Camera-IMU Extrinsic Calibration Quality Monitoring for Autonomous Ground Vehicles

Xuesu Xiao, Yulin Zhang, *Member, IEEE*, Haifeng Li, Hongpeng Wang, *Member, IEEE*, and Binbin Li

***Abstract*—Highly accurate sensor extrinsic calibration is critical for data fusion from multiple sensors, such as camera and Inertial Measurement Unit (IMU) sensor suit. A pre-calibrated extrinsics, however, may no longer be accurate due to external disturbances, e.g., vehicle vibration, which will lead to significant performance deterioration of autonomous vehicles. Existing approaches rely on online recalibration at a fixed frequency regardless of whether the extrinsics have actually been changed or recalibration is needed, which is computationally inefficient. In this letter, we present an approach to monitor extrinsic calibration quality for camera-IMU sensor suite to determine when recalibration is actually necessary. We propose an efficient algorithm to detect robust road image features, utilize IMU data to capture the mismatches of those features, and quantify extrinsic calibration error through three commonly-used error metrics. Our algorithm is demonstrated to be effective using both simulated and real-world data.**

***Index Terms*—Calibration and identification, sensor fusion, SLAM.**

I. Introduction

HETEROGENEOUS sensors with complementary characteristics, such as camera [1]–[5] and Inertial Measurement Unit (IMU) [6], [7], are often used in autonomous vehicles or mobile robots [8]. Their data is usually fused to reduce individual sensor noise and to provide robust vehicle state estimation. For example, camera images contain rich semantic information, but are sensitive to illumination conditions [1]; IMUs stream inertia-based high-frequency ego-motion estimates, which suffer from long-term positional drift [9]. Fusing aforementioned sensor readings efficiently avoids the shortcomings of different sensor modalities [10]. However, the data from different sensors may have distinct coordinate systems, formats, resolutions, etc. To facilitate information fusion from heterogeneous sensors, it is critical to transform the data from one sensor's perspective to the other's. Such relative transformation between two sensors needs to be accurately acquired through a process commonly referred as extrinsic sensor calibration.

Classical extrinsic sensor calibration approaches calibrate the sensors only once (e.g., in factory) since they assume that the extrinsic parameters between different sensors are static during runtime. For systems which are subject to vibrations, part replacements, or accidents, the sensor-to-sensor transformation can change from time to time. For example, as shown in Fig. 1, at time $t = 1$–2 s, the camera image and IMU data correspond as pre-calibrated in factory setting. However, such correspondence does not hold anymore since $t = 3$ s due to environmental disturbances to the vehicle. As a consequence, the transformations, indicated by the colored triangles, may change dynamically. On the other hand, systems such as [11]–[13] perform online calibration to recalibrate the system in the estimation pipeline regardless of whether recalibration is actually necessary, which is time-consuming and computationally inefficient.

To avoid unnecessary online sensor recalibration, this letter focuses on monitoring sensor calibration quality for a sensor suite including a camera and an IMU. To the best of our knowledge, this is the first of its kind that can efficiently monitor the necessity of sensor recalibration without performing the "real" estimation task. Instead of measuring errors in the physical space, we characterize sensor calibration quality in terms of mismatches between corresponding features in the image space by fusing IMU data. By doing so, it avoids the intensive image-world reconstruction cost. One of the key challenges is that not all feature points capture the calibration quality appropriately: the feature points from moving objects introduce significant depth error from the camera, which is hard to be distinguished from the sensor calibration error. As a consequence, we narrow down the set of feature points to the static ones on the road, and develop a polynomial algorithm leveraging geometric properties to identify the mismatches through different error metrics. Experimental results, using both simulated data and the real-world KITTI dataset [14] on challenging scenarios, show that our algorithm is robust to identify poor sensor calibration quality in terms of three well-known error metrics, i.e., Sampson error, residual error, and symmetric epipolar distance.

II. Related Work

Light-weight, low-cost, visual-inertial systems allow accurate state estimation in GPS-denied environments and provide critical environment awareness for autonomous driving [15], [16].

Manuscript received September 9, 2021; accepted February 7, 2022. Date of publication February 16, 2022; date of current version March 1, 2022. This letter was recommended for publication by Associate Editor Ezio Malis and Editor Eric Marchand upon evaluation of the reviewers' comments. This work was supported by the National Natural Science Foundation of China under Grants 61973173 and 91848203, and in part by the National Key Research and Development Project of China under Grant 2019YFB1310601. *(Corresponding author: Binbin Li.)*

Xuesu Xiao and Yulin Zhang are with the Department of Computer Science, University of Texas at Austin, Austin, TX 78712 USA (e-mail: xiao@cs.utexas.edu; yulin@cs.utexas.edu).

