rate of the IMU (100-1000Hz) is much faster than that of the camera (10-80Hz). Due to the difference in sensors' latency, there exists temporal misalignment (time offset) between IMU and camera measurements. This temporal misalignment can be resolved by temporal calibration[18][19][20][21] that estimates the time offset or hardware synchronization[22], Figure 4b shows the periodic time-synchronized IMU and camera measurement timestamps with possible keyframe selection. Camera-IMU extrinsic can be identified by offline and online spatial calibration, which we discuss later in this paper. For now, we assume IMU and camera data are time-synchronized and camera-IMU extrinsic is known.



(a) Camera-IMU Extrinsic

(b) Time-synchronized IMU and camera measurement rate with possible keyframes selection

Figure 4: Spatial and Temporal difference between Camera and IMU

The states of interest in VIN normally consist of poses and landmarks. In structureless visual-inertial estimation, landmarks are excluded from the state of interest. For filter-based estimation, the estimated state usually contains only one pose at the current moment, while in an optimization-based framework, a fixed or incremental number of poses will be involved in the state. In this work, we adopt Carlone *et al.*[23] convention, which classifies *target* variables and *support* variables in the state variables, where target variables are normally of interest variables and support variables are essential for estimation of target variables. In VIN, target variables are the poses (translation and rotation) of a robot. Support variables contain the velocity and bias of IMU. Landmarks can be target or support variables depending on the focus of estimation. For example, in visual-inertial odometry (VIO), the current state of the robot including pose and its time derivatives are the most concerned, whereas, in visual-inertial simultaneous localization and mapping (VI-SLAM), accurate estimation of both landmarks and poses throughout the trajectory is required.

Note that the IMU dynamic model and the camera model are based on IMU and camera measurement respectively, in which case the maximum state estimation rate of VIN is bounded by the sensor measurement rate of the camera and IMU. A new state estimate can be computed whenever a new IMU or camera measurement arrives. However, the frequency of IMU measurements is high, which will result in a high computational burden and the prediction of estimation is prone to drift before a new camera measurement arrives. In this case, the poses' estimate of the robot is normally considered with the corresponding camera frames instead of all IMU frames, which means the poses in states are usually poses of camera frames' corresponding IMU frames (assume camera-imu extrinsic is known).

For generality, we define active states in time k in VIN as,

$$\mathcal{X}_k = \{\mathcal{I}_k, \mathcal{L}_k\}$$

where states in time k consist of a set of IMU frames $\mathcal{I}_k$ and a set of 3D landmarks $\mathcal{L}_k$. We assume all camera frames have corresponding time-synchronized IMU frames. The selections and number of frames are different depending on estimation frameworks, we here show some of the typical cases of frames' composition.

- Filter-based framework:

  Only contain one most recent camera frame, or the latest selected keyframe, or the last IMU frame in time k.

  $$\mathcal{I}_k = \{\mathbf{F}_k\}, \text{ or } \{\mathbf{K}_k\}, \text{ or } \{\mathbf{I}_k\}$$

  where $\mathbf{F}_k$ denotes the last camera frame corresponded IMU frame in time k, $\mathbf{K}_k$ denotes the last camera keyframe corresponded IMU frame in time k, $\mathbf{I}_k$ denotes the last IMU frame in time k.

- Fixed-Lag Smoother framework:

  Normally contain a fixed sliding window of recent frames and consecutive keyframes in time k.

  $$\mathcal{I}_k = \{\mathbf{K}_{k-m}, \cdots, \mathbf{K}_k, \mathbf{F}_{k-n}, \cdots, \mathbf{F}_k\}$$

  where $\{\mathbf{K}_{k-m}, \cdots, \mathbf{K}_k\}$ denotes $m+1$ recent consecutive keyframes corresponded IMU frames in time k, $\{\mathbf{F}_{k-n}, \cdots, \mathbf{F}_k\}$ denotes $n+1$ recent camera frames corresponded IMU frames in time k. There is no overlap between keyframes and recent frames, a new camera frame will be selected as a new keyframe if it passes the keyframe selection test, otherwise, it will be a new recent frame.

- Full batch optimization-based framework:

  Contain the entire keyframes in time k that incremented over time.

  $$\mathcal{I}_k = \{\mathbf{K}_1, \cdots, \mathbf{K}_k\}$$

- Covisibility-based framework:

  Contain frames that share sufficient common features with the most recent keyframe in time k (covisibility graph). This may involve past frames selected independently of time.

  $$\mathcal{I}_k = \left\{\mathbf{K}_k, \mathbf{K}^k_{c_1}, \cdots, \mathbf{K}^k_{c_o}\right\}$$

  where $\mathbf{K}^k_{c_1}, \cdots, \mathbf{K}^k_{c_o}$ denote a set of camera frames corresponded IMU frames in the covisibility graph of the last keyframe $\mathbf{K}_k$.

Note that the *keyframes* mentioned above are carefully selected from camera frames that exhibit good feature tracking quality and significant parallax with previous keyframes. The purpose of selecting keyframes from camera frames is to reduce the uncertainty of initialization and to improve the efficiency of graph-based optimization by preserving the sparsity that only a subset of camera frames is maintained.

Recent IMU frames $\mathbf{I}$, camera frames' corresponding IMU frames $\mathbf{F}$, and camera keyframes' corresponding IMU frames $\mathbf{K}$ contain pose information of the IMU frame and its direct support variables. In VIN, these IMU frames can be represented in IMU state $\mathbf{x}_I$, where camera-IMU extrinsic $({}_C\mathbf{t}_{CI}, \mathbf{q}_{CI})$ is assumed given and camera-IMU measurements are time-synchronized. For visual-inertial state estimation with a monocular camera, an IMU state is generally defined as followed,

$$\mathbf{x}_I = \left[{}_W\mathbf{t}^\intercal_{WI}, \mathbf{q}^\intercal_{WI}, {}_W\mathbf{v}^\intercal_{WI}, \mathbf{b}^\intercal_g, \mathbf{b}^\intercal_a\right]^\intercal$$

where

${}_W\mathbf{t}_{WI} \in \mathbb{R}^3$: Position of IMU frame origin wrt. world frame expressed in world frame
$\mathbf{q}_{WI} \in \mathbb{S}^3$: Rotation from IMU frame to world frame in Hamiltonian unit quaternion
${}_W\mathbf{v}_{WI} \in \mathbb{R}^3$: Velocity of IMU frame wrt. world frame expressed in world frame
$\mathbf{b}_g \in \mathbb{R}^3$: Gyro bias of IMU frame wrt. world frame expressed in IMU frame
$\mathbf{b}_a \in \mathbb{R}^3$: Accelerometer bias of IMU frame wrt. world frame expressed in IMU frame

The set of 3D landmarks $\mathcal{L}_k$ in time k contains all of the landmarks observed in the camera frames corresponding to the IMU frames $\mathcal{I}_k$. For generality, we define

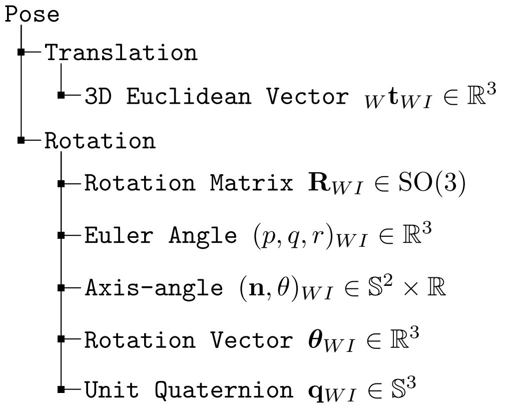
$$\mathcal{L}_k = \left\{\mathbf{L}^k_1, \cdots, \mathbf{L}^k_{l_k}\right\}$$

where $\mathbf{L}^k_j$ $j = 1, \cdots, l_k$ denotes a single 3D landmark, l_k denotes the number of active 3D landmarks in $\mathcal{I}_k$. Each 3D landmark normally contains the position and indexing information. Position information can be straightforwardly presented in 3D Euclidean space, more parameterizations are presented in Section 2.2. The indexing information for a 3D landmark contains a "serial number" that corresponds to the 2D feature in the set of camera frames that observed the landmark, and the indices of this set of camera frames. This "serial number" is a feature descriptor like ORB[24] or BRIEF[25], etc. in descriptor-based data association.

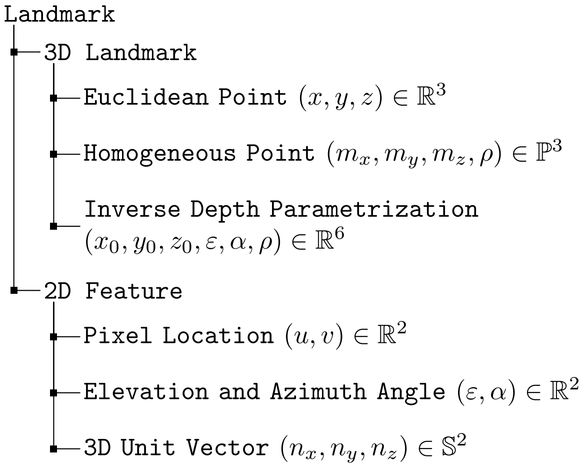
2.2 Related Quantities in VIN and thier Parametrizations

In this section, we discuss all relevant quantities involved in VIN estimation and the properties of their typical representations (parametrization). Pose and landmark is the most concerned quantities in VIN. The pose of frames consists of translation and rotation. We adopt Furgale's conversion[26], where coordinates of 3-vectors in VIN like translation, velocity, and acceleration need three elements of decoration to clearly specify, which contain a physical quantity from one frame to another in addition to being represented in a specific reference frame. For rotation, two elements are required to represent the origin and destination frame. Assume for the moment that poses and landmarks are all expressed in the world frame.

Parametrizations define a set of representations of the same quantity, like a 2D coordinate can be represented in both Polar and Cartesian Coordinates. However, some limitations come with certain parametrization, like singularities and constraints. The translation lives in vector space with no singularity and constraints. However, the rotation has several forms, including rotation matrix, Euler angle, axis-angle, rotation vector, and unit quaternion shown in the below notation tree. The summary of their properties is shown in Table 1. Unit quaternion compared to other parametrizations, contains fewer constraints and no singularity becoming a "standard" way to represent rotation. Since rotation is not in vector space but in a smooth manifold (Lie group), conventional additive and subtractive operations in vector space cannot be used. In this case, in order to use canonical filtering and optimization methods built on vector space operations, it is necessary to transform the quantities in Lie groups into operable vector spaces (Lie algebras). We will discuss this later in Section 3.3.



Pose
- Translation
  - 3D Euclidean Vector ${}_W\mathbf{t}_{WI} \in \mathbb{R}^3$
- Rotation
  - Rotation Matrix $\mathbf{R}_{WI} \in \mathrm{SO}(3)$
  - Euler Angle $(p, q, r)_{WI} \in \mathbb{R}^3$
  - Axis-angle $(\mathbf{n}, \theta)_{WI} \in \mathbb{S}^2 \times \mathbb{R}$
  - Rotation Vector $\boldsymbol{\theta}_{WI} \in \mathbb{R}^3$
  - Unit Quaternion $\mathbf{q}_{WI} \in \mathbb{S}^3$



Landmark
- 3D Landmark
  - Euclidean Point $(x, y, z) \in \mathbb{R}^3$
  - Homogeneous Point $(m_x, m_y, m_z, \rho) \in \mathbb{P}^3$
  - Inverse Depth Parametrization $(x_0, y_0, z_0, \varepsilon, \alpha, \rho) \in \mathbb{R}^6$
- 2D Feature
  - Pixel Location $(u, v) \in \mathbb{R}^2$
  - Elevation and Azimuth Angle $(\varepsilon, \alpha) \in \mathbb{R}^2$
  - 3D Unit Vector $(n_x, n_y, n_z) \in \mathbb{S}^2$

	Dimension	Constraints	Direct Composition	Singularity
Rotation matrix	9	6	Matrix Multiplication	None
Euler angle	3	0	None	Gimbal lock
Axis-angle	4	1	Compositable over Single Axis	$\theta = 0$ and π
Rotation Vector	3	0	Compositable over Single Axis	None
Unit Quaternion	4	1	Quaternion Multiplication	None

Table 1: Rotation Parametrizations

Point landmarks in VIN exist in 3D space and are observed in their 2D projection. Geometrically, the camera is a bearing sensor, and distances to landmarks cannot be measured with a single frame. 3D landmarks in Euclidean space cannot be initialized at the first observation. In this case, the concept of undelayed landmark initialization (ULI) is introduced by Solà[27], which represents the unmeasured DOF by a Gaussian prior. This involves two important ULI landmark parametrizations: Homogeneous points (HP) and inverse depth parametrization (IDP). HP lives in 4-vector projective space, where (m_x, m_y, m_z) denotes the 3D vector toward the landmark and the scalar ρ is proportional to inverse distance. IDP is introduced by Montiel *et al.* in [28] and extended by Civera in [29]. IDP maintains relatively low non-linearity in measurement equation at low and high feature parallax compared to Euclidean parametrization. Compared to HP, IDP adds a predicted first-seen frame as an anchor, which allows to represent landmark uncertainty locally, thus reducing the accumulated linearization error[30]. In [31], Solà *et al.* show that IDP establishes better consistency in monocular EKF-SLAM compared to HP. 2D features are originally measured in the image coordinate with the pixel unit. However, in pixel coordinate parameterization, the same pixel coordinates represent different bearing directions in different cameras (perspective, refraction, or catadioptric cameras). For generality, the 3D bearing unit vector is used for representing the 2D features in the camera frame, it has no singular configurations compared to elevation and azimuth angle (north and south poles).

Other time-varying quantities like velocity and acceleration support the estimation of poses, whereas IMU bias, gravity, and earth rotation effect affect the estimate of velocity and acceleration. In addition, in the case of the monocular camera, scale ambiguity also needs to be considered. Local gravity and earth angular velocity expressed in IMU frame are fixed-length vectors ${}_I\mathbf{g} = g {}_I\hat{\mathbf{g}}$ and ${}_I\boldsymbol{\omega}_E = \omega_\epsilon {}_I\hat{\boldsymbol{\omega}}_E$, where $g \approx 9.81\text{m/s}^2$ and $\omega_\epsilon \approx 7.29 \times 10^{-5}$rad/s with 3D unit direction vectors ${}_I\hat{\mathbf{g}}$ and ${}_I\hat{\boldsymbol{\omega}}_E$, symbol $\hat{\{\cdot\}}$ indicates unit vector. All other quantities live in 3D vector space. Note angular velocity and linear acceleration are measured by gyroscope and accelerometer respectively. Earth rotation effect is measurable when using high-end IMUs, but is negligible in many VINs using low-cost IMUs. For simplicity, we neglect it in this tutorial, for considering this effect, we refer readers to [12][3].

Velocity and Acceleration

- Linear Velocity ${}_W\mathbf{v}_{WI} \in \mathbb{R}^3$
- Angular Velocity ${}_I\boldsymbol{\omega}_{WI} \in \mathbb{R}^3$
- Linear Acceleration ${}_I\boldsymbol{a}_{WI} \in \mathbb{R}^3$

IMU Bias, Gravity, Scale and Earth Rate

- Gyroscope Bias $\mathbf{b}_g \in \mathbb{R}^3$
- Accelerometer Bias $\mathbf{b}_a \in \mathbb{R}^3$
- Local Gravity Direction ${}_I\hat{\mathbf{g}} \in \mathbb{S}^2$
- Monocular Scale $s \in \mathbb{R}$
- Local Earth Rate Direction ${}_I\hat{\boldsymbol{\omega}}_E \in \mathbb{S}^2$

Besides time-varying quantities that must be estimated online, there are time-invariant quantities that can be estimated offline or online (out-of-box operation). These quantities are associated with the sensors properties of the camera and IMU, which can be classified into the camera, IMU, and camera-IMU related quantities shown in the below notation tree. For camera-related quantities, camera intrinsic should be considered since images are rendered in image coordinates in pixel units with lens distortion. Some VIN systems[32][33][34] rely on the photometric intensity of feature pixels (direct methods), in which case time-varying auto exposure time, camera response function and the attenuation factors due to vignetting are also needed to be calibrated[35][36] for better direct feature tracking. For IMU-related quantities, the effects of axis misalignment, scale factor errors, and linear acceleration on the gyroscope should be considered, especially for low-cost, consumer-grade MEMS IMUs. For camera-IMU-related quantities, the temporal and spatial differences between these two sensors should be considered including camera-IMU extrinsic and their time offset. In a multi-camera multi-IMU setup, these differences in camera-camera and IMU-IMU should also be calibrated.