Haifeng Li is with Computer Science Department, Civil Aviation University of China, Tianjin 300300, China (e-mail: lihaifeng666@gmail.com).

Hongpeng Wang is with the College of Artificial Intelligence, Nankai University, Tianjin 300350, China (e-mail: hpwang@nankai.edu.cn).

Binbin Li is with the Department of Computer Science and Engineering, Texas A&M University, College Station, TX 77843 USA (e-mail: binbinli@tamu.edu).

Digital Object Identifier 10.1109/LRA.2022.3151970

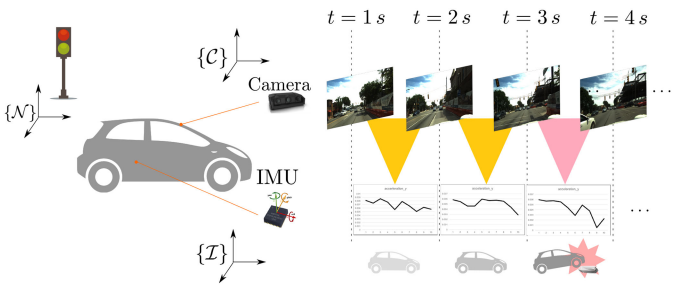


Fig. 1. An autonomous vehicle is on road when fusing both camera images and IMU data. The transformation between camera and IMU, as denoted by the colored triangles, changes under external disturbances from the environment.

Accurate camera-to-IMU extrinsic calibration is an important factor to maintain system performance.

A. Offline Calibration

Among different approaches, offline calibration requires stationary calibration targets before runtime, and moves the sensing suites around to collect IMU measurements and camera images for calibration. Rehder *et al.* [17] have considered individual accelerometer axes and modals for camera measurements to account for motion blur and defocus, which has improved the precision for camera-IMU calibration. Rehder *et al.* [18] have further derived an approach to calibrate a sensor suite comprising one or multiple IMUs and one or more exteroceptive sensors in a single estimator. Fu *et al.* [19] propose to utilize multiple cameras to perform calibration with IMU sensor to achieve smaller lower bound on the covariance of the estimated extrinsic parameters. Furgale *et al.* [20] present a jointly estimation of temporal offset and spatial displacements of different sensors with respect to each other through continuous-time batch estimation. Voges *et al.* [21] present a method for IMU-camera system by determining an interval that is guaranteed to contain the timestamp offset between sensors with bounded error. Different offline calibration approaches conduct calibration before runtime and assumes the pre-calibrated extrinsics remain unchanged during vehicle runtime. However, such assumption does not always hold, because the pre-calibrated displacement between sensors can change due to external disturbances from the environment when the vehicle is moving (see Fig. 1).

B. Online Calibration

To address the aforementioned problem, online calibration methods have been developed to constantly calibrate the extrinsics between sensors during runtime to account for unexpected runtime displacement. Most online approaches employ the observations of naturally occurring point features, in conjunction with the inertial measurements for estimating the camera-IMU transformation, while others use hand-eye calibration [22], [23]. Fleps *et al.* [24] model the IMU-camera calibration problem in a nonlinear optimization framework by modeling the sensors' trajectory. Li *et al.* [25] have proposed a methodology that is able to initialize velocity, gravity, visual scale, and camera-IMU extrinsic calibration on the fly. Leutenegger *et al.* [12] have combined inertial terms and reprojection error from camera image in a single cost function and marginalized old states to bound the algorithm complexity. Geneva *et al.* [26] have introduced the visual processing frontend, full visual-inertial simulator, and modular on-manifold EKF SLAM framework including camera-to-IMU calibration. Although those methods are effective to obtain the camera-to-IMU extrinsics, none of them have an effective monitoring system to quantify the sensor extrinsic quality and identify when recalibration is necessary. Ling *et al.* [27] propose to perform an edge alignment and IMU-aided external check to address the problem of tracking aggressive motions with real-time state estimates and camera-IMU extrinsic calibration. Huang *et al.* [28] apply geometric constraints among stereo cameras and IMU sensor to estimate extrinsic parameters through a three-step process in a coarse-to-fine manner. Walters *et al.* [22] and Heng *et al.* [23] do not use naturally occurring point features, but seek help from manually engineered features, such as hand-eye calibration. Although online calibration methods can assure precise extrinsic calibration during runtime, they usually blindly re-calibrate sensor extrinsics at a fixed and (in most cases) high frequency to recover the system from perturbed extrinsics and incur very high onboard computation burden. Our proposed system, however, reduces such constant re-calibration computation burden by monitoring the sensor extrinsics with an online low-cost light-weight algorithm, and only triggering the high-computation online calibration process when necessary.