Relevant Time Invariant Quantities

- Camera Intrinsic
  - Principal Point $(c_x, c_y) \in \mathbb{R}^2$
  - Focal length $(f_x, f_y) \in \mathbb{R}^2$
  - Radial and Tangential Distortion Parameters $(\mathbf{k}, \mathbf{p}) \in \mathbb{R}^6 \times \mathbb{R}^2$
- IMU Axis misalignment, Scale factor, and Linear acceleration effect on gyro
  - Gyroscope and Accelerometer Axis misalignment $\mathbf{M}_g, \mathbf{M}_a \in \boldsymbol{L}^{3\times3}$
  - Gyroscope and Accelerometer Scale factor $\mathbf{S}_g, \mathbf{S}_a \in \boldsymbol{\Lambda}^{3\times3}$
  - Linear acceleration effect on gyro $\mathbf{B}_g \in \mathbb{R}^{3\times3}$
- Camera-IMU Extrinsic and Time Offset
  - Camera-IMU Displacement ${}_C\mathbf{t}_{CI} \in \mathbb{R}^3$
  - Camera-IMU Orientation $\mathbf{q}_{CI} \in \mathbb{S}^3$
  - Camera-IMU Time Offset $t_d \in \mathbb{R}$

The active states in VIN including poses-related variables and landmarks are relative quantities expressed in a relative reference frame. Based on the reference frame of the active state, VIN can be divided into world-centric and robot-centric frameworks. Landmarks are measured locally, presenting them on a local frame limits the level of uncertainty and thus reduces the linearization errors and increases consistency in EKF framework[16][17][37].

Reference Frame of Active States

- World-Centirc ${}_W\{\cdot\}$
- Robot-Centirc ${}_I\{\cdot\}$

2.3 IMU and its Kinematic Model

IMU, as a proprioceptive sensor, measures angular velocity ${}_I\tilde{\boldsymbol{\omega}}_{WI}$ and linear acceleration ${}_I\tilde{\boldsymbol{a}}_{WI}$ by gyroscope and accelerometer respectively. Generally, IMUs are often grouped into five categories in terms of performance: consumer-grade, tactical-grade, intermediate-grade, aviation-grade, and marine-grade with decreasing bias and random walk but dramatically increasing price[2]. There are also four major types of IMUs based on their mechanism: mechanical gyroscopes, RLG (ring laser gyroscopes), FOG (fiber optic gyroscopes), and MEMS (micro-electro-mechanical systems)[38][39]. Although MEMS IMUs normally perform in consumer-grade and tactical-grade, they are extremely low-cost, lightweight, and have low power consumption.

A general IMU measurement model is shown in Equation 2a and 2b. However, consumer-grade IMUs often exist axis misalignment, scale factors, and linear acceleration effects on gyroscope. They can be estimated by IMU calibration[40] and be used in a extended IMU measurement model[41].

$$\textbf{Gyroscope Measurement:} \qquad {}_I\tilde{\boldsymbol{\omega}}_{WI} = {}_I\boldsymbol{\omega}_{WI} + \mathbf{b}_g + \mathbf{n}_g \tag{2a}$$

$$\textbf{Accelerometer Measurement:} \qquad {}_I\tilde{\boldsymbol{a}}_{WI} = \mathbf{R}_{IW}\ {}_W\boldsymbol{a}_{WI} + {}_I\mathbf{g} + \mathbf{b}_a + \mathbf{n}_a \tag{2b}$$

where

${}_I\tilde{\boldsymbol{\omega}}_{WI} \in \mathbb{R}^3$: Measured angular velocity of IMU frame wrt. world frame expressed in IMU frame
${}_I\tilde{\boldsymbol{a}}_{WI} \in \mathbb{R}^3$: Measured acceleration of IMU frame wrt. world frame expressed in IMU frame
${}_I\boldsymbol{\omega}_{WI} \in \mathbb{R}^3$: True angular velocity of IMU frame wrt. world frame expressed in IMU frame
${}_W\boldsymbol{a}_{WI} \in \mathbb{R}^3$: True acceleration of IMU frame wrt. world frame expressed in IMU frame
$\mathbf{n}_g \in \mathbb{R}^3$: Gyroscope measurement noise (angular random walk)
$\mathbf{n}_a \in \mathbb{R}^3$: Accelerometer measurement noise (velocity random walk)
$\mathbf{R}_{IW} \in \mathrm{SO}(3)$: Rotation matrix from world frame to IMU frame
${}_I\mathbf{g} \in g \cdot \mathbb{S}^2$: Gravitational acceleration in IMU frame

Define Hamiltonian unit quaternion to rotation matrix transformation,

$$\mathbf{q} = \begin{bmatrix} q_w \\ q_x \\ q_y \\ q_z \end{bmatrix} \in \mathbb{S}^3, \quad \mathbf{R} = \begin{bmatrix} q_w^2 + q_x^2 - q_y^2 - q_z^2 & 2\left(q_x q_y - q_w q_z\right) & 2\left(q_x q_z + q_w q_y\right) \\ 2\left(q_x q_y + q_w q_z\right) & q_w^2 - q_x^2 + q_y^2 - q_z^2 & 2\left(q_y q_z - q_w q_x\right) \\ 2\left(q_x q_z - q_w q_y\right) & 2\left(q_y q_z + q_w q_x\right) & q_w^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix} \in \mathrm{SO}(3)$$

In VIN's IMU dynamic models, measurements of angular velocity ${}_I\tilde{\boldsymbol{\omega}}_{WI}$ and acceleration ${}_I\tilde{\boldsymbol{a}}_{WI}$ are normally treated as input. The IMU biases are modeled as random walk processes, driven by the white Gaussian noise. The continuous-time IMU dynamic is expressed in the nonlinear state space equation as shown,

$$\textbf{Translation:} \qquad {}_W\dot{\mathbf{t}}_{WI} = {}_W\mathbf{v}_{WI} \tag{3a}$$

$$\textbf{Velocity:} \qquad {}_W\dot{\mathbf{v}}_{WI} = {}_W\boldsymbol{a}_{WI} \tag{3b}$$

$$\textbf{Rotation:} \qquad \dot{\mathbf{q}}_{WI} = \frac{1}{2}\mathbf{q}_{WI} \otimes \begin{bmatrix} 0 \\ {}_I\boldsymbol{\omega}_{WI} \end{bmatrix} \tag{3c}$$

$$\textbf{Gyroscope bias:} \qquad \dot{\mathbf{b}}_g = \mathbf{n}_{\mathbf{b}_g} \tag{3d}$$

$$\textbf{Accelerometer bias:} \qquad \dot{\mathbf{b}}_a = \mathbf{n}_{\mathbf{b}_a} \tag{3e}$$

where

$\mathbf{n}_{\mathbf{b}_g} \in \mathbb{R}^3$: Gyroscope random walk (rate random walk)
$\mathbf{n}_{\mathbf{b}_a} \in \mathbb{R}^3$: Accelerometer random walk (acceleration random walk)

Since the angular velocity is expressed locally in IMU frame, Equation 3c can be obtained by quaternion multiplication, where the local angular velocity is the second term of the product in pure quaternion representation. The quaternion multiplication is shown in Equation 4,

$$\dot{\mathbf{q}}_{WI} = \frac{1}{2}\mathbf{q}_{WI} \otimes \begin{bmatrix} 0 \\ {}_I\boldsymbol{\omega}_{WI} \end{bmatrix} = \frac{1}{2}\boldsymbol{\Omega}\left(\begin{bmatrix} 0 \\ {}_I\boldsymbol{\omega}_{WI} \end{bmatrix}\right)\mathbf{q}_{WI} \tag{4}$$

where,

$$\begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} 0 \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \in \mathbb{H}_p, \qquad \boldsymbol{\Omega}(\begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix}) = \begin{bmatrix} 0 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 0 & -\omega_3 & \omega_2 \\ \omega_2 & \omega_3 & 0 & -\omega_1 \\ \omega_3 & -\omega_2 & \omega_1 & 0 \end{bmatrix} \in \mathbb{R}^{4\times 4}$$

Note that gyroscope and accelerometer random walks rarely appear in the datasheet, but can be obtained by using Allan standard deviation[42][43][44]. In this work, these four noises $\mathbf{n}_g, \mathbf{n}_a, \mathbf{n}_{\mathbf{b}_g}, \mathbf{n}_{\mathbf{b}_a}$ are all assumed to be zero mean and uncorrelated white Gaussian processes. These noise parameters (noise covariance matrix $\mathbf{Q}$) should be determined offline or adaptively by sensor calibration. In addition, for high-end IMUs (navigation grade or aviation grade), the earth rotation effect is measurable, and the IMU dynamics model considering this effect can be found in [12][3]. By substituting Equation 2a and 2b into Equation 3c and 3b respectively, the non-linear continuous-time state space Equation of IMU dynamic can be obtained in Equation 5.

$$\dot{\mathbf{x}}_I = \mathbf{f}(\mathbf{x}_I, \tilde{\mathbf{u}}, \mathbf{w}) \tag{5}$$

where

$\mathbf{x}_I \in \mathbb{R}^3 \times \mathbb{S}^3 \times \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3$: IMU states, $({}_W\mathbf{t}_{WI}, \mathbf{q}_{WI}, {}_W\mathbf{v}_{WI}, \mathbf{b}_g, \mathbf{b}_a)$
$\tilde{\mathbf{u}} \in \mathbb{R}^3 \times \mathbb{R}^3$: IMU measurements, $({}_I\tilde{\boldsymbol{\omega}}_{WI}, {}_I\tilde{\boldsymbol{a}}_{WI})$
$\mathbf{w} \in \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3$: IMU noises, $(\mathbf{n}_g, \mathbf{n}_a, \mathbf{n}_{\mathbf{b}_g}, \mathbf{n}_{\mathbf{b}_a})$

In state estimation, we can propagate the state estimate and its uncertainty through a dynamic model. To propagate state estimates in actual implementations, discretization of nonlinear continuous-time dynamic models is required. The discretization can be obtained by exact or numerical integration during period $[t_k, t_{k+1}]$. For simplifying the notations, we denote IMU states at time step k (time t_k) as $\mathbf{x}_{I_k} = \left[{}_W\mathbf{t}_k^\intercal, \mathbf{q}_{WI_k}^\intercal, {}_W\mathbf{v}_k^\intercal, \mathbf{b}_{g_k}^\intercal, \mathbf{b}_{a_k}^\intercal\right]^\intercal$ and IMU measurements at time step k as $\tilde{\mathbf{u}}_k = [{}_{I_k}\tilde{\boldsymbol{\omega}}^\intercal, {}_{I_k}\tilde{\boldsymbol{a}}^\intercal]^\intercal$, and define the periodic interval between time steps $\Delta = t_{k+1} - t_k$.

For generality, we adopt the constant linear velocity model from [45] (see the second term in RHS of Equation 6a). In addition, we assume that the direction of the angular velocity does not change in the interval[46], as shown in Equation 6c. For the time-varying rotation axis of angular velocity, we refer readers to [46] and [47]. As such, the discretized model with exact integration can be shown in the Equations 6a-6e

Translation: $${}_W\mathbf{t}_{k+1} = {}_W\mathbf{t}_k + {}_W\mathbf{v}_k\Delta - \frac{1}{2}{}_W\mathbf{g}\Delta^2 + \mathbf{R}_{WI_k}\int_{t_k}^{t_{k+1}}\int_{t_k}^{\tau}\mathbf{R}_{I_kI_t}\left({}_{I_t}\tilde{\boldsymbol{a}} - \mathbf{b}_{a_t} - \mathbf{n}_a\right)\mathrm{d}t\mathrm{d}\tau \tag{6a}$$

Velocity: $${}_W\mathbf{v}_{k+1} = {}_W\mathbf{v}_k - {}_W\mathbf{g}\Delta + \mathbf{R}_{WI_k}\int_{t_k}^{t_{k+1}}\mathbf{R}_{I_kI_t}\left({}_{I_t}\tilde{\boldsymbol{a}} - \mathbf{b}_{a_t} - \mathbf{n}_a\right)\mathrm{d}t \tag{6b}$$

Rotation: $$\mathbf{q}_{WI_{k+1}} = \mathbf{q}_{WI_k} \otimes \exp\left(\begin{bmatrix} 0 \\ \frac{1}{2}\int_{t_k}^{t_{k+1}}\left({}_{I_t}\tilde{\boldsymbol{\omega}} - \mathbf{b}_{g_t} - \mathbf{n}_g\right)\mathrm{d}t \end{bmatrix}\right) \tag{6c}$$

Gyro. bias: $$\mathbf{b}_{g_{k+1}} = \mathbf{b}_{g_k} + \int_{t_k}^{t_{k+1}}\mathbf{n}_{\mathbf{b}_g}\mathrm{d}t \tag{6d}$$

Acc. bias: $$\mathbf{b}_{a_{k+1}} = \mathbf{b}_{a_t} + \int_{t_k}^{t_{k+1}}\mathbf{n}_{\mathbf{b}_a}\mathrm{d}t \tag{6e}$$

where $\exp(\cdot)$ is the quaternion exponential that maps the angular velocity in pure quaternion (Lie Algebra) to quaternion (Lie Group).

$$\begin{bmatrix} 0 \\ \frac{\boldsymbol{\omega}}{2} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 0 \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \in \mathbb{H}_p, \qquad \exp\left(\begin{bmatrix} 0 \\ \frac{\boldsymbol{\omega}}{2} \end{bmatrix}\right) = \begin{cases} \left[\cos\left(\frac{\|\boldsymbol{\omega}\|}{2}\right), \frac{\boldsymbol{\omega}^\intercal}{\|\boldsymbol{\omega}\|}\sin\left(\frac{\|\boldsymbol{\omega}\|}{2}\right)\right]^\intercal & \in \mathbb{S}^3, \text{otherwise} \\ \left[1, \frac{1}{2}\boldsymbol{\omega}^\intercal\right]^\intercal & \in \mathbb{H}, \text{if } \|\boldsymbol{\omega}\| \to 0 \end{cases}$$

However, the exact discretized model contains integral terms that may or may not have the corresponding closed-form solution in analytical form. Also, discrete IMU measurements are sampled at periodic time steps, so certain assumptions (piecewise constant or piecewise linear) should be made during interval integration. In this case, closed-form or numerical integration with certain assumptions is used to eliminate the integral terms. Note that in Equation 6d and 6e the integration of noises does not shift mean but increases the uncertainty of bias. Isolating the integral terms, we obtain the "preintegrated" IMU measurements as follows in Equations 7a-7c,

$$\boldsymbol{\alpha}_{I_k I_{k+1}} = \int_{t_k}^{t_{k+1}} \int_{t_k}^{\tau} \mathbf{R}_{I_k I_t} \left({}_{I_t}\tilde{\boldsymbol{a}} - \mathbf{b}_{a_t} - \mathbf{n}_a\right) \mathrm{d}t\mathrm{d}\tau \tag{7a}$$

$$\boldsymbol{\beta}_{I_k I_{k+1}} = \int_{t_k}^{t_{k+1}} \mathbf{R}_{I_k I_t} \left({}_{I_t}\tilde{\boldsymbol{a}} - \mathbf{b}_{a_t} - \mathbf{n}_a\right) \mathrm{d}t \tag{7b}$$

$$\boldsymbol{\gamma}_{I_k I_{k+1}} = \exp\left(\begin{bmatrix} 0 \\ \frac{1}{2}\int_{t_k}^{t_{k+1}} \left({}_{I_t}\tilde{\boldsymbol{\omega}} - \mathbf{b}_{g_t} - \mathbf{n}_g\right) \mathrm{d}t \end{bmatrix}\right) \tag{7c}$$

These three integral terms $(\boldsymbol{\alpha}_{I_k I_{k+1}}, \boldsymbol{\beta}_{I_k I_{k+1}}, \boldsymbol{\gamma}_{I_k I_{k+1}})$ can be preintegrated with zero initial conditions (identity for rotation) since they only depend on IMU measurements and bias in interval $[t_k, t_{k+1}]$. They can be obtained by closed-form or numerical integration with certain piece-wise assumptions. The numerical integration is normally presented by Runge-Kutta integration in different orders(1^{th} order Euler, 2^{nd} order Tustin or mid-point, or 4^{th} order RK4) with increasing precision and computational cost. The closed-form integration is derived by Eckenhoff *et al.* in [14]. In terms of the piece-wise assumptions during integration, Forster *et al.* assume piecewise constant $({}_I\boldsymbol{\omega}_{WI}, {}_W\boldsymbol{a}_{WI})$ in [48][49], where Eckenhoff *et al.*[50][14] consider two models that assumes piecewise constant $({}_I\tilde{\boldsymbol{\omega}}_{WI}, {}_I\tilde{\boldsymbol{a}}_{WI})$ and $({}_I\tilde{\boldsymbol{\omega}}_{WI}, {}_I\boldsymbol{a}_{WI})$. Since preintegration terms also depend on the IMU bias, a first-order Taylor expansion on the bias linearization point is usually used to avoid re-preintegration when the bias linearization point changes[14][48][13].

By applying the closed-form or numerical integration with proper piece-wise assumption, a noise-free discretized model without integral terms $\mathbf{f}_d(\cdot)$ can be obtained to propagate state estimate in the filter-based framework or to construct the residual terms in the optimization-based framework in Equation 8a and 8b respectively. $\ominus$ denotes a generic minus operation.

$$\textbf{State Propagation:} \qquad \mathbf{x}_{I_{k+1}} = \mathbf{f}_d\left(\mathbf{x}_{I_k}, \tilde{\mathbf{u}}_k, 0\right) \tag{8a}$$

$$\textbf{State Residuals:} \qquad \mathbf{e}_k = \mathbf{x}_{I_{k+1}} \ominus \mathbf{f}_d\left(\mathbf{x}_{I_k}, \tilde{\mathbf{u}}_k, 0\right) \tag{8b}$$

Since the IMU measurement rate is much faster than that of the camera, the state propagation will take place several times before a new camera measurement arises for state update. In this case, we drop the conventional notion of the posteriori estimate $\{\cdot\}^+$ after update and the priori estimate $\{\cdot\}^-$ through propagation.