III. Problem Definition

The vehicle is equipped with a frontal view camera and an IMU. We have the following assumptions,

- a.1 The camera is pre-calibrated, and the nonlinear distortion of images has been removed.
- a.2 IMU accelerometer and gyroscope measurement noise and random work noise are known.
- a.3 All sensor readings are hardware-synchronized.
- a.4 The initial coordinate system transformations between camera and IMU are known by prior calibration.

All coordinate systems are right hand system, they are shown in Figure 1, and formally defined as follows,

- $\{\mathcal{C}\}$ defines the camera coordinate system with x-axis pointing to the right of the vehicle lateral direction and z-axis pointing forward coinciding with the front-view camera's principal axis. Define $\mathbf{p}_{k,j} = [u\ v]^\intercal$ to be the j-th feature position in image $\mathbf{I}_k$, where (u, v) is the image coordinate, and k refers to the k-th image keyframe.
- $\{\mathcal{I}\}$ defines the IMU coordinate system with x-axis pointing in the vehicle forward direction, y-axis pointing to the left, and z-axis pointing upward[1]. Define $\mathbf{w}_{k,i}$ and $\mathbf{a}_{k,i}$ to be the i-th IMU gyroscope and accelerometer measurements between the keyframe $\mathbf{I}_k$ and $\mathbf{I}_{k+1}$, respectively. Here, $i = 0, 1, \ldots, c$, and c is the IMU data number between consecutive keyimages.

[1]Alignment of $\{\mathcal{C}\}$ and $\{\mathcal{I}\}$ with vehicle axes may vary over time, which is the motivation of the proposed online monitoring system.

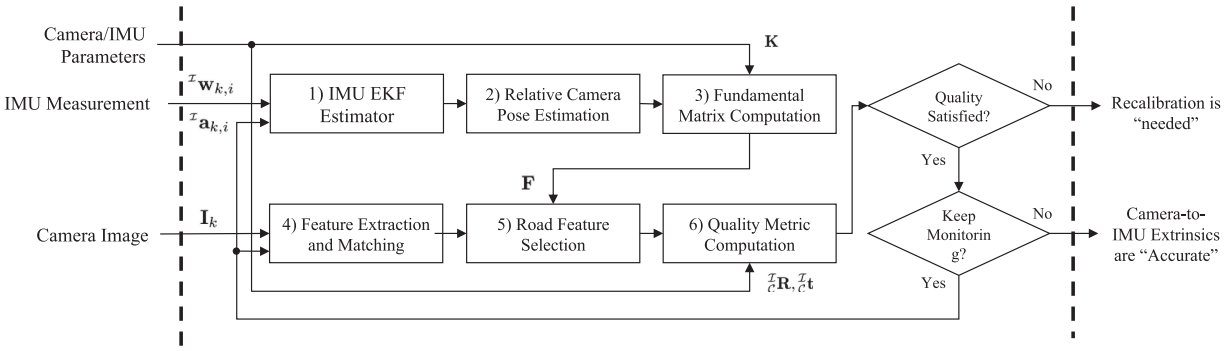


Fig. 2. System diagram: Each block is explained in detail in the corresponding subsection in Section IV with explicit reference back to this diagram.

- $\{\mathcal{N}\}$ defines the navigation coordinate system which overlaps with $\{\mathcal{I}\}$ at the vehicle starting position.
- ${}^{\mathcal{Y}}_{\mathcal{X}}\mathbf{R}$ denotes the rotation matrix from the frame $\{\mathcal{X}\}$ to $\{\mathcal{Y}\}$.
- ${}^{\mathcal{Y}}_{\mathcal{X}}\mathbf{t}$ denotes translation vector from frame $\{\mathcal{X}\}$ to $\{\mathcal{Y}\}$.
- $\check{\mathbf{X}} = [\mathbf{X}^\intercal, 1]^\intercal$ denotes the homogeneous vector, where $\mathbf{X}$ denotes the inhomogeneous counterpart of $\check{\mathbf{X}}$.