To propagate uncertainty through dynamics, we adopt concepts from Solà's paper[51] on error state dynamics, in which case, the true state $\mathbf{x}_I^{\text{true}}$ can be decomposed into a nominal state $\mathbf{x}_I$ containing a large signal and an error state $\delta\mathbf{x}_I$ containing a small signal.

$$\mathbf{x}_I^{\text{true}} = \mathbf{x}_I \oplus \delta\mathbf{x}_I$$

where $\oplus$ denotes a generic plus operation. Hence, there exist two dynamics: nominal state and error state dynamic, where the nominal state dynamic can be used for propagating the state estimate (mean), and the error state dynamic can be used for propagating uncertainty (covariance). The nominal state is propagated by $\mathbf{f}_d(\cdot)$. In order to propagate the error state, we first define the error state in VIN as,

$$\delta\mathbf{x}_I = [{}_W\delta\mathbf{t}_{WI}^\intercal, \delta\boldsymbol{\theta}_{WI}^\intercal, {}_W\delta\mathbf{v}_{WI}^\intercal, \delta\mathbf{b}_g^\intercal, \delta\mathbf{b}_a^\intercal]^\intercal \in \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3$$

where

$\delta\boldsymbol{\theta}_{WI} \in \mathbb{R}^3$: Perturbation of rotation, which is Lie Algebra of unit quaternion in Cartesian vector space

By using quaternion exponential $\exp(\cdot)$ that maps Lie Algebra to Lie Group, the composition of nominal and error state of rotation can be obtained as follows,

$$\mathbf{q}^{\text{true}} = \mathbf{q} \otimes \exp\left(\begin{bmatrix} 0 \\ \frac{\delta\boldsymbol{\theta}}{2} \end{bmatrix}\right)$$

Error state consists of only small signals, it can be considered as a perturbation of the nominal state. Thus, the error state dynamic preserves linearity. We adopt the linearized error-state process model in [51] but drop the gravity term for generality.

$$
\begin{aligned}
&\textbf{Translation:} && {}_W\delta\dot{\mathbf{t}}_{WI} = {}_W\delta\mathbf{v}_{WI} && \text{(9a)}\\
&\textbf{Velocity:} && {}_W\delta\dot{\mathbf{v}}_{WI} = -\mathbf{R}_{WI}\lfloor {}_I\tilde{\boldsymbol{a}}_{WI} - \mathbf{b}_a \rfloor_\times \delta\boldsymbol{\theta}_{WI} - \mathbf{R}_{WI}\delta\mathbf{b}_a - \mathbf{R}_{WI}\mathbf{n}_{\mathbf{b}_a} && \text{(9b)}\\
&\textbf{Rotation:} && \delta\dot{\boldsymbol{\theta}}_{WI} = -\lfloor {}_I\tilde{\boldsymbol{\omega}}_{WI} - \mathbf{b}_g \rfloor_\times \delta\boldsymbol{\theta}_{WI} - \delta\mathbf{b}_g - \mathbf{n}_{\mathbf{b}_g} && \text{(9c)}\\
&\textbf{Gyroscope bias:} && \delta\dot{\mathbf{b}}_g = \mathbf{n}_{\mathbf{b}_g} && \text{(9d)}\\
&\textbf{Accelerometer bias:} && \delta\dot{\mathbf{b}}_a = \mathbf{n}_{\mathbf{b}_a} && \text{(9e)}
\end{aligned}
$$

where the skew operation $\lfloor\cdot\rfloor_\times$ is defined as,

$$\boldsymbol{\omega} = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \in \mathbb{R}^3, \qquad \lfloor\boldsymbol{\omega}\rfloor_\times = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \in \mathbb{R}^{3\times 3}$$

A continuous-time linearization error state dynamic model over the nominal state (linearization point) can be obtained by Equations 9a-9e in state space form as,

$$\delta\dot{\mathbf{x}}_I = \mathbf{F}(\mathbf{x}_I)\delta\mathbf{x}_I + \mathbf{G}(\mathbf{x}_I)\mathbf{w} \tag{10}$$

where

$\mathbf{F}(\mathbf{x}_I) \in \mathbb{R}^{15\times 15}$: Linearized IMU state Jacobian over nominal state $\mathbf{x}_I$
$\mathbf{G}(\mathbf{x}_I) \in \mathbb{R}^{15\times 12}$: IMU process noise Jacobian over nominal state $\mathbf{x}_I$

Note that $\mathbf{w} = \left[\mathbf{n}_g^\intercal, \mathbf{n}_a^\intercal, \mathbf{n}_{\mathbf{b}_g}^\intercal, \mathbf{n}_{\mathbf{b}_a}^\intercal\right]^\intercal$ is the process noise of the dynamic, we assume the time-invariant process noise covariance matrix $\mathbf{Q}$ is known by sensor calibration. The IMU state error covariance $\mathbf{P}_I$ can be propagated through continuous-time Riccati differential equation as follow,

$$\dot{\mathbf{P}}_I = \mathbf{F}(\mathbf{x}_I)\mathbf{P}_I + \mathbf{P}_I\mathbf{F}^\intercal(\mathbf{x}_I) + \mathbf{G}(\mathbf{x}_I)\mathbf{Q}\mathbf{G}(\mathbf{x}_I)^\intercal$$

where

$\mathbf{P}_I \in \mathbb{R}^{15\times 15}$: Continuous-time IMU state error covariance
$\mathbf{Q} \in \mathbb{R}^{12\times 12}$: Continuous-time IMU process noise covariance matrix

However, obtaining the IMU error covariance in a new time step requires solving the matrix Riccati differential equation by techniques like matrix fraction decomposition. Alternatively, for simplicity, we can discretize the linearized error state dynamics, then propagate the error covariance through the discretized linearized error state dynamics in Equation 11. By exact integration with the constant assumption of Jacobian $\mathbf{F}\left(\mathbf{x}_{I_k}\right)$ and $\mathbf{G}\left(\mathbf{x}_{I_k}\right)$ over the interval $[t_k, t_{k+1}]$, we obtain

$$\delta\mathbf{x}_{I_{k+1}} = e^{\mathbf{F}\left(\mathbf{x}_{I_k}\right)(t_{k+1}-t_k)}\delta\mathbf{x}_{I_k} + \int_{t_k}^{t_{k+1}} e^{\mathbf{F}\left(\mathbf{x}_{I_k}\right)(t_{k+1}-\tau)}\mathbf{G}\left(\mathbf{x}_{I_k}\right)\mathbf{w}\,\mathrm{d}\tau \tag{11}$$

where

$e^{\mathbf{F}\left(\mathbf{x}_{I_k}\right)(t_{k+1}-t_k)} \in \mathbb{R}^{15\times 15}$: Matrix exponential (state-transition matrix) over interval $[t_k, t_{k+1}]$. For simplicity, denote it as $\boldsymbol{\Phi}(t_{k+1}, t_k)$

Given the discretized error state dynamic, the error state covariance matrix is propagated,

$$\mathbf{P}_{I_{k+1}} = \mathbf{\Phi}(t_{k+1}, t_k)\mathbf{P}_{I_k}\mathbf{\Phi}^\intercal(t_{k+1}, t_k) + \mathbf{Q}_k \tag{12}$$

where

$\mathbf{P}_{I_k} \in \mathbb{R}^{15\times 15}$: IMU state error covariance in time step k
$\mathbf{Q}_k \in \mathbb{R}^{15\times 15}$: Discrete-time process noise covariance matrix, derived by Equation 13

$$\mathbf{Q}_k = \int_{t_k}^{t_{k+1}} \mathbf{\Phi}\left(t_{k+1}, \tau\right) \mathbf{G}\left(\mathbf{x}_{I_k}\right) \mathbf{Q}\mathbf{G}^\intercal\left(\mathbf{x}_{I_k}\right) \mathbf{\Phi}^\intercal\left(t_{k+1}, \tau\right) \mathrm{d}\tau \tag{13}$$

In conclusion, the noise-free IMU state dynamic and its linearized error state dynamic enable the propagation of the state estimate and uncertainty. After deriving the corresponding discretization models, the IMU propagation step in the filter-based framework and the IMU cost in the optimization-based framework can be obtained as follows with the notation $\|\boldsymbol{a}\|^2_{\mathbf{M}} = \boldsymbol{a}^\intercal\mathbf{M}^{-1}\boldsymbol{a}$,

Filter-Based Propagation

$$\begin{aligned}\mathbf{x}_{I_{k+1}} &= \mathbf{f}_d\left(\mathbf{x}_{I_k}, \tilde{\mathbf{u}}_k, 0\right)\\ \mathbf{P}_{I_{k+1}} &= \mathbf{\Phi}(t_{k+1}, t_k)\mathbf{P}_{I_k}\mathbf{\Phi}^\intercal(t_{k+1}, t_k) + \mathbf{Q}_k\end{aligned}$$

Optimization-Based IMU Cost

$$\mathbf{C}_{I_k} = \sum_{I_k \in \mathcal{I}_k} \left\|\mathbf{x}_{I_{k+1}} \ominus \mathbf{f}_d\left(\mathbf{x}_{I_k}, \tilde{\mathbf{u}}_k, 0\right)\right\|^2_{\mathbf{P}_{I_{k+1}}}$$

The optimization cost is in typical nonlinear least square form, which requires iterative optimization based on linearization with reasonable initialization. We will discuss this later in Section 3.2 and 3.4.

2.4 Camera and its Measurement Model

Camera is the classical exteroceptive sensor. It captures the visual information of the visible local regions from scene radiance to pixel brightness. There are various cameras for vision-based motion estimation including monocular, stereo, RGB-D, fisheye, catadioptric, and event cameras. Monocular cameras have the simplest setup with only one camera, but it needs to take into account scale ambiguity. This scale vague can be resolved by stereo cameras using two-view geometry. RGB-D cameras have better depth estimation under textureless conditions with the help of an active infrared emitter added to stereo cameras. In terms of large field of view (FoV) cameras (fisheye and catadioptric cameras), Zhang *et al.* in [52] found that they are more suitable in narrow and small environments, while smaller FoV cameras perform better in larger scale scenarios. In recent years, event cameras have gained much attention for their high frame rates and wide dynamic range by only capturing the brightness changes at pixel level[53]. Regarding image capture modes of the camera, there are two distinct modes: Rolling Shutter and Global Shutter. Rolling shutter cameras read off image row by row, while global shutter cameras capture the whole picture simultaneously. In most VI-SLAM-related datasets, a global shutter camera is used by default, although it is more expensive than a rolling shutter camera. There are two major types of projection models for describing camera measurements: perspective and catadioptric projection[54]. In this work, we focus on the classical pinhole perspective projection model. We assume data association (feature extraction and matching or tracking) is done in this article. First, transform the 3D landmark from the world frame to the camera frame. Note that we use the Euclidean parametrization of 3D landmarks for generality.

Transformation:

$${}_C\mathbf{L} = \mathbf{h}_\mathbf{T}\big(\mathbf{x}_I, {}_W\mathbf{L}, (\mathbf{R}_{CI}, {}_C\mathbf{t}_{CI})\big) \triangleq \mathbf{R}_{CI}\mathbf{R}_{IW}\left({}_W\mathbf{L} - {}_W\mathbf{t}_{WI}\right) + {}_C\mathbf{t}_{CI} \tag{14}$$

where

${}_W\mathbf{L} = [x_w, y_w, z_w]^\intercal \in \mathbb{R}^3$: Landmark in world frame, ${}_W\mathbf{t}_{WL}$
${}_C\mathbf{L} = [x_c, y_c, z_c]^\intercal \in \mathbb{R}^3$: Landmark in camera frame, ${}_C\mathbf{t}_{CL}$
$\mathbf{R}_{IW} \in \mathrm{SO}(3)$: Rotation from world frame to IMU frame obtained from $\mathbf{x}_I$
${}_I\mathbf{t}_{IW} \in \mathbb{R}^3$: Position of world frame wrt. IMU frame expressed in IMU frame obtained from $\mathbf{x}_I$
$(\mathbf{R}_{CI}, {}_C\mathbf{t}_{CI}) \in \mathrm{SO}(3) \times \mathbb{R}^3$: Camera-IMU extrinsic

Then the 3D point in the camera frame is then projected into the 2D image frame.

$$\textbf{Projection:} \qquad \mathbf{z}_p = \mathbf{h}_\mathbf{P}\left({}_C\mathbf{L}\right) \triangleq \begin{bmatrix} x_c/z_c \\ y_c/z_c \end{bmatrix} \tag{15}$$

where

$\mathbf{z}_p = [x_p, y_p]^\intercal \in \mathbb{R}^2$: Projected 2D points in image frame

The projected 2D point is then distorted by the distortion function,

$$\textbf{Distortion:} \quad \mathbf{z}_d = \mathbf{h}_\mathbf{D}\big(\mathbf{z}_p, (\mathbf{k}, \mathbf{p})\big) \triangleq \underbrace{\frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6}}_{\text{Radial Distortion}} \begin{bmatrix} x_p \\ y_p \end{bmatrix} + \underbrace{\begin{bmatrix} 2p_1 x_p y_p + p_2\left(r^2 + 2{x_p}^2\right) \\ 2p_2 x_p y_p + p_1\left(r^2 + 2{y_p}^2\right) \end{bmatrix}}_{\text{Tangential Distortion}} \tag{16}$$

where $r^2 = {x_p}^2 + {y_p}^2$,

$\mathbf{z}_d = [x_d, y_d]^\intercal \in \mathbb{R}^2$	: Distorted 2D points in image frame
$r \in \mathbb{R}$	: Radius from the origin of the image to projected point in meter
$\mathbf{k} = [\mathrm{k}_1, \mathrm{k}_2, \mathrm{k}_3, \mathrm{k}_4, \mathrm{k}_5, \mathrm{k}_6]^\intercal \in \mathbb{R}^6$	: Radial distortion coefficients
$\mathbf{p} = [\mathrm{p}_1, \mathrm{p}_2]^\intercal \in \mathbb{R}^2$	: Tangential distortion coefficients

Finally, the distorted 2D points in the image frame will move the origin from the center to the top right corner and transfer the units from meters to pixels, as shown below.

$$\textbf{Origin shift and Unit change:} \qquad \tilde{\mathbf{z}} = \mathbf{h}_\mathbf{K}\left(\mathbf{z}_d, \mathbf{K}\right) + \mathbf{v} \triangleq \begin{bmatrix} f_x & 0 \\ 0 & f_y \end{bmatrix} \begin{bmatrix} x_d \\ y_d \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix} + \begin{bmatrix} n_u \\ n_v \end{bmatrix} \tag{17}$$

where

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

$\tilde{\mathbf{z}} = [u, v]^\intercal \in \mathbb{R}^2$	: Distorted camera measurement in image frame in pixel unit
$c_x, c_y \in \mathbb{R}$	: Principal point in origin shifted image frame in pixel unit
$f_x, f_y \in \mathbb{R}$	: Focal lengths in pixel
$\mathbf{K} \in \mathbb{R}^{3\times 3}$	: Camera intrinsics matrix
$\mathbf{v} = [n_u, n_v]^\intercal \in \mathbb{R}^2$	: Image noise vector

The camera-IMU extrinsic, camera intrinsic, and distortion coefficients are assumed to be given through camera calibration. In summary, a single 3D landmark can be observed by the camera frame in the camera measurement model in Equation 18,

$$\begin{aligned} \tilde{\mathbf{z}} &= \mathbf{h}\big(\mathbf{x}_I, {}_W\mathbf{L}, (\mathbf{R}_{CI}, {}_C\mathbf{t}_{CI}), (\mathbf{k}, \mathbf{p}), \mathbf{K}\big) + \mathbf{v} \\ &\triangleq \mathbf{h}_\mathbf{K}\left(\mathbf{h}_\mathbf{D}\left(\mathbf{h}_\mathbf{P}\left(\mathbf{h}_\mathbf{T}\big(\mathbf{x}_I, {}_W\mathbf{L}, (\mathbf{R}_{CI}, {}_C\mathbf{t}_{CI})\big)\right), (\mathbf{k}, \mathbf{p})\right), \mathbf{K}\right) + \mathbf{v} \end{aligned} \tag{18}$$

Since the image distortion can be removed in a preprocessing step, the undistorted camera measurement model can be obtained in Equation 19,

$$\begin{aligned} \tilde{\mathbf{z}} &= \mathbf{h}\big(\mathbf{x}_I, {}_W\mathbf{L}, (\mathbf{R}_{CI}, {}_C\mathbf{t}_{CI}), \mathbf{K}\big) + \mathbf{v} \\ &\triangleq \mathbf{h}_\mathbf{K}\left(\mathbf{h}_\mathbf{P}\left(\mathbf{h}_\mathbf{T}\big(\mathbf{x}_I, {}_W\mathbf{L}, (\mathbf{R}_{CI}, {}_C\mathbf{t}_{CI})\big)\right), \mathbf{K}\right) + \mathbf{v} \end{aligned} \tag{19}$$

Since $(\mathbf{R}_{CI}, {}_C\mathbf{t}_{CI}), \mathbf{K}$ are assumed known constants and camera and IMU are time synchronized, the undistorted camera measurement model can be simplified as

$$\tilde{\mathbf{z}} = \mathbf{h}\left(\mathbf{x}_I, {}_W\mathbf{L}\right) + \mathbf{v} \quad \xRightarrow{\text{indexing}} \quad \tilde{\mathbf{z}}_j^k = \mathbf{h}\left(\mathbf{x}_{I_k}, {}_W\mathbf{L}_j\right) + \mathbf{v}$$

where

$\mathbf{x}_{I_k} \in \mathbb{R}^3 \times \mathbb{S}^3 \times \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3$: IMU state at time step k, where there is a time-synchronized camera frame token in this time step
${}_W\mathbf{L}_j \in \mathbb{R}^3$: Landmark in world frame with a unique index j
$\tilde{\mathbf{z}}_j^k \in \mathbb{R}^2$: 2D feature measurement pixel location of single landmark j token at camera frame at time step k in image coordinate

Note that 2D features in camera measurement contain both geometric and photometric information. In this case, there are two types of measurement errors: **geometric** and **photometric** differences, used as innovation terms for the update in the filter-based framework or as measurement residuals in the optimization-based framework. The geometric difference is normally called "reprojection error" in computer vision since it measures the difference between the landmark measurement and its prediction obtained by reprojecting the landmark prediction onto the predicted camera frame. Define the reprojection residual (measurement residual or innovation term) of one 2D feature as Equation 20,

$$\textbf{Geometric Reprojection Residual:} \qquad \mathbf{r}_j^k = \mathbf{r_g}\left(\tilde{\mathbf{z}}_j^k, \mathbf{h}\left(\mathbf{x}_{I_k}, {}_W\mathbf{L}_j\right)\right) \tag{20}$$

where

$\mathbf{r}_j^k \in \mathbb{R}$ or $\mathbb{R}^2$ or $\mathbb{R}^3$: Reprojection residual of single landmark j prediction and its measurement in camera frame at time step k based on the geometric error metrics $\mathbf{r_g}(\cdot)$. For example, it contains 2 dimensions in pixel location difference and 3 dimensions in unit bearing vector error.

Based on the parametrization of 2D feature measurement, different error metrics can be used for parametrizing the reprojection error. One standard reprojection error is the pixel location difference (image plane error) on the image plane, Zhang *et al.* [52] introduce a similar faster error metric on the unit plane. In order to present some typical error metrics, we first define back-projection $\boldsymbol{\pi}^{-1}(\cdot) : \mathbb{R}^2 \mapsto \mathbb{P}^2$ that recovers the bearing vector(up to scale) from the undistorted camera measurement as Equation 21,

$$\textbf{Back-projection(up to scale):} \qquad \mathbf{m} = \boldsymbol{\pi}^{-1}(\tilde{\mathbf{z}}) \triangleq \mathbf{K}^{-1}\underline{\tilde{\mathbf{z}}} \tag{21}$$

where

$\mathbf{m} = [m_x, m_y, m_z]^\intercal \in \mathbb{P}^2$: Bearing vector(up to scale) from measurement's camera frame
$\underline{\tilde{\mathbf{z}}} = [\tilde{\mathbf{z}}, 1]^\intercal \in \mathbb{P}^2$: Camera measurement in homogeneous form with symbol $\underline{\{\cdot\}}$

Here are some of the commonly used error metrics for the geometric difference in Equation 22a-22d.

$$\textbf{Image Plane Error:} \qquad \mathbf{r}_{\tilde{\mathbf{z}}} = \tilde{\mathbf{z}} - \mathbf{h}\left(\mathbf{x}_I, {}_W\mathbf{L}\right) \tag{22a}$$

$$\textbf{Unit Plane Error:} \qquad \mathbf{r}_{\bar{\mathbf{m}}} = \begin{bmatrix} m_x/m_z \\ m_y/m_z \end{bmatrix} - \begin{bmatrix} x_c/z_c \\ y_c/z_c \end{bmatrix} \tag{22b}$$

$$\textbf{Unit Bearing Vector Error:} \qquad \mathbf{r}_{\hat{\mathbf{m}}} = \frac{\mathbf{m}}{\|\mathbf{m}\|} - \frac{{}_C\mathbf{L}}{\|{}_C\mathbf{L}\|} \tag{22c}$$

$$\textbf{Bearing Angle Error:} \qquad \mathrm{r}_\theta = \arccos\left(\frac{\mathbf{m}^\intercal {}_C\mathbf{L}}{\|\mathbf{m}\|\|{}_C\mathbf{L}\|}\right) \tag{22d}$$

where

$\mathbf{r}_{\tilde{\mathbf{z}}} \in \mathbb{R}^2$: Error on image plane
$\mathbf{r}_{\bar{\mathbf{m}}} \in \mathbb{R}^2$: Error on unit plane, where $\bar{\mathbf{m}} = \mathbf{m}/m_z$ is the bearing vector on unit plane
$\mathbf{r}_{\hat{\mathbf{m}}} \in \mathbb{R}^3$: Error between two unit bearing vectors, where $\hat{\mathbf{m}} = \mathbf{m}/\|\mathbf{m}\|$ is the unit bearing vector
$\mathrm{r}_\theta \in \mathbb{R}$: Angular error between two bearing vectors

Essentially, the geometric information captured by the camera is the bearing angle. For cameras with a large FoV, in-plane reprojection error becomes less sensitive at the edge of the plane where a large difference on the plane may correspond to a small bearing angle difference, as shown in Figure 5. In this case, Zhang *et al.* [52] suggest unit plane error metric for small FoVs cameras and unit

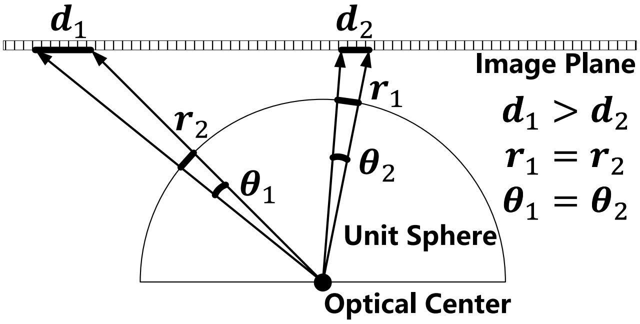


Figure 5: Reprojection Error on the image plane, unit sphere, and bearing angle in two different viewing angles with the same bearing angle difference, where d_1 and d_2 are the image plane error, r_1 and r_2 are unit bearing vector error, θ_1 and θ_2 are bearing angle error.

bearing vector error for large FoVs cameras based on their efficiency and performance over FoVs. In this article, for generality, we adopt the standard image plane error as the geometric residual for further discussion.

Note that in the above geometric error metrics, the prediction of the 3D landmark can be parametrized as a bearing vector in the *local* camera frame, which corresponds to the anchored point parametrization suggested by Solà *et al.* in [30]. In this case, we adopt the IDP parametrization[28][29] for 3D landmark prediction, which consists of a unit bearing vector $\hat{\mathbf{m}}$ and an inverse depth ρ in the local frame. Here, we define the corresponding back-projection $\boldsymbol{\pi}^{-1}(\cdot) : \mathbb{R}^2 \times \mathbb{R} \mapsto \mathbb{R}^3$ that recovers the 3D landmark from IDP parametrization in Equation 23.

$$\textbf{Back-projection:} \qquad {}_C\mathbf{L} = \boldsymbol{\pi}^{-1}(\tilde{\mathbf{z}}, \rho) \triangleq \frac{\hat{\mathbf{m}}}{\rho} \tag{23}$$

where

$\hat{\mathbf{m}} = \mathbf{m}/\|\mathbf{m}\| \in \mathbb{S}^2$: Unit bearing vector from camera frame origin to landmark
$\rho \in \mathbb{R}$: Inverse depth from camera frame origin to landmark

This back-projected landmark in the camera frame can be inversely transformed into the world frame by the inverse transformation in Equation 24,

$$\textbf{Inverse Transformation:} \qquad {}_W\mathbf{L} = \mathbf{h}_{\mathbf{T}}^{-1}(\mathbf{x}_I, {}_C\mathbf{L}) \triangleq \mathbf{R}_{IW}^{\intercal}\mathbf{R}_{CI}^{\intercal}\left({}_C\mathbf{L} - {}_C\mathbf{t}_{CI}\right) + {}_W\mathbf{t}_{WI} \tag{24}$$

Note that when evaluating the geometric difference between landmark measurement and prediction, the reprojection residual is constructed based on the 3D-to-2D correspondences. Compared to 2D-to-2D correspondences, 3D-to-2D correspondences include the estimation of 3D landmarks. According to Scaramuzza and Fraundorfer's tutorial [54], 2D-to-2D and 3D-to-2D methods are more accurate than 3D-to-3D methods. However, in the case of photometric difference, since the image intensity does not live in 3D space, there are only pair-wise 2D-to-2D correspondences (theoretically 2D-to-3D-to-2D) between two images in photometric residual. For representing photometric residual, we firstly define an image intensity function over a pixel location of camera frame at time k as $\mathbf{I}_k(\cdot) : \mathbb{R}^2 \mapsto \mathbb{R}$. Since photometric residual is based on 2D-to-2D correspondences, we define this residual for single feature j visible in an image pair at time step k_1 and k_2 as Equation 25,

$$\textbf{Photometric Residual:} \qquad \mathrm{r}_j^{k_1 \rightarrow k_2} = \mathrm{r}_{\mathbf{p}}\left(\tilde{\mathbf{z}}_j^{k_1}, \rho_j^{k_1}, \mathbf{x}_{I_{k_1}}, \mathbf{x}_{I_{k_2}}\right) \triangleq \mathbf{I}_{k_1}\left(\tilde{\mathbf{z}}_j^{k_1}\right) - \mathbf{I}_{k_2}\left(\mathbf{w}\left(\tilde{\mathbf{z}}_j^{k_1}\right)\right) \tag{25}$$

where wrap function $\mathbf{w}(\cdot) : \mathbb{R}^2 \mapsto \mathbb{R}^2$, that back-projecting a feature from k_1 frame to world frame then transforms and projects it onto k_2 frame, shown as follow,

$$\mathbf{w}\left(\tilde{\mathbf{z}}_j^{k_1}\right) = \mathbf{h}\Big(\mathbf{x}_{I_{k_2}}, \mathbf{h}_{\mathbf{T}}^{-1}\big(\mathbf{x}_{I_{k_1}}, \boldsymbol{\pi}^{-1}(\tilde{\mathbf{z}}_j^{k_1}, \rho_j^{k_1})\big)\Big)$$

$\mathrm{r}_j^{k_1 \to k_2} \in \mathbb{R}$: Photometric difference of feature j between its intensity measurement on k_1 frame and its predicted location's intensity on k_2 frame based on photometric error metrics $\mathrm{r}_{\mathbf{p}}(\cdot)$.
$\tilde{\mathbf{z}}_j^{k_1} \in \mathbb{R}^2$: 2D feature measurement pixel location of single landmark j token at camera frame at time step k_1 in image coordinate
$\rho_j^{k_1} \in \mathbb{R}$: Inverse depth of 3D landmark corresponding to feature j on k_1 frame
$\mathbf{x}_{I_{k_1}}, \mathbf{x}_{I_{k_2}}$: IMU states at time step k_1 and k_2

Note that in Equation 25, the photometric residual is computed over a pair of pixels, which is commonly used in dense (DTAM[55], DVO[56][57]) and semi-dense (LSD-SLAM[34]) feature frameworks for ease of computation. However, in sparse (VI-DSO[58][59], ROVIO[60], SVO[58][61]) feature frameworks, this photometric residual is computed in a pair of pixel patches that contain the weighted average intensity difference of neighboring pixels centered on the feature pixel. In DSO[33], Engel *et al.* evaluate the accuracy and efficiency of nine different pixel patch patterns. They also use a more precise photometric camera model that considers exposure time, camera response function, and vignetting effect, since the photometric errors are sensitive to varying brightness. For simplicity, we adopt pixel-wise instead of patch-wise photometric residual for ease of notation.

In order to update the state and error covariance in the filter-based framework or construct the visual cost (bundle adjustment (BA) in computer vision) on the optimization-based framework, we first define the measurement Jacobian matrix $\mathbf{H}_k$ over the linearization point $\mathcal{X}_k$. This measurement Jacobian matrix can be represented in rows form, where one rows is corresponding to a single feature 2D measurement. It also can be divided into columns, where left columns are derivatives over IMU states $\frac{\partial \mathbf{h}}{\partial \mathcal{I}}$ and right columns are derivatives over landmarks $\frac{\partial \mathbf{h}}{\partial \mathcal{L}}$ for all measurements.

$$\textbf{Measurement Jacobian:} \qquad \mathbf{H}_k = \begin{bmatrix} \frac{\partial \mathbf{h}_1}{\partial \mathcal{X}} \\ \vdots \\ \frac{\partial \mathbf{h}_j}{\partial \mathcal{X}} \\ \vdots \\ \frac{\partial \mathbf{h}_m}{\partial \mathcal{X}} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{h}}{\partial \mathcal{I}} & \frac{\partial \mathbf{h}}{\partial \mathcal{L}} \end{bmatrix}, \quad \frac{\partial \mathbf{h}_j}{\partial \mathcal{X}} = \begin{bmatrix} \frac{\partial \mathbf{h}_j}{\partial \mathcal{I}} & \frac{\partial \mathbf{h}_j}{\partial \mathcal{L}} \end{bmatrix} \tag{26}$$

where $\frac{\partial \mathbf{h}_j}{\partial \mathcal{X}} = \left.\frac{\partial \mathbf{h}_j}{\partial \mathcal{X}}\right|_{\mathcal{X}=\mathcal{X}_k}$ is the Jacobian corresponding to feature j 2D measurement. The Jacobian of each measurement can be divided into columns, which include derivatives over IMU states $\frac{\partial \mathbf{h}_j}{\partial \mathcal{I}}$ and derivatives over landmarks $\frac{\partial \mathbf{h}_j}{\partial \mathcal{L}}$.

The visual update step in the filter-based framework and visual cost in the optimization-based framework can be obtained as follows, where $\|\cdot\|_\gamma$ is a robust penalty function (L_1, Huber, Cauchy, etc.). $\mathbf{R}$ denotes the visual measurement noise covariance.

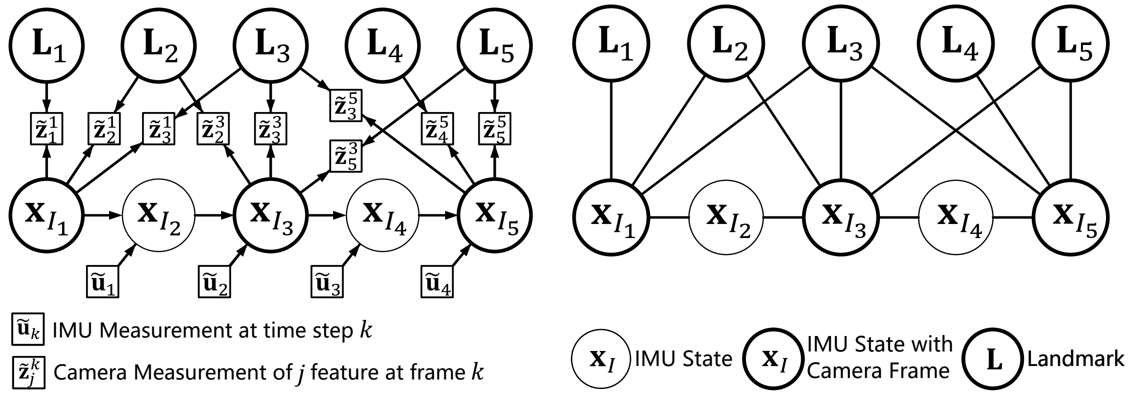
Filter-Based Upate **Optimization-Based Visual Cost**

$$\begin{aligned} \mathbf{K} &= \mathbf{P}\mathbf{H}_k^{\intercal}\left(\mathbf{H}_k\mathbf{P}\mathbf{H}_k^{\intercal} + \mathbf{R}\right)^{-1} \\ \mathbf{x}_{I_k} &= \mathbf{x}_{I_k} + \mathbf{K}\mathbf{r} \\ \mathbf{P}_{I_k} &= \left(\mathbf{I} - \mathbf{K}\mathbf{H}_k\right)\mathbf{P}_{I_k}\left(\mathbf{I} - \mathbf{K}\mathbf{H}_k\right)^{\intercal} + \mathbf{K}\mathbf{R}\mathbf{K}^{\intercal} \end{aligned} \qquad \mathbf{C}_{V_k} = \begin{cases} \sum\limits_{k \in \mathcal{I}_k} \sum\limits_{\mathbf{L} \in \mathcal{L}_k} \sum\limits_{j \in \mathrm{obs}(\mathbf{L})} \left\| \mathbf{r}_{\mathbf{g}}\left(\tilde{\mathbf{z}}_j^k, \mathbf{h}\left(\mathbf{x}_{I_k}, {}_W\mathbf{L}_j\right)\right)\right\|_{\mathbf{R}_j^k}^2 \\ \sum\limits_{k \in \mathcal{I}_k} \sum\limits_{\mathbf{L} \in \mathcal{L}_k} \sum\limits_{j \in \mathrm{obs}(\mathbf{L})} \sum\limits_{i \in \mathbf{N}(k)} \left\| \mathrm{r}_{\mathbf{p}}\left(\tilde{\mathbf{z}}_j^k, \rho_j^k, \mathbf{x}_{I_k}, \mathbf{x}_{I_i}\right)\right\|_\gamma \end{cases}$$

where in the filter-based update, $\mathbf{r}$ is the visual measurement residuals (innovation) from the geometric $\mathbf{r}_{\mathbf{g}}(\cdot)$ or photometric $\mathrm{r}_{\mathbf{p}}(\cdot)$ difference. In optimization-based visual cost, $k \in \mathcal{I}_k$ denotes the frame index k in IMU frame set I_k, $\mathbf{L} \in \mathcal{L}_k$ denotes landmark $\mathbf{L}$ in landmark set $\mathcal{L}_k$, $j \in \mathrm{obs}(\mathbf{L})$ denotes the feature index j from observable landmark $\mathbf{L}$ in IMU frame k corresponded camera frame. $i \in \mathbf{N}(k)$ denotes the neighborhood frame i around frame k.