We also have $\mathbf{K}$ as the front-view camera intrinsic matrix. Denote the pre-calibrated camera-IMU extrinsic transformation by ${}^{\mathcal{I}}_{\mathcal{C}}\hat{\mathbf{T}}$. ${}^{\mathcal{I}}_{\mathcal{C}}\hat{\mathbf{T}}$ is the rigid body transformation from frame $\{\mathcal{C}\}$ to $\{\mathcal{I}\}$,

$$ {}^{\mathcal{I}}_{\mathcal{C}}\hat{\mathbf{T}} = \begin{bmatrix} {}^{\mathcal{I}}_{\mathcal{C}}\hat{\mathbf{R}} & {}^{\mathcal{I}}_{\mathcal{C}}\hat{\mathbf{t}} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix}. $$

Denote the ground-truth rotation matrix and translation vector for camera w.r.t. IMU to be ${}^{\mathcal{I}}_{\mathcal{C}}\mathbf{R} = \Delta^{\mathcal{I}}_{\mathcal{C}}\mathbf{R}\,{}^{\mathcal{I}}_{\mathcal{C}}\hat{\mathbf{R}}$ and ${}^{\mathcal{I}}_{\mathcal{C}}\mathbf{t} = {}^{\mathcal{I}}_{\mathcal{C}}\hat{\mathbf{t}} + \Delta^{\mathcal{I}}_{\mathcal{C}}\mathbf{t}$, respectively, where ${}^{\mathcal{I}}_{\mathcal{C}}\hat{\mathbf{R}}$ and ${}^{\mathcal{I}}_{\mathcal{C}}\mathbf{t}$ are the pre-calibrated camera-to-IMU extrinsics before runtime, and $\Delta^{\mathcal{I}}_{\mathcal{C}}\mathbf{R}$ and $\Delta^{\mathcal{I}}_{\mathcal{C}}\mathbf{t}$ are caused by runtime sensor displacement.

Condition 1: The camera-IMU extrinsics are considered to be accurate when $|\angle(\Delta^{\mathcal{I}}_{\mathcal{C}}\mathbf{R})| \prec \Delta\theta \cdot \mathbf{1}$ and $|\Delta^{\mathcal{I}}_{\mathcal{C}}\mathbf{t}| \prec \Delta t \cdot \mathbf{1}$, where $|\angle(\cdot)|$ converts rotation matrices to absolute Euler angle representation, $\Delta\theta$ and Δt are pre-defined threshold variables, $\mathbf{1}$ presents 3×1 vector of ones, and $\prec$ denotes the vector component-wise less than operator such that $\mathbf{u}(l) < \mathbf{v}(l)$ $(\forall l \in \{1, 2, \ldots, n\})$ for vector $\mathbf{u}$ and $\mathbf{v}$.

With the assumptions and notations defined, our problem is defined as follows,

Problem 1: Given camera image $\mathbf{I}_{1:k}$, gyroscope measurement $\mathbf{w}_{1:k,1:c}$, accelerometer measurement $\mathbf{a}_{1:k,1:c}$, and pre-calibrated camera-to-IMU extrinsic matrix ${}^{\mathcal{I}}_{\mathcal{C}}\hat{\mathbf{T}}$, monitor $\Delta^{\mathcal{I}}_{\mathcal{C}}\mathbf{R}$ and $\Delta^{\mathcal{I}}_{\mathcal{C}}\mathbf{t}$ and report when Condition 1 is not satisfied.

IV. Methodology

When an autonomous vehicle is driving on a street, it receives images from the front-view ego camera and IMU measurements with higher frequency. We employ an extended Kalman filter (EKF) based estimator to track the pose of IMU-affixed frame $\{\mathcal{I}\}$ with respect to the world coordinate $\{\mathcal{N}\}$, and obtain the fundamental matrix for neighboring image keyframes [29]. The IMU measurements are processed immediately as they become available to propagate EKF state and covariance. We then perform a two-step approach to obtain features from keyframes on the road: we first utilize Chi-squared hypothesis testing to select road features through epipolar constraints; we then compare the road normal vector to refine the road feature set by utilizing the fundamental matrix computed from the IMU EKF estimator. Finally, due to the lack of ground truth during vehicle runtime (${}^{\mathcal{I}}_{\mathcal{C}}\mathbf{R}$ and ${}^{\mathcal{I}}_{\mathcal{C}}\mathbf{t}$), we use a set of error metrics to approximate the degree to which Condition 1 is violated. (see Fig. 2).