2.5 Factor Graph representation of VIN

The evolution of visual-inertial navigation can be naturally represented in probabilistic graphical models in terms of a directed acyclic graph (Bayesian Network) or an undirected graph (Markov Random Field), as shown in Figure 6. In the Bayesian network shown in Figure 6a, nodes clearly indicate three types of quantities: to be estimated states, observed measurements, and controllable actions, while in VIN, control inputs are replaced by IMU measurements. Arrows in the Bayesian network clearly indicate causal relationships between nodes, in which the IMU state is propagated through IMU measurements, and camera measurements depends on both the state of IMU and landmarks. In Markov Random Fields shown in Figure 6b, the dependencies of nodes are clearly presented through undirected links. The sampling rate of the IMU measurements is much faster than the camera measurements, and in this case, since we assume IMU and camera measurements are time-synchronized, there will be several IMU states between two camera-synchronized IMU states (For simplicity, there is only one IMU state between the two camera-synced IMU states in Figure 6). Under the Gaussian uncertainty assumption, VIN can be represented in Gaussian Markov Random Fields[62], where the linkage and correlation between nodes can be characterized by the information matrix. The information matrix of the graphical model in VIN preserves certain sparsity. The nonzero entries of the information matrix indicate the links between nodes, and the magnitude of entries indicates the "strength" of the link(correlation between nodes). The sparsity of the information matrix directly relates to the computational efficiency of VIN[63]. Optimizing the VIN problem based on its graph structure is important for the efficiency of graph-based optimization, i.e. reasonably reducing the number of nodes (marginalization) and links (sparsification)[64][65]. For example, marginalizing the pass nodes in the filter (MSCKF[12], ROVIO[66][60], OpenVINS[67]) and fixed-lag smoother (OKVIS[68], VINS-Mono[13], VI-DSO[59], BASALT[69]), and removing the weak links for sparsification (SEIF[63][70]). Marginalization and sparsification will eventually affect the pattern of the information matrix, we refer readers to Eustice's thesis[71] for an in-depth understanding of inference in the information form.



(a) Bayesian Network

(b) Markov Random Field

Figure 6: Probabilistic Graphical Models in VIN

Factor graph is a bipartite graph originally operated by sum-product algorithm (belief propagation algorithm)[72][73]. It contains two types of nodes: variables (unknown states) and factors (known functions). Compared to Bayesian Network and Markov Random Field, factor graphs present relationships (factors) between variables. VIN's factor graph has three types of factors: an optional prior factor, IMU factors, and visual factors. The prior factor is a prior distribution over the IMU states, by including this factor, the minimization of optimization cost in VIN can be viewed as a maximum a posteriori (MAP) estimate. IMU factors are characterized by IMU dynamic model with IMU measurements and their noises and visual factors are characterized by the camera measurement model with camera measurements and their noises. Note that in this section the graph includes all states (full batch optimization) at each time step and is incremented over time. The prior distribution describes the uncertainty of the initial state, whereas, in fixed-lag smoother, it contains prior information from marginalization. The full factor graph describing VIN is shown in Figure 7.

As the relationship between the Gaussian Markov Random field and information matrix, the factor graph is naturally related to the Jacobian matrix in VIN where rows in Jacobian show factors that indicate the relation between variables. IMU factors and visual factors are binary factors, where IMU factors link two IMU states, and visual factors link one IMU state and one landmark respectively. The prior factor is a uni-nary factor for the initial state in batch optimization or n-nary factors in fixed-lag smoother depending on marginalization. A conceptual diagram representing the relationship between rows and columns in the Jacobian and Hessian (information matrix) is shown in Figure 8.

The factor graph of VIN can be formulated as an optimization cost function in terms of factor terms. The full cost of VIN in batch optimization includes three costs: Prior cost, IMU cost, and Visual cost shown in Equations 27a-27c, where the prior cost of the initial states assumes Gaussian

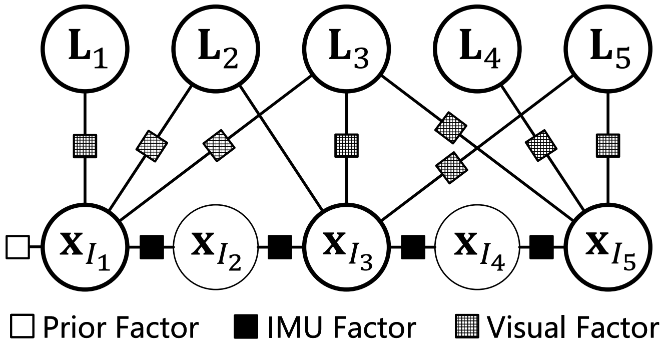


Figure 7: Factor Graph in VIN

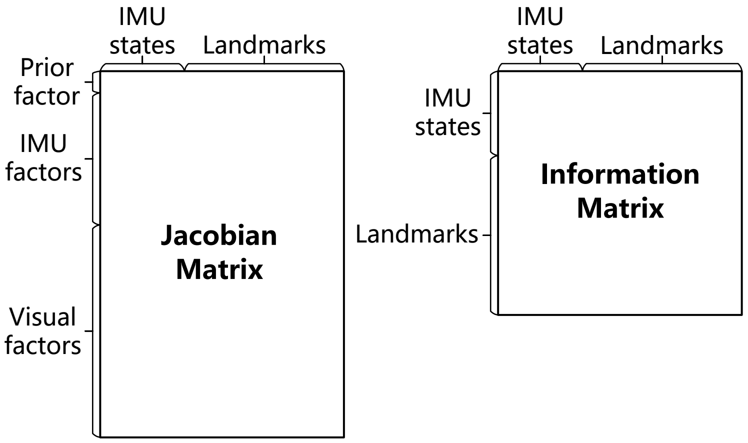


Figure 8: Jacobian and Hessian(Information Matrix) in VIN

prior distribution with mean $\mathbf{x}_p$ and covariance $\mathbf{P}_p$.

$$\textbf{Prior Cost:} \quad \mathbf{C}_p = \|\mathbf{x}_{I_0} \ominus \mathbf{x}_p\|^2_{\mathbf{P}_p} \tag{27a}$$

$$\textbf{IMU Cost:} \quad \mathbf{C}_{I_k} = \sum_{I_k \in \mathcal{I}_k} \left\|\mathbf{x}_{I_{k+1}} \ominus \mathbf{f}_d\left(\mathbf{x}_{I_k}, \tilde{\mathbf{u}}_k, 0\right)\right\|^2_{\mathbf{P}_{I_{k+1}}} \tag{27b}$$

$$\textbf{Visual Cost:} \quad \mathbf{C}_{V_k} = \begin{cases} \sum\limits_{k\in\mathcal{I}_k} \sum\limits_{\mathbf{L}\in\mathcal{L}_k} \sum\limits_{j\in\text{obs}(\mathbf{L})} \left\|\mathbf{r_g}\left(\tilde{\mathbf{z}}^k_j, \mathbf{h}\left(\mathbf{x}_{I_k}, {}_W\mathbf{L}_j\right)\right)\right\|^2_{\mathbf{R}^k_j} & \text{Geometric} \\ \sum\limits_{k\in\mathcal{I}_k} \sum\limits_{\mathbf{L}\in\mathcal{L}_k} \sum\limits_{j\in\text{obs}(\mathbf{L})} \sum\limits_{i\in\mathbf{N}(k)} \left\|\mathrm{r}_{\mathbf{p}}\left(\tilde{\mathbf{z}}^k_j, \rho^k_j, \mathbf{x}_{I_k}, \mathbf{x}_{I_i}\right)\right\|_\gamma & \text{Photometric} \end{cases} \tag{27c}$$

The full cost can be written in Equation 28. By minimizing the cost, the optimal states and their uncertainties estimation can be obtained, with Gaussian assumption, the optimal states and uncertainties are presented in means and covariances (or information matrix) respectively.

$$\mathbf{C}(\mathcal{X}_k) = \mathbf{C}_p + \mathbf{C}_{I_k} + \mathbf{C}_{V_k} \tag{28}$$

3 State Estimation Methods in VIN

In the preview section, we introduce the relevant quantities (states of interest and support quantities) and models (IMU and camera models) in VIN and their graph-based representations. In this section, we discuss the existing methodologies for VIN state estimation. In this tutorial, we mainly focus on tightly-coupled visual-inertial fusion. There are two main tightly-coupled visual-inertial state estimation schemes: filter-based and optimization-based. As such, we briefly classify these state estimation methods in terms of the width of the state horizon.

- **Filter (one-step recursive estimation):**

It normally estimates the latest state given the latest measurement. It boosts efficiency but loses accuracy due to the marginalization of all past information. The loss of accuracy is also due to the accumulation of linearization errors[49].

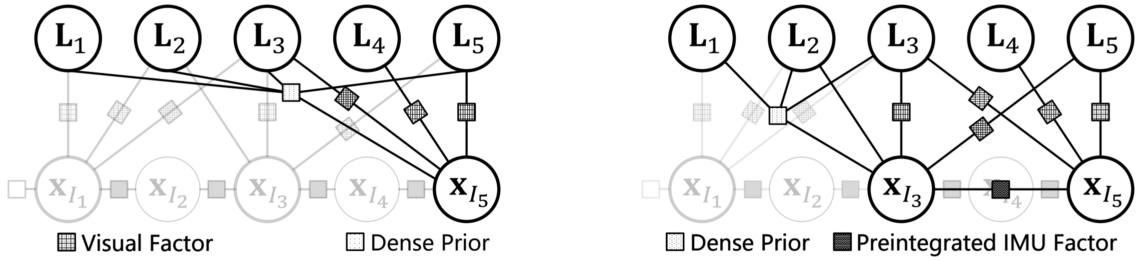
- **Fixed-lag smoother (moving horizon estimator or sliding window filter):**

  It estimates the recent states given the recent measurements. It balances the speed and precision of estimation by changing the width of the moving horizon. Note that fixed-lag smoother is normally based on optimization.

- **Batch estimator (full horizon estimation or full smoother):**

  It estimates all states given all measurements using nonlinear optimization. It gradually becomes computationally intractable with continuously increasing states.

The factor graphs of the filter and the fixed-lag smoother are shown in Figure 9a and 9b respectively. The factor graph of the batch estimator is shown in the previous section in Figure 7. Note that due to the marginalization of past states, a dense prior factor will be introduced in both filter and fixed-lag smoother. This will increase the density of the information matrix, and eventually reduce the efficiency of the estimation. In this case, to reduce the computational burden from the "fill-in" in the information matrix caused by the densely connected prior factor, certain sparsification methods are designed to balance the efficiency and accuracy (either "break" the links or "drop" the nodes).



(a) Filter

(b) Fixed-lag Smoother

Figure 9: Filter and Smoother in VIN

3.1 Filter-based Methods

All filter-based methods in VIN are built on the foundation of Bayes filters. Bayes filters provide a unified framework for probabilistic state estimation[74]. In the Bayes filter, the state information is represented by the probability distribution (probability density function), and the evolution of the state information can be characterized as two stages: Propagation and Update. In the propagation phase, the state loses information due to propagating the state through the dynamic model, while in the update phase, the state gains information by obtaining measurements from the measurement model. However, parameterizing state information as probability distributions is computationally intractable. Numerical approximations or assumptions should be made to deploy Bayesian concepts into actual implementations. The typical numerical approximation of the Bayes filter is the particle filter based on Monte Carlo simulation and importance sampling techniques. Also, under the linear Gaussian assumption, the Bayesian filter becomes a Kalman filter, which is the optimal state estimator under this assumption. However, in VIN, both the IMU dynamics and camera measurement models are nonlinear. In this case, keeping the reasonable Gaussian noises assumption, the Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF) are often used in VIN to cope with nonlinear models. In Figure 10, We briefly illustrate the relationship between these filters.

The particle filter is capable of working under the nonlinear model with non-Gaussian noise. However, since particle filters are built based on importance sampling techniques, a large number of particles (sampling points) are required to characterize the distributions of high-dimensional states (curse of dimensionality). In this case, the Rao-Blackwellisation technique is used to marginalize out some of the variables[75]. Rao-Blackwellised particle filters (RBPF) have been successfully applied in many visual-based navigation works, such as FastSLAM[76], FastSLAM2[77], and its improved versions[78][79][80] with inverse depth landmark initialization, fewer particles, and better particle weight computation respectively. Although the above-mentioned RBPFs work well in the

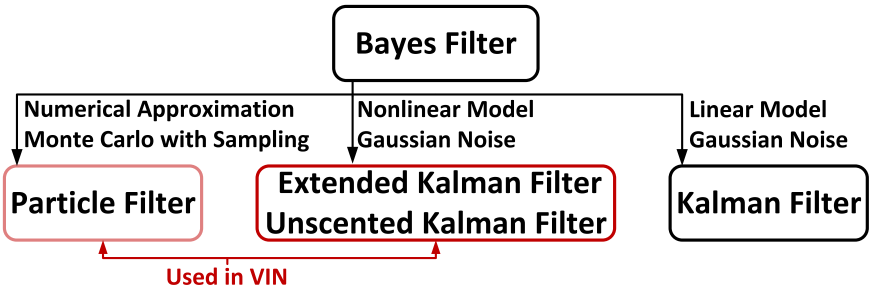


Figure 10: Filter-based Methods in VIN

2D environment, they are not computationally effective compared to UKFs and EKFs methods with high-dimensional states, such as VIN with motion in three dimensions, more dynamic states (velocity), and sensor biases[81]. Both UKFs and EKFs are designed to overcome nonlinearities in the models, with Gaussian noise assumptions, and they have similar theoretical complexities. EKFs require Jacobian over linearization points, while UKFs need to calculate the sigma points in each iteration. Compared to the analytical Jacobian matrix, the sigma points require repeated calculations and are more computationally intensive, but UKFs and EKFs have similar performance in SLAM[82]. In this case, filter-based methods in VIN mostly concentrate on the EKF and its dual-form extended information filter (EIF) because of their efficiency. EKFs are initially implemented in visual-only SLAM, including MonoSLAM[83][83] that consider a constant velocity motion model for active feature searching, and robocentric frameworks[17][16] that change the reference frame from world to local for improving the consistency of EKFs-based SLAM. In order to reduce computational burdens while keeping the information, sparsification methods are introduced including sparse extended information filters (SEIF)[63][70] that neglecting or conservatively eliminating the distant landmarks[70] in the information form.

The observability and consistency of the proposed estimator are always of-interest in filter-based methods. In [84], an analysis of the observability of non-linear 2D world-centric SLAM is proposed, which indicates that the direct use of linear observability tools yields inconsistency. This problem is further investigated in EKF-based SLAM[85] by Huang *et al.* and first-estimates Jacobian EKF (FEJ-EKF)[86] is proposed to improve the consistency and reduce the linearization error, where Jacobians are computed using the first available estimate for each state variable. FEJ-EKF is further improved by observability-constrained EKF (OC-EKF)[87], which selects the linearization points that preserve observability while minimizing the error between the linearization point and true state. The concept of OC-EKF was then applied to 3D VIN by Hesch *et al.*'s work observability constrained VINS (OC-VINS) [88][89][90].

One of the most well-known filter-based methods in VIN is the multi-state constraint Kalman filter (MSCKF)[12] introduced by Mourikis and Roumeliotis in 2007. MSCKF is a structureless EKF-based method that marginalizes all landmarks by nullspace operation. Bloesch *et al.* propose ROVIO[66][60], an EFK and iterated extended Kalman filter(IEKF) based method that utilizes the photometric innovation instead of typical geometric innovation for state update. ROVIO includes the dynamic of a fixed number of stationary landmarks for robust estimation. While conventional EKFs operate in vector spaces, some states of VIN lie in Lie groups. In this case, Brossard *et al.* propose a UKF on Lie groups[91][92] based on invariant Kalman filter(I-KF) theory[93]. Solà presents a dedicated tutorial on error-state extended Kalman filter(ES-EKF)[51] that enables uncertainty propagation of the quaternion Lie group under error-state dynamics.

3.2 Optimization-based Methods

Compared to EKF-based methods, most optimization-based methods are also based on linearization operations but are able to iteratively re-linearize at new linearization points and repeat until the optimization converges. In [94], Bell and Cathey prove the equivalency between IEKF and Gauss-Newton optimization at filter update. This relinearization process in optimization reduces the linearization error. The optimization-based methods are also called graph-based methods[95] in SLAM. As indicated in Section 2.5, VIN has a well-forming graphical interpretation in factor graphs. The optimization problem in VIN can be formulated as a summation of cost functions over the factors as shown in Equations 27a-27c including prior, IMU, and visual cost. Over the past decade, optimization-based methods have experienced tremendous development due to the maturity of optimization tools for state estimation including Google Ceres[96], g2o[97], GTSAM[98], etc. There are many optimization-based masterpieces in VIN are introduced during this period, including OKVIS[68], SVO+GTSAM[48][49], ORB-VIO[99], IS-VIO[100], VINS-Mono[13], VI-DSO[59], Kimera[101], BASALT[69], DM-VIO[102], etc. To balance efficiency and accuracy, these fixed-lag smoother-based methods maintain a subset of the full state compared to batch estimators. OKVIS[68] introduces a recent keyframe-based graph structure as a moving horizon in the fixed-lag smoother framework. VINS-Mono[13] and SVO+GTSAM[48][49] remove IMU states between consecutive camera keyframes by using IMU preintegrated factors. ORB-VIO[99] uses the covisibility graph that includes the past camera frames sharing enough common features with the current keyframes. IS-VIO[100] introduces a specific sparsification method to sparsify dense priors while keeping information loss to a minimum. Kimera[101] supports metric-semantic mapping with dense mesh. BASALT[69] adopts the sparsification methods proposed in NFR[65] that maintain the information measured by Kullback–Leibler divergence after marginalizing nodes. The above-mentioned methods are indirect methods that minimize the geometric error in optimization, where VI-DSO[59] and DM-VIO[102] are direct methods based on photometric error. They both include the scale and gravity direction in the tight-coupled optimization while utilizing dynamic and delayed marginalization respectively.

For long-term large-scale mapping, batch estimators with loop closure are commonly used in VIN. In this scenario, the computational complexity increases unboundedly as the graph grows. Kaess *et al*. propose incremental nonlinear optimization methods: iSAM[103] and iSAM2[104], that avoiding repeated batch processing steps. However, the computational complexity depends not only on the increasing scale of the graph but also on the connection density of the graph. In this case, nodes marginalization(or removal) and edges sparsification are introduced including generic node removal (GLC)[64] and nonlinear factor recovery (NFR)[65].

3.3 On-manifold Operation

In VIN, the motion of the pose is normally represented by rigid body transformation in special Euclidean group SE(3), where rotation has various parametrizations with different properties discussed in Section 2.2. All of these parameterizations related to pose transformation are Lie groups, which is a smooth manifold. However, perturbations in Lie groups are not simple addition and subtraction operations like in vector spaces. Fortunately, all elements in the Lie group can be mapped into a corresponding tangent space (Lie algebra) in its vector form. In this way, optimization tools that typically operate on vector spaces can be used for on-manifold optimization. The bijective mapping functions between Lie group $\mathcal{M}$ and Lie algebra in vector space $\mathbb{R}^m$ are Exponential mapping function $\mathrm{Exp}(\cdot) : \mathbb{R}^m \mapsto \mathcal{M}$ and Logarithm mapping function $\mathrm{Log}(\cdot) : \mathcal{M} \mapsto \mathbb{R}^m$. In the previous section, we denote generic plus and minus operations $\oplus$ and $\ominus$, we specify this operation in vector space and Lie groups respectively as followed.

	Vector Space	**Lie group and Lie algebra**
PLus ($\oplus$)	$\mathbf{y} = \mathbf{x} + \delta\mathbf{x}$	$\mathcal{Y} = \mathcal{X}\mathrm{Exp}\left({}^{\mathcal{X}}\boldsymbol{\tau}\right)$
Minus ($\ominus$)	$\delta\mathbf{x} = \mathbf{y} - \mathbf{x}$	${}^{\mathcal{X}}\boldsymbol{\tau} = \mathrm{Log}\left(\mathcal{X}^{-1}\mathcal{Y}\right)$

where we adopt the right-hand convention

$\mathbf{x}, \mathbf{y} \in \mathbb{R}^m$	: Quantities in vector space
$\delta\mathbf{x} \in \mathbb{R}^m$	: Perturbation around $\mathbf{x}$ in vector space
$\mathcal{X}, \mathcal{Y} \in \mathcal{M}$	: Quantities in Lie group
${}^{\mathcal{X}}\boldsymbol{\tau} \in \mathbb{R}^m$	: Perturbation over $\mathcal{X}$ (right-hand convention) in Lie algebra with vector form

The back-and-forth mapping (Exponential and Logarithm) between the manifold and tangent space enables the iterative optimization algorithms on manifolds, such as gradient descent, Gauss-Newton, and Levenberg-Marquardt methods, etc. Solà *et al*. provide a detailed tutorial on Lie theory for robotics state estimation[105], Forster *et al*. also provide a comprehensive formulation of on-manifold operations for IMU preintegration[48][49]. In addition, these on-manifold operations are important for uncertainty propagation on Lie groups. By assuming Gaussian noise on the tangent space of the manifold, Barfoot and Furgale[106] give an accurate characterization of uncertainty propagation over SE(3) with left-hand convention. Mangelson *et al*.[107] extend the work of Barfoot and Furgale by considering jointly correlated poses.

3.4 Calibration and Initialization in VIN

Calibration in VIN is to estimate the model-related quantities (mostly time-invariant) discussed in Section 2.2. Since two types of sensory modalities are used in VIN, the calibration involves self-calibration and inter-calibration over visual and inertial sensors. Furthermore, these quantities can be calibrated during estimation (online calibration) or beforehand (offline calibration). Camera self-calibration mainly involves the estimation of the camera intrinsic[108][109] including the camera's principal point, focal length, and distortion parameters. For stereo camera setup, stereo rectification[110] is also needed which makes the epipolar lines horizontal with proper scale for directly capturing the image disparity. Among the direct methods relying on the photometric differences of feature pixels, a more accurate photometric camera model is needed that takes into account the time-varying auto-exposure time, camera response function, and vignetting-induced attenuation factors[35][59][66]. IMU calibration normally requires the assistance of exteroceptive sensors (magnetometers or cameras). In IMU calibration, the most concerned quantities are the gyroscope and accelerometer biases. However, these two biases are time-varying quantities as random walk processes, and in this case, both biases are included in the state vector in VIN. The noise parameters in IMU can be obtained by using Allan standard deviation[42][43][44]. For low-cost consumer-grade IMU, the effects of axis misalignment, scale factor errors, and linear acceleration on the gyroscope should be also considered[40][41]. IMU-camera calibration is to estimate the geometric and temporal difference between two types of sensors. The geometric difference is normally called IMU-camera extrinsic, which includes the displacement and orientation between two sensors. The temporal difference is the time offset between camera and IMU measurements caused by their different latency and sampling rate. IMU-camera calibration can be solved offline by kalibr[111][40]. The IMU-camera extrinsic is calibrated online in many VIN works including OKVIS[68], ROVIO[66][60], VINS-Mono[13], and OpenVINS[67] in filter-based or optimization-based frameworks. The time offset between IMU and camera measurements can also be estimated online in [19][20] or avoided by hardware synchronization[22].

The initialization is critical for the nonlinear least square optimization in fixed-lag smoother and batch estimator in VIN. Proper initialization can prevent the optimization from converging to false local minima due to nonlinearity and non-convexity. Although, Olson *et al.*[112] have shown that stochastic gradient descent (SGD) can be used for poor initial estimates. In optimization-based visual-only and visual-inertial navigation, initialization consists of the majority of the workload since optimization toolboxes(Ceres, g2o, GTSAM, etc) are used for final code implementation. The initialization from the visual-only navigation relies on the classical multiple-view geometry in computer vision. In quantities required to be initialized in tight-coupled optimization-based VIN contains active states (the pose of camera frames corresponding IMU frames, velocity, gyroscope and accelerometer biases, and 3D landmarks position), scale, and gravity direction. For the monocular camera, the relative 2-view camera pose can be estimated by the commonly used five-point algorithm[113] by Nister or eight-point algorithm[114] by Longuet-Higgins, or by decomposing a homography matrix if viewing planar scenes[115]. After obtaining the relative pose (rotation and up-to-scale translation), all features observed in these two frames can be triangulated to estimate their 3D positions up to scale. Then the absolute pose of the camera frame that observes these 3D landmarks knowing their 3D positions can be estimated by the perspective-n-point (PnP) methods like P3P[116][117] and EPnP[118], etc. Kneip and Furgale provide a unified software library OpenGV[119] for relative and absolute camera pose estimation. Dong-Si and Mourikis[120][121] present closed-form solutions for VIN while considering the number of features and frames with certain trajectories needed for possible solutions. Martinelli[122] derives intuitive closed-form solutions that investigate in detail the number of distinct solutions with different numbers of features and frames with different motions under the biased and unbiased case. In ORB-VIO[99], Mur-Artal and Tardos propose a novel IMU initialization method that first estimates gyroscope bias by consecutive keyframes visual odometry then approximates the scale and gravity direction given the preintegration over at least four keyframes, after this accelerometer bias is estimated and scale and gravity direction are refined in a similar manner, eventually velocity can be estimated given two consecutive position estimates, gravity vector estimate, and the preintegrated factors between them. In VINS-Mono[13], Qin *et al.* presents an efficient loosely coupled initialization procedure that ignores accelerometer bias, which is difficult to observe because most of the magnitude of the acceleration is due to gravity. Campos *et al.*[123] introduces an outstanding inertial-only optimization for visual-inertial initialization that jointly optimizes all the IMU-related variables and the scale factor in one step using MAP estimation.

4 Performance Evaluation and Improvement in VIN

The performance of VIN can be evaluated in terms of accuracy, efficiency, and robustness. Accuracy and efficiency can be assessed quantitatively, while robustness assessment is more qualitative based on illumination changes, motion blur, and low-texture scenes. Accuracy metrics in VIN include absolute trajectory error (ATE) and relative pose error (RPE) that evaluate the geometric error over the whole trajectory or over segmented sub-trajectory respectively[124][125][126]. When considering the uncertainty of the estimate, the accuracy can be evaluated in terms of consistency by normalized estimation error squared (NEES), which can be tested only in simulations[127]. Efficiency metrics in VIN are normally evaluated in terms of computing resource usage and average processing time instead of floating point operations per second (FLOPs) for precise computational complexity measurement which is hard to compute depending on the convergence of iterative optimization and randomness induced by RANSAC. Delmerico and Scaramuzza present a detailed comparison[128] of monocular VIO in terms of accuracy, latency, and computational constraints over popular VIO pipelines. We list some of the public-available datasets for VIN performance evaluation in Table 2.

Dataset	Environment	Platform	Groudtruth	Year	Ref.
NTU VIRAL	Outdoor	UAV	Total Station	2022	[129]
Newer College	Outdoor	Handheld	LiDAR Scan Prior Map	2021	[130]
MADMAX	Outdoor	Rover	RTK-GNSS	2021	[131]
Hilti	Indoor Outdoor	Handheld	Motion Capture System Total Station	2021	[132]
4Seasons	Outdoor	Car	RTK-GNSS	2020	[133]
UZH-FPV Drone	Indoor Outdoor	UAV	Total Station	2019	[134]
KAIST Urban	Outdoor	Car	RTK-GNSS SLAM	2019	[135]
TUM VI	Indoor Outdoor	Handheld	Motion Capture System	2018	[136]
Canoe VI	Outdoor River	USV	GPS/INS	2018	[137]
Oxford Car	Outdoor	Car	GPS/INS	2017	[138]
NCLT	Outdoor	Rover	RTK-GNSS SLAM	2016	[139]
EuRoC	Indoor	UAV	Motion Capture System Total Station	2016	[140]
KITTI	Outdoor	Car	RTK-GNSS/INS	2012	[124]

Table 2: VIN Datasets

4.1 Accuracy, Efficiency, and Robustness Improvements

In this section, we summarize efforts to improve the accuracy, efficiency, and robustness of existing filter-based and optimization-based frameworks. Since EKF-based methods are more efficient than UKF-based methods while having similar accuracy performance[82], we focus on analyzing EKF-based methods. EKF-based and optimization-based methods are normally based on linearization and Gaussian assumptions. In EKF-based frameworks, linearization error is the main cause of inaccuracy. In this case, inverse depth landmark parametrization is presented[31][28][29], which reduces the linearization error of the measurement model in low-parallax scenes. Robot-centric map joining[16][17] also improves the linearization of the model by the robot-centered representation that bounds the uncertainty. This linearization error can also be reduced by carefully selecting linearization points[86][88] or iterative update linearization points in IEKF[60]. However, these filter-based methods continuously marginalize past states, which leads to a densely connected prior factor with landmarks as shown in Figure 9a. Strasdat *et al.*[141][142] indicate that the computational cost of filter-based methods scales poorly with the number of landmarks. They conclude that the number of landmarks involved is the key factor for increasing the accuracy of visual SLAM, while in VIN, Bloesch *et al.*[60] point out that the quality of tracked landmarks is more important than the quantity because the IMU provides a good motion prior. Optimization-based methods utilize the iterative optimization methods that re-linearize the model in a new linearization point in each iteration, this mechanism naturally reduces the linearization error. These smoothing methods also maintain the past information that will be marginalized in the filter-based methods. For long-term navigation, loop closure by pose graph optimization and covisibility graph[143][99] increase the accuracy while maintaining the efficiency.

The efficiency of VIN can be improved by reducing nodes and factors (edges in MRF) or incremental solutions. Nodes can be marginalized or removed, and resulting dense connections caused by fill-in in marginalization can be re-assigned by different topologies[65] or simply breaking the weak link[63][70]. In VIN, the past visual nodes (camera frame with corresponding IMU frame) are marginalized in filter and fixed-lag smoother, which leads to a dense prior factor. In filter-based works, sparsification that drops the weak link is used[63][70], whereas in fixed-lag smoother, specific sparsification(IS-VIO[100]) that drops connections between landmarks and velocity and biases nodes or general sparsifications that enable different Markov blanket approximation including tree, subgraph and cliquey subgraph topologies while keeping the information loss to a minimum(NRF[65]) are used for increasing the efficiency. In keyframes-based works including OKVIS[68], VINS-Mono[13], and SVO+GTSAM[48][49], the visual factors are simply dropped in non-keyframe visual nodes, and the IMU nodes are preintegrated as a preintegration factor between consecutive keyframes. In some extreme cases, all landmarks are marginalized out using nullspace operation[12] or using smart factors[23][48][49]. Compared to feature-based indirect methods, direct methods are faster because they skip the data association stage avoiding the expensive feature extraction, matching, and outlier removal processes[58][59].

In real-world scenes, visual conditions involving dynamic objects, motion blur, low texture, and illumination changes often occur. In VIN, the combination of visual and inertial sensing modalities naturally robustifies the estimation since visual information avoids the drift caused by biased dead reckoning in inertial integration and IMU provides motion hints for visual sensing. The robustness of VIN can be further improved by alleviating the effect caused by misleading visual information, including outlier rejection and inlier selection or robust M-estimators. The outlier rejection is commonly achieved by RANSAC[144] or Mahalanobis distance test over the landmark innovation and its covariance prediction[12][66][60]. By involving the motion model as constraints, fewer points are needed in RANSAC motion estimation procedures, from normally five[113] or eight points[114] for six-degree motion, to two points in 2D relative motion[145], to one point in 2D motion considering Ackermann steering model[146][147][148]. Additionally, the motion prior provided by IMU can be used to robustly track the static features in a highly dynamic environment[149] with aggressive camera motion[150]. In terms of inlier selection, Zhao and Vela present an active feature selection and matching algorithm[151][152] that reduces computational cost while maintaining the accuracy and robustness of pose tracking. The robust M-estimators are commonly used in VIN for reducing the effects caused by unrejected outliers. Mactavish and Barfoot provide a comparison[153] of different robust cost functions in visual navigation. Yang *et al.*[154] introduce a robust penalty function with a control parameter for graduated non-convexity, they claim that the proposed approach can be a valid replacement for RANSAC with better performance in many spatial perception problems. In specific VIN works, ROVIO[66][60] robustifies its filtering estimation by considering the dynamics of the static landmarks. VINS-Mono[13] presents a robust initialization procedure to deal with the gyro bias. Failure detection and recovery procedures are also important for the overall system robustness, as presented in typical direct methods (DSO[33], etc) and indirect methods (ORB-SLAM[155], VINS-Mono, etc). DSO detects the failure motion by examining the current motion's RMSE and attempts to recover by trying up to 27 different small rotations in different directions. ORB-SLAM detects the failure motion by examining the solutions provided by decomposing a homography or fundamental matrix and repeats the initialization process for recovery if not a clear winner solution is provided. VINS-Mono detects the failure by examining the number of features, continuity of position or rotation in consecutive frames, and change in gyro bias estimate. It repeats the initialization like ORB-SLAM for failure recovery.