A. Fundamental Matrix Construction From IMU Data

We start with obtaining the fundamental matrix by estimating the relative rotation matrix and translation vector between two neighboring keyframes through an IMU EKF estimator (see box 1 of Fig. 2). Recall $\mathbf{w}_{k,i}$ and $\mathbf{a}_{k,i}$ are the i-th IMU gyroscope and accelerometer measurements between the keyframe $\mathbf{I}_k$ and $\mathbf{I}_{k+1}$, respectively. The IMU state is described by the vector,

$$ {}^{\mathcal{I}}\mathbf{x}_{k,i} = [{}^{\mathcal{I}}_{\mathcal{N}}\mathbf{q}^\intercal_{k,i} \quad \mathbf{b}_g{}^\intercal \quad {}^{\mathcal{N}}\mathbf{v}^\intercal_{k,i} \quad \mathbf{b}_a{}^\intercal \quad {}^{\mathcal{N}}\mathbf{p}^\intercal_{k,i}]^\intercal, \tag{1} $$

where ${}^{\mathcal{I}}_{\mathcal{N}}\mathbf{q}_{k,i}$ is the unit quaternion describing the rotation from navigation frame $\{\mathcal{N}\}$ to IMU frame $\{\mathcal{I}\}$, ${}^{\mathcal{N}}\mathbf{p}_{k,i}$ and ${}^{\mathcal{N}}\mathbf{v}_{k,i}$ are the IMU position and velocity with respect to $\{\mathcal{N}\}$, and $\mathbf{b}_g$ and $\mathbf{b}_a$ are 3×1 vectors that describe the biases affecting the gyroscope and accelerometer measurements, respectively. The IMU biases are modeled as random walk processes, driven by the white Gaussian noise $\mathbf{n}_{wg}$ and $\mathbf{n}_{wa}$, respectively. Let $\mathbf{R}(\cdot)$ be the rotation matrix operator from a quaternion, and we have,

$$ \begin{aligned} {}^{\mathcal{I}}\mathbf{w}_{k,i} &= \mathbf{R}({}^{\mathcal{I}}_{\mathcal{N}}\mathbf{q}_{k,i})^{\mathcal{N}}\mathbf{w}_{k,i} + \mathbf{b}_g + \mathbf{n}_g, \\ {}^{\mathcal{I}}\mathbf{a}_{k,i} &= \mathbf{R}({}^{\mathcal{I}}_{\mathcal{N}}\mathbf{q}_{k,i})({}^{\mathcal{N}}\mathbf{a}_{k,i} - {}^{\mathcal{N}}\mathbf{g}) + \mathbf{b}_a + \mathbf{n}_a, \end{aligned} \tag{2} $$

where ${}^{\mathcal{N}}\mathbf{g}$ is the gravity vector in frame $\{\mathcal{N}\}$, and $\mathbf{n}_g$ and $\mathbf{n}_a$ are zero-mean Gaussian noise process to model the measurement noise, respectively. Considering linearization of error-state is small and remains accurate than the nominal state, we deploy the IMU error-state to be [30],

$$ {}^{\mathcal{I}}\tilde{\mathbf{x}}_{k,i} = [{}^{\mathcal{I}}_{\mathcal{N}}\tilde{\theta}^\intercal_{k,i} \quad \tilde{\mathbf{b}}_g{}^\intercal \quad {}^{\mathcal{N}}\tilde{\mathbf{v}}^\intercal_{k,i} \quad \tilde{\mathbf{b}}_a{}^\intercal \quad {}^{\mathcal{N}}\tilde{\mathbf{p}}^\intercal_{k,i}]^\intercal, \tag{3} $$

where ${}^{\mathcal{I}}_{\mathcal{N}}\tilde{\theta}^{\intercal}_{k,i}$ represents a small angle rotation, and we utilize $\tilde{(\cdot)}$ to present the error in the estimate of a quantity. We have the error-state kinematics for IMU to be,

$$ {}^{\mathcal{I}}\dot{\tilde{\mathbf{x}}}_{k,i} = \mathbf{A}\, {}^{\mathcal{I}}\tilde{\mathbf{x}}_{k,i} + \mathbf{G}\mathbf{n}, \tag{4}$$