5 Learning Era in VIN

Both the filter-based and optimization-based approaches discussed above are typical model-based methods built on models of IMU dynamics and camera measurements. Due to the vast development of deep learning, data-driven methods gradually challenge(or replace) the classical model-based methods in many areas, especially in the domain of natural language processing and computer vision. The successful data-driven applications related to visual and inertial navigation involve single image depth estimation (SIDE), deep visual odometry, deep inertial odometry, novel view synthesis (NVS), semantic SLAM, feature detection and matching, etc. Single image depth estimation[156][157][158] provided a learning-based solution for feature depth estimation which can be used for monocular visual odometry and dense mapping. For deep visual odometry, Yang *et al.*[159] introduce D3VO that predicts monocular depth, photometric uncertainty, and relative camera pose in CNN-based network architectures. Koestler *et al.* present TANDEM[160], a real-time monocular tracking and dense mapping framework, which combines DSO[33] and dense depth maps rendered from the global truncated signed distance function (TSDF) model to achieve visual odometry. Ummenhofer *et al.* introduce a supervised CNNs-based framework DeMoN[161] for depth and pose estimation of two frames, where the network is based on encoder-decoder pairs with an iterative loop structure. For deep inertial odometry, Chen *et al.* propose IONet[162], which is the first end-to-end learning framework that takes raw IMU measurements and outputs 2D inertial odometry trajectories. Yan *et al.* provide RoNIN[163] dataset and propose three deep network architectures based on ResNet, LSTM, and TCN for data-driven inertial navigation. Brossard *et al.* propose the AI-IMU[164], which adopts the filter-based framework for IMU state estimation and uses a deep neural network to dynamically adapt the noise parameters. Liu *et al.* introduce TLIO[165] which incorporates learning-based displacement distribution estimation into EKF-based inertial odometry. Buchanan *et al.* introduce a learning-based IMU bias predictions[166] using two commonly used sequential networks: LSTMs and Transformers. In the case of novel view synthesis, Mildenhall *et al.* introduce neural radiance field (NeRF) [167], a state-of-the-art view synthesis method using a fully connected deep network and principles from classical ray tracing.

Learning-based methods can also exact the semantic information in visual measurements. Landmarks in the environment not only contain geometric information but also semantic category information, jointly utilizing the semantic and geometric can achieve more robust and informative landmarks distinguishment. Xiao *et al.* proposed Dynamic-SLAM[168], which added semantic segmentation to distinguish static and dynamic objects under the ORB-SLAM framework. Doherty *et al.* [169] propose a robust semantic SLAM with probabilistic data association. Rosinol *et al.* provide an open-source C++ library Kimera[101] that enables real-time VIN with 3D mesh reconstruction and semantic labeling.

In terms of feature extraction and matching, DeTone *et al.* propose a CNN-based feature detection and description algorithm SuperPoint[170]. Sarlin *et al.* introduce SuperGlue[171], a learning-based feature matching based on graph neural networks. The data-driven approach achieves excellent performance across many components of the vision and initial navigation pipelines. However, there is still room for the complete end-to-end learning-based VIN. We refer the reader to paper [172] for considering the limitations and potentials of the learning-based methods in robotics perception. Chen *et al.* provide a comprehensive survey[173] for data-driven visual and inertial localization and mapping.

6 Conclusions

Visual inertial navigation fuses the information provided by the camera and IMU sensor to obtain navigation-related geometric information. The complementary property and low-cost lightweight characteristic of these two sensory modalities make them popular in many navigation applications. VIN is a typical state estimation problem, in this article, we clearly define the relevant quantities in VIN and their parametrization and symbolization. Furthermore, the IMU dynamic and camera measurement models are also presented while considering IMU dynamic propagation with preintegration and its linearized discretized error state dynamic, and camera model with geometric and photometric residuals. In model-based methods including filter and optimization, VIN can be straightforwardly visualized from a factor graph perspective. The performance of VIN is continuously improved using graph-based optimization methods. The data-driven methods have revolutionized many aspects of the visual part of VIN, these learning-based methods provide alternatives with respect to the classical model-based methods including the end-to-end learning methods and hybrid methods that combine data-driven and model-based methods. Overall, This article hopes to provide a comprehensive overview of the VIN in terms of its relevant quantity presentation, model formulation, and possible methodologies with certain improvements.

References

[1] Peter Corke, Jorge Lobo, and Jorge Dias. An introduction to inertial and visual sensing. *The International Journal of Robotics Research*, 26(6):519–535, 2007.

[2] Groves Paul. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems, Second Edition.* Artech, 2013.

[3] Timothy D Barfoot. *State estimation for robotics.* Cambridge University Press, 2017.

[4] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312, 1996.

[5] Ke Sun, Kartik Mohta, Bernd Pfrommer, Michael Watterson, Sikang Liu, Yash Mulgaonkar, Camillo Jose Taylor, and Vijay Kumar. Robust stereo visual inertial odometry for fast autonomous flight. In *International Conference on Robotics and Automation*, volume 3, pages 965–972, 2018.

[6] Yonggen Ling, Tianbo Liu, and Shaojie Shen. Aggressive quadrotor flight using dense visual-inertial fusion. In *2016 IEEE International Conference on Robotics and Automation (ICRA).* IEEE, 2016.

[7] Yulin Yang and Guoquan Huang. Acoustic-inertial underwater navigation. In *2017 IEEE International Conference on Robotics and Automation (ICRA).* IEEE, 2017.

[8] Helen Oleynikova, Michael Burri, Simon Lynen, and Roland Siegwart. Real-time visual-inertial localization for aerial and ground robots. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE, 2015.

[9] Jeremy Ma, Max Bajracharya, Sara Susca, Larry Matthies, and Matt Malchano. Real-time pose estimation of a dynamic quadruped in gps-denied environments for 24-hour operation. *The International Journal of Robotics Research*, 35(6):631–653, 2016.

[10] Anastasios I Mourikis, Nikolas Trawny, Stergios I Roumeliotis, Andrew E Johnson, Adnan Ansar, and Larry Matthies. Vision-aided inertial navigation for spacecraft entry, descent, and landing. *IEEE Transactions on Robotics*, 25(2):264–280, 2009.

[11] Kejian Wu, Ahmed Ahmed, Georgios Georgiou, and Stergios Roumeliotis. A square root inverse filter for efficient vision-aided inertial navigation on mobile devices. In *Robotics: Science and Systems XI.* Robotics: Science and Systems Foundation, 2015.

[12] Anastasios I. Mourikis and Stergios I. Roumeliotis. A multi-state constraint kalman filter for vision-aided inertial navigation. In *IEEE International Conference on Robotics and Automation.* IEEE, 2007.

[13] Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018.

[14] Kevin Eckenhoff, Patrick Geneva, and Guoquan Huang. *High-Accuracy Preintegration for Visual-Inertial Navigation*, pages 48–63. Springer International Publishing, 2020.

[15] Dan Simon. *Optimal state estimation: Kalman, H infinity, and nonlinear approaches.* John Wiley & Sons, 2006.

[16] J.A. Castellanos, R. Martinez-Cantin, J.D. Tardós, and J. Neira. Robocentric map joining: Improving the consistency of ekf-slam. *Robotics and Autonomous Systems*, 55(1):21–29, 2007.

[17] José A Castellanos, José Neira, and Juan D Tardós. Limits to the consistency of ekf-based slam. *IFAC Proceedings Volumes*, 37(8):716–721, 2004.

[18] Jonathan Kelly, Nicholas Roy, and Gaurav S. Sukhatme. Determining the time delay between inertial and visual sensor measurements. *IEEE Transactions on Robotics*, 30(6):1514–1523, 2014.

[19] Mingyang Li and Anastasios I. Mourikis. Online temporal calibration for camera–imu systems: Theory and algorithms. *The International Journal of Robotics Research*, 33(7):947–964, 2014.

[20] Tong Qin and Shaojie Shen. Online temporal calibration for monocular visual-inertial systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018.

[21] Yulin Yang, Patrick Geneva, and Xingxing Zuo. Online self-calibration for visual-inertial navigation systems: Models, analysis and degeneracy. *arXiv preprint arXiv:2201.09170*, 2022.

[22] Janosch Nikolic, Joern Rehder, Michael Burri, Pascal Gohl, Stefan Leutenegger, Paul T. Furgale, and Roland Siegwart. A synchronized visual-inertial sensor system with fpga pre-processing for accurate real-time slam. In *IEEE International Conference on Robotics and Automation*. IEEE, 2014.

[23] Luca Carlone, Zsolt Kira, Chris Beall, Vadim Indelman, and Frank Dellaert. Eliminating conditionally independent sets in factor graphs: A unifying perspective based on smart factors. In *IEEE International Conference on Robotics and Automation*. IEEE, 2014.

[24] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International Conference on Computer Vision*. IEEE, 2011.

[25] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. *BRIEF: Binary Robust Independent Elementary Features*, pages 778–792. Springer Berlin Heidelberg, 2010.

[26] Paul Furgale. Representing robot pose: The good, the bad, and the ugly. In *workshop on Lessons Learned from Building Complex Systems, IEEE International Conference on Robotics and Automation (ICRA). http://paulfurgale. info/news/2*, volume 14, page 9, 2014.

[27] J. Sola, A. Monin, M. Devy, and T. Lemaire. Undelayed initialization in bearing only slam. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2005.

[28] José Marıa Martinez Montiel, Javier Civera, and Andrew J Davison. Unified inverse depth parametrization for monocular slam. In *Robotics: Science and Systems*, 2006.

[29] J. Civera, A.J. Davison, and J. Montiel. Inverse depth parametrization for monocular slam. *IEEE Transactions on Robotics*, 24(5):932–945, 2008.

[30] Joan Solà, Teresa Vidal-Calleja, Javier Civera, and José María Martínez Montiel. Impact of landmark parametrization on monocular ekf-slam with points and lines. *International Journal of Computer Vision*, 97(3):339–368, 2012.

[31] Joan Solà. Consistency of the monocular ekf-slam algorithm for three different landmark parametrizations. In *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010.

[32] Jon Zubizarreta, Iker Aguinaga, and Jose Maria Martinez Montiel. Direct sparse mapping. *IEEE Transactions on Robotics*, 36(4):1363–1370, 2020.

[33] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[34] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *European Conference on Computer Vision*, pages 834–849, 2014.

[35] Paul Bergmann, Rui Wang, and Daniel Cremers. Online photometric calibration of auto exposure video for realtime visual odometry and slam. *IEEE Robotics and Automation Letters*, 3(2):627–634, 2017.

[36] M.D. Grossberg and S.K. Nayar. Modeling the space of camera response functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1272–1282, 2004.

[37] Juan D. Tardós, José Neira, Paul M. Newman, and John J. Leonard. Robust mapping and localization in indoor environments using sonar data. *The International Journal of Robotics Research*, 21(4):311–330, 2002.

[38] N. Barbour and G. Schmidt. Inertial sensor technology trends. *IEEE Sensors Journal*, 1(4):332–339, 2001.

[39] Naser El-Sheimy and Ahmed Youssef. Inertial sensors technologies for navigation applications: state of the art and future trends. *Satellite Navigation*, 1(1), 2020.

[40] Joern Rehder, Janosch Nikolic, Thomas Schneider, Timo Hinzmann, and Roland Siegwart. Extending kalibr: Calibrating the extrinsics of multiple imus and of individual axes. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4304–4311. IEEE, 2016.

[41] Christian Krebs. Generic imu-camera calibration algorithm: Influence of imu-axis on each other. *Autonomous Systems Lab, ETH Zurich, Tech. Rep*, 2012.

[42] Oliver J Woodman. An introduction to inertial navigation. Report, University of Cambridge, Computer Laboratory, 2007.

[43] Leslie Barreda Pupo. *Characterization of errors and noises in MEMS inertial sensors using Allan variance method.* Thesis, Polytechnic University of Catalonia, 2016.

[44] Naser El-Sheimy, Haiying Hou, and Xiaoji Niu. Analysis and modeling of inertial sensors using allan variance. *IEEE Transactions on Instrumentation and Measurement*, 57(1):140–149, 2008.

[45] Todd Lupton and Salah Sukkarieh. Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions. *IEEE Transactions on Robotics*, 28(1):61–76, 2012.

[46] Michael Boyle. The integration of angular velocity. *Advances in Applied Clifford Algebras*, 27(3):2345–2374, 2017.

[47] Nikolas Trawny and Stergios I. Roumeliotis. Indirect kalman filter for 3d attitude estimation. *University of Minnesota, Dept. of Comp. Sci. and Eng., Tech. Rep*, 2:2005, 2005.

[48] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. Imu preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation. In *Robotics: Science and Systems.* Georgia Institute of Technology, 2015.

[49] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. On-manifold preintegration for real-time visual-inertial odometry. *IEEE Transactions on Robotics*, 33(1):1–21, 2017.

[50] Kevin Eckenhoff, Patrick Geneva, and Guoquan Huang. Closed-form preintegration methods for graph-based visual–inertial navigation. *The International Journal of Robotics Research*, 38(5):563–586, 2019.

[51] Joan Solà. Quaternion kinematics for the error-state kalman filter. *arXiv pre-print server*, 2017.

[52] Zichao Zhang, Henri Rebecq, Christian Forster, and Davide Scaramuzza. Benefit of large field-of-view cameras for visual odometry. In *2016 IEEE International Conference on Robotics and Automation (ICRA).* IEEE, 2016.

[53] Simon Klenk, Jason Chui, Nikolaus Demmel, and Daniel Cremers. Tum-vie: The tum stereo visual-inertial event dataset. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 8601–8608. IEEE, 2021.

[54] Friedrich Fraundorfer and Davide Scaramuzza. Visual odometry: Part i: The first 30 years and fundamentals. *IEEE Robotics and Automation Magazine*, 18(4):80–92, 2011.

[55] Richard A. Newcombe, Steven J. Lovegrove, and Andrew J. Davison. Dtam: Dense tracking and mapping in real-time. In *International Conference on Computer Vision*, pages 2320–2327, 2011.

[56] Christian Kerl, Jurgen Sturm, and Daniel Cremers. Dense visual slam for rgb-d cameras. In *Intelligent Robots and Systems*, pages 2100–2106, 2013.

[57] Christian Kerl, Jurgen Sturm, and Daniel Cremers. Robust odometry estimation for rgb-d cameras. In *2013 IEEE International Conference on Robotics and Automation.* IEEE, 2013.

[58] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *International Conference on Robotics and Automation*, pages 15–22, 2014.

[59] Lukas Von Stumberg, Vladyslav Usenko, and Daniel Cremers. Direct sparse visual-inertial odometry using dynamic marginalization. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018.

[60] Michael Bloesch, Michael Burri, Sammy Omari, Marco Hutter, and Roland Siegwart. Iterated extended kalman filter based visual-inertial odometry using direct photometric feedback. *The International Journal of Robotics Research*, 36(10):1053–1072, 2017.

[61] Christian Forster, Zichao Zhang, Michael Gassner, Manuel Werlberger, and Davide Scaramuzza. Svo: Semidirect visual odometry for monocular and multicamera systems. *IEEE Transactions on Robotics*, 33(2):249–265, 2017.

[62] Yair Weiss and William T. Freeman. Correctness of belief propagation in gaussian graphical models of arbitrary topology. *Neural Computation*, 13(10):2173–2200, 2001.

[63] Sebastian Thrun, Yufeng Liu, Daphne Koller, Andrew Y. Ng, Zoubin Ghahramani, and Hugh Durrant-Whyte. Simultaneous localization and mapping with sparse extended information filters. *The International Journal of Robotics Research*, 23(7-8):693–716, 2004.

[64] Nicholas Carlevaris-Bianco, Michael Kaess, and Ryan M. Eustice. Generic node removal for factor-graph slam. *IEEE Transactions on Robotics*, 30(6):1371–1385, 2014.

[65] Mladen Mazuran, Wolfram Burgard, and Gian Diego Tipaldi. Nonlinear factor recovery for long-term slam. *The International Journal of Robotics Research*, 35(1-3):50–72, 2016.

[66] Michael Bloesch, Sammy Omari, Marco Hutter, and Roland Siegwart. Robust visual inertial odometry using a direct ekf-based approach. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 298–304, 2015.

[67] Patrick Geneva, Kevin Eckenhoff, Woosik Lee, Yulin Yang, and Guoquan Huang. Openvins: A research platform for visual-inertial estimation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020.

[68] Stefan Leutenegger, Simon Lynen, Michael Bosse, Roland Siegwart, and Paul Furgale. Keyframe-based visual–inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, 34(3):314–334, 2015.

[69] Vladyslav Usenko, Nikolaus Demmel, David Schubert, Jorg Stuckler, and Daniel Cremers. Visual-inertial mapping with non-linear factor recovery. *IEEE Robotics and Automation Letters*, 5(2):422–429, 2020.

[70] U. Frese. A proof for the approximate sparsity of slam information matrices. In *IEEE International Conference on Robotics and Automation*. IEEE, 2005.

[71] Ryan M Eustice. *Large-area visually augmented navigation for autonomous underwater vehicles*. Thesis, Massachusetts Institute of Technology, 2005.

[72] F. R. Kschischang, B. J. Frey, and H. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001.

[73] H. Loeliger. An introduction to factor graphs. *IEEE Signal Processing Magazine*, 21(1):28–41, 2004.

[74] Sebastian Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, 2002.

[75] Kevin Murphy and Stuart Russell. *Rao-Blackwellised particle filtering for dynamic Bayesian networks*, pages 499–515. Springer, 2001.

[76] Michael Montemerlo, Sebastian Thrun, Daphne Koller, and Ben Wegbreit. Fastslam: A factored solution to the simultaneous localization and mapping problem. *Aaai/iaai*, 593598, 2002.

[77] Michael Montemerlo, Sebastian Thrun, Daphne Koller, and Ben Wegbreit. Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *Int. Joint Conf. Artif. Intell.*, volume 3, pages 1151–1156, 2003.

[78] Ethan Eade and Tom Drummond. Scalable monocular slam. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1 (CVPR'06)*. IEEE, 2006.

[79] G. Grisetti, C. Stachniss, and W. Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters. *ieee transactions on robotics*, 23(1):34–46, 2007.

[80] Teddy Yap, Mingyang Li, Anastasios I. Mourikis, and Christian R. Shelton. A particle filter for monocular vision-aided odometry. In *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011.

[81] Fredrik Gustafsson. Particle filter theory and practice with positioning applications. *IEEE Aerospace and Electronic Systems Magazine*, 25(7):53–82, 2010.

[82] Zeyneb Kurt-Yavuz and Sirma Yavuz. A comparison of ekf, ukf, fastslam2.0, and ukf-based fastslam algorithms. In *2012 IEEE 16th International Conference on Intelligent Engineering Systems (INES)*. IEEE, 2012.

[83] Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proceedings Ninth IEEE International Conference on Computer Vision*. IEEE, 2003.

[84] Kwang Lee, W Wijesoma, and Javier Guzman. On the observability and observability analysis of slam. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006.

[85] Guoquan Huang, Anastasios I Mourikis, and Stergios I Roumeliotis. Genearalized analysis and improvement of the consistency for ekf-based slam. *University of Minnesota*, pages 2008–0001, 2008.

[86] Guoquan P. Huang, Anastasios I. Mourikis, and Stergios I. Roumeliotis. *A First-Estimates Jacobian EKF for Improving SLAM Consistency*, pages 373–382. Springer Berlin Heidelberg, 2009.

[87] Guoquan P. Huang, Anastasios I. Mourikis, and Stergios I. Roumeliotis. Observability-based rules for designing consistent ekf slam estimators. *The International Journal of Robotics Research*, 29(5):502–528, 2010.

[88] Joel A Hesch, Dimitrios G Kottas, Sean L Bowman, and Stergios I Roumeliotis. Observability-constrained vision-aided inertial navigation. *University of Minnesota, Dept. of Comp. Sci. & Eng., MARS Lab, Tech. Rep*, 1:6, 2012.

[89] Joel A. Hesch, Dimitrios G. Kottas, Sean L. Bowman, and Stergios I. Roumeliotis. Consistency analysis and improvement of vision-aided inertial navigation. *IEEE Transactions on Robotics*, 30(1):158–176, 2014.

[90] Dimitrios G. Kottas, Joel A. Hesch, Sean L. Bowman, and Stergios I. Roumeliotis. *On the Consistency of Vision-Aided Inertial Navigation*, pages 303–317. Springer International Publishing, 2013.

[91] Martin Brossard, Silvere Bonnabel, and Axel Barrau. Invariant kalman filtering for visual inertial slam. In *2018 21st International Conference on Information Fusion (FUSION)*. IEEE, 2018.

[92] Martin Brossard, Silvere Bonnabel, and Axel Barrau. Unscented kalman filter on lie groups for visual inertial odometry. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018.

[93] Axel Barrau and Silvère Bonnabel. Invariant kalman filtering. *Annual Review of Control, Robotics, and Autonomous Systems*, 1(1):237–257, 2018.

[94] B.M. Bell and F.W. Cathey. The iterated kalman filter update as a gauss-newton method. *IEEE Transactions on Automatic Control*, 38(2):294–297, 1993.

[95] G Grisetti, R Kummerle, C Stachniss, and W Burgard. A tutorial on graph-based slam. *IEEE Intelligent Transportation Systems Magazine*, 2(4):31–43, 2010.

[96] 2022.

[97] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. G¡sup¿2¡/sup¿o: A general framework for graph optimization. In *2011 IEEE International Conference on Robotics and Automation*, pages 3607–3613, 2011.

[98] Frank Dellaert. Factor graphs and gtsam: A hands-on introduction. Report, Georgia Institute of Technology, 2012.

[99] Raul Mur-Artal and Juan D. Tardos. Visual-inertial monocular slam with map reuse. *IEEE Robotics and Automation Letters*, 2(2):796–803, 2017.

[100] Jerry Hsiung, Ming Hsiao, Eric Westman, Rafael Valencia, and Michael Kaess. Information sparsification in visual-inertial odometry. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018.

[101] Antoni Rosinol, Marcus Abate, Yun Chang, and Luca Carlone. Kimera: an open-source library for real-time metric-semantic localization and mapping. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1689–1696. IEEE, 2020.

[102] Lukas Von Stumberg and Daniel Cremers. Dm-vio: Delayed marginalization visual-inertial odometry. *IEEE Robotics and Automation Letters*, 7(2):1408–1415, 2022.

[103] M. Kaess, A. Ranganathan, and F. Dellaert. isam: Incremental smoothing and mapping. *IEEE Transactions on Robotics*, 24(6):1365–1378, 2008.

[104] Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John J Leonard, and Frank Dellaert. isam2: Incremental smoothing and mapping using the bayes tree. *The International Journal of Robotics Research*, 31(2):216–235, 2012.

[105] Joan Solà, Jeremie Deray, and Dinesh Atchuthan. A micro lie theory for state estimation in robotics. *arXiv preprint arXiv:1812.01537*, 2020.

[106] Timothy D. Barfoot and Paul T. Furgale. Associating uncertainty with three-dimensional poses for use in estimation problems. *IEEE Transactions on Robotics*, 30(3):679–693, 2014.

[107] Joshua G. Mangelson, Maani Ghaffari, Ram Vasudevan, and Ryan M. Eustice. Characterizing the uncertainty of jointly distributed poses in the lie algebra. *IEEE Transactions on Robotics*, 36(5):1371–1388, 2020.

[108] R. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal on Robotics and Automation*, 3(4):323–344, 1987.

[109] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.

[110] C. Loop and Zhengyou Zhang. Computing rectifying homographies for stereo vision. In *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*. IEEE Comput. Soc, 1999.

[111] Paul Furgale, Joern Rehder, and Roland Siegwart. Unified temporal and spatial calibration for multi-sensor systems. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013.

[112] E. Olson, J. Leonard, and S. Teller. Fast iterative alignment of pose graphs with poor initial estimates. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006*. IEEE, 2006.

[113] D. Nister. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756–770, 2004.

[114] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293(5828):133–135, 1981.

[115] Ezio Malis and Manuel Vargas. *Deeper understanding of the homography decomposition for vision-based control*. Thesis, INRIA, 2007.

[116] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):930–943, 2003.

[117] Laurent Kneip, Davide Scaramuzza, and Roland Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *CVPR 2011*. IEEE, 2011.

[118] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision*, 81(2):155–166, 2009.

[119] Laurent Kneip and Paul Furgale. Opengv: A unified and generalized approach to real-time calibrated geometric vision. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014.

[120] Tue-Cuong Dong-Si and Anastasios I Mourikis. Closed-form solutions for vision-aided inertial navigation. Technical report, University of California, 2011.

[121] Tue-Cuong Dong-Si and Anastasios I. Mourikis. Estimator initialization in vision-aided inertial navigation with unknown camera-imu calibration. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012.

[122] Agostino Martinelli. Closed-form solution of visual-inertial structure from motion. *International Journal of Computer Vision*, 106(2):138–152, 2014.

[123] Carlos Campos, José MM Montiel, and Juan D Tardós. Inertial-only optimization for visual-inertial initialization. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 51–57. IEEE, 2020.

[124] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012.

[125] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 573–580, 2012.

[126] Zichao Zhang and Davide Scaramuzza. A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018.

[127] Yaakov Bar-Shalom, X Rong Li, and Thiagalingam Kirubarajan. *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.

[128] Jeffrey Delmerico and Davide Scaramuzza. A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018.

[129] Thien-Minh Nguyen, Shenghai Yuan, Muqing Cao, Yang Lyu, Thien H Nguyen, and Lihua Xie. Ntu viral: A visual-inertial-ranging-lidar dataset, from an aerial vehicle viewpoint. *The International Journal of Robotics Research*, 41(3):270–280, 2022.

[130] Lintong Zhang, Marco Camurri, and Maurice Fallon. Multi-camera lidar inertial extension to the newer college dataset. *arXiv preprint arXiv:2112.08854*, 2021.

[131] Lukas Meyer, Michal Smíšek, Alejandro Fontan Villacampa, Laura Oliva Maza, Daniel Medina, Martin J. Schuster, Florian Steidle, Mallikarjuna Vayugundla, Marcus G. Müller, Bernhard Rebele, Armin Wedler, and Rudolph Triebel. The madmax data set for visual-inertial rover navigation on mars. *Journal of Field Robotics*, 38(6):833–853, 2021.

[132] Michael Helmberger, Kristian Morin, Nitish Kumar, Danwei Wang, Yufeng Yue, Giovanni Cioffi, and Davide Scaramuzza. The hilti slam challenge dataset. *arXiv preprint arXiv:2109.11316*, 2021.

[133] Patrick Wenzel, Rui Wang, Nan Yang, Qing Cheng, Qadeer Khan, Lukas von Stumberg, Niclas Zeller, and Daniel Cremers. 4seasons: A cross-season dataset for multi-weather slam in autonomous driving. In *DAGM German Conference on Pattern Recognition*, pages 404–417. Springer, 2020.

[134] Jeffrey Delmerico, Titus Cieslewski, Henri Rebecq, Matthias Faessler, and Davide Scaramuzza. Are we ready for autonomous drone racing? the uzh-fpv drone racing dataset. In *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019.

[135] Jinyong Jeong, Younggun Cho, Young-Sik Shin, Hyunchul Roh, and Ayoung Kim. Complex urban dataset with multi-level sensors from highly diverse urban environments. *The International Journal of Robotics Research*, 38(6):642–657, 2019.

[136] David Schubert, Thore Goll, Nikolaus Demmel, Vladyslav Usenko, Jörg Stückler, and Daniel Cremers. The tum vi benchmark for evaluating visual-inertial odometry. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 1680–1687. IEEE, 2018.

[137] Martin Miller, Soon-Jo Chung, and Seth Hutchinson. The visual–inertial canoe dataset. *The International Journal of Robotics Research*, 37(1):13–20, 2018.

[138] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The oxford robotcar dataset. *The International Journal of Robotics Research*, 36(1):3–15, 2017.

[139] Nicholas Carlevaris-Bianco, Arash K Ushani, and Ryan M Eustice. University of michigan north campus long-term vision and lidar dataset. *The International Journal of Robotics Research*, 35(9):1023–1035, 2016.

[140] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W. Achtelik, and Roland Siegwart. The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35(10):1157–1163, 2016.

[141] Hauke Strasdat, José MM Montiel, and Andrew J Davison. Visual slam: why filter? *Image and Vision Computing*, 30(2):65–77, 2012.

[142] Hauke Strasdat, J M M Montiel, and Andrew J Davison. Real-time monocular slam: Why filter? In *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010.

[143] Hauke Strasdat, Andrew J. Davison, J.M.M. Montiel, and Kurt Konolige. Double window optimisation for constant time visual slam. In *2011 International Conference on Computer Vision*. IEEE, 2011.

[144] Martin A. Fischler and Robert C. Bolles. Random sample consensus. *Communications of the ACM*, 24(6):381–395, 1981.

[145] Diego Ortin and José Marıa Martınez Montiel. Indoor robot motion based on monocular images. *Robotica*, 19(3):331–342, 2001.

[146] Davide Scaramuzza. 1-point-ransac structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints. *International Journal of Computer Vision*, 95(1):74–85, 2011.

[147] Davide Scaramuzza. Performance evaluation of 1-point-ransac visual odometry. *Journal of Field Robotics*, 28(5):792–811, 2011.

[148] Davide Scaramuzza, Friedrich Fraundorfer, and Roland Siegwart. Real-time monocular visual odometry for on-road vehicles with 1-point ransac. In *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009.

[149] Deok-Hwa Kim, Seung-Beom Han, and Jong-Hwan Kim. *Visual Odometry Algorithm Using an RGB-D Sensor and IMU in a Highly Dynamic Environment*, pages 11–26. Springer International Publishing, 2015.

[150] Myung Hwangbo, Jun-Sik Kim, and Takeo Kanade. Inertial-aided klt feature tracking for a moving camera. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009.

[151] Yipu Zhao and Patricio A. Vela. Good feature selection for least squares pose optimization in vo/vslam. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018.

[152] Yipu Zhao and Patricio A. Vela. Good feature matching: Toward accurate, robust vo/vslam with low latency. *IEEE Transactions on Robotics*, 36(3):657–675, 2020.

[153] Kirk Mactavish and Timothy D. Barfoot. At all costs: A comparison of robust cost functions for camera correspondence outliers. In *2015 12th Conference on Computer and Robot Vision*. IEEE, 2015.

[154] Heng Yang, Pasquale Antonante, Vasileios Tzoumas, and Luca Carlone. Graduated non-convexity for robust spatial perception: From non-minimal solvers to global outlier rejection. *IEEE Robotics and Automation Letters*, 5(2):1127–1134, 2020.

[155] Raúl Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. Orb-slam: A versatile and accurate monocular slam system. *ieee transactions on robotics*, 31(5):1147–1163, 2015.

[156] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, 27, 2014.

[157] Fayao Liu, Chunhua Shen, and Guosheng Lin. Deep convolutional neural fields for depth estimation from a single image. In *IEEE conference on computer vision and pattern recognition*, pages 5162–5170, 2015.

[158] Ravi Garg, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European conference on computer vision*, pages 740–756. Springer, 2016.

[159] Nan Yang, Lukas von Stumberg, Rui Wang, and Daniel Cremers. D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1281–1292, 2020.

[160] Lukas Koestler, Nan Yang, Niclas Zeller, and Daniel Cremers. Tandem: Tracking and dense mapping in real-time using deep multi-view stereo. In *Conference on Robot Learning*, pages 34–45, 2022.

[161] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

[162] Changhao Chen, Xiaoxuan Lu, Andrew Markham, and Niki Trigoni. Ionet: Learning to cure the curse of drift in inertial odometry. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[163] Hang Yan, Sachini Herath, and Yasutaka Furukawa. Ronin: Robust neural inertial navigation in the wild: Benchmark, evaluations, and new methods. *arXiv preprint arXiv:1905.12853*, 2019.

[164] Martin Brossard, Axel Barrau, and Silvere Bonnabel. Ai-imu dead-reckoning. *IEEE Transactions on Intelligent Vehicles*, 5(4):585–595, 2020.

[165] Wenxin Liu, David Caruso, Eddy Ilg, Jing Dong, Anastasios I. Mourikis, Kostas Daniilidis, Vijay Kumar, and Jakob Engel. Tlio: Tight learned inertial odometry. *IEEE Robotics and Automation Letters*, 5(4):5653–5660, 2020.

[166] Russell Buchanan, Varun Agrawal, Marco Camurri, Frank Dellaert, and Maurice Fallon. Deep imu bias inference for robust visual-inertial odometry with factor graphs. *IEEE Robotics and Automation Letters*, 8(1):41–48, 2022.

[167] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. *NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis*, pages 405–421. Springer International Publishing, 2020.

[168] Linhui Xiao, Jinge Wang, Xiaosong Qiu, Zheng Rong, and Xudong Zou. Dynamic-slam: Semantic monocular visual localization and mapping based on deep learning in dynamic environment. *Robotics and Autonomous Systems*, 117:1–16, 2019.

[169] Kevin J Doherty, David P Baxter, Edward Schneeweiss, and John J Leonard. Probabilistic data association via mixture models for robust semantic slam. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1098–1104, 2020.

[170] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018.

[171] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020.

[172] Niko Sünderhauf, Oliver Brock, Walter Scheirer, Raia Hadsell, Dieter Fox, Jürgen Leitner, Ben Upcroft, Pieter Abbeel, Wolfram Burgard, Michael Milford, and Peter Corke. The limits and potentials of deep learning for robotics. *The International Journal of Robotics Research*, 37(4-5):405–420, 2018.

[173] Changhao Chen, Bing Wang, Chris Xiaoxuan Lu, Niki Trigoni, and Andrew Markham. A survey on deep learning for localization and mapping: Towards the age of spatial machine intelligence. *arXiv preprint arXiv:2006.12567*, 2020.