where $\mathbf{n} = [\mathbf{n}_g^{\intercal}\ \mathbf{n}_{wg}^{\intercal}\ \mathbf{n}_a^{\intercal}\ \mathbf{n}_{wa}^{\intercal}\ ]^{\intercal}$ is the system noise. Through (4), we obtain the nominal state of IMU through integration [30]. Define $\mathbf{I}_3$ to be the 3×3 identify matrix, $\mathbf{0}$ to be the 3×3 zero matrix, and $\lfloor\cdot\rfloor$ to be the skew symmetric matrix operator of a vector. We also have the following,

$$\mathbf{A} = \begin{bmatrix} -\lfloor\tilde{\mathbf{w}}\rfloor_{\times} & -\mathbf{I}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\mathbf{R}^{\intercal}({}^{\mathcal{I}}_{\mathcal{N}}\tilde{\mathbf{q}})\lfloor\tilde{\mathbf{a}}\rfloor_{\times} & \mathbf{0} & \mathbf{0} & -\mathbf{R}^{\intercal}({}^{\mathcal{I}}_{\mathcal{N}}\tilde{\mathbf{q}}) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_3 & \mathbf{0} & \mathbf{0} \end{bmatrix}, \tag{5}$$

and

$$G = \begin{bmatrix} -\mathbf{I}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_3 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{R}^{\intercal}({}^{\mathcal{I}}_{\mathcal{N}}\tilde{\mathbf{q}}) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_3 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \tag{6}$$

where ${}^{\mathcal{I}}_{\mathcal{N}}\tilde{\mathbf{q}} = [\frac{1}{2}{}^{\mathcal{I}}_{\mathcal{N}}\tilde{\theta}^{\intercal}_{k,i}1]^{\intercal}$, $\tilde{\mathbf{w}} = {}^{\mathcal{I}}\mathbf{w}_{k,i} - \hat{\mathbf{b}}_g$, and $\tilde{\mathbf{a}} = {}^{\mathcal{I}}\mathbf{a}_{k,i} - \hat{\mathbf{b}}_a$. Here, $\hat{\cdot}$ is the estimation of a variable. The EKF measurement model is,

$$ {}^{\mathcal{I}}\mathbf{z}_{k,i} = \mathbf{H}\, {}^{\mathcal{I}}\tilde{\mathbf{x}}_{k,i} + \mathbf{n}_z, \tag{7}$$

where ${}^{\mathcal{I}}\mathbf{z}_{k,i} = [{}^{\mathcal{I}}_{\mathcal{N}}\tilde{\theta}^{\intercal}_{k,i}\, {}^{\mathcal{N}}\tilde{\mathbf{v}}^{\intercal}_{k,i}]^{\intercal}$ through from IMU reading postprocessing [30], [31], $\mathbf{n}_z$ is the white zero-mean Gaussian noise, and

$$\mathbf{H} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_3 & \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{8}$$

With the error estimator ${}^{\mathcal{I}}\tilde{\mathbf{x}}$ in (3), the true state are ${}^{\mathcal{I}}_{\mathcal{N}}\mathbf{q}_{k,i} = {}^{\mathcal{I}}_{\mathcal{N}}\tilde{\mathbf{q}}_{k,i} \otimes {}^{\mathcal{I}}_{\mathcal{N}}\hat{\mathbf{q}}_{k,i}$ and ${}^{\mathcal{N}}\mathbf{p}_{k,i} = {}^{\mathcal{N}}\hat{\mathbf{p}}_{k,i} + {}^{\mathcal{N}}\tilde{\mathbf{p}}_{k,i}$, where $\otimes$ is the quaternion multiplication. From (4) and (7), we obtain the relative rotation matrix through discrete-time integration, and have the quaternion ${}^{\mathcal{I}}_{\mathcal{N}}\mathbf{q}_{k,0}$ and translation vector ${}^{\mathcal{N}}\mathbf{p}_{k,0}$ at the time index when $\mathbf{I}_k$ is taken. We then obtain the relative rotation matrix and translation vector for IMU,

$$\begin{aligned} {}^{\mathcal{I}}\mathbf{R}^{k}_{k-1} &= \mathbf{R}^{\intercal}({}^{\mathcal{I}}_{\mathcal{N}}\mathbf{q}_{k-1,0})\, \mathbf{R}({}^{\mathcal{I}}_{\mathcal{N}}\mathbf{q}_{k,0}) \\ {}^{\mathcal{I}}\mathbf{t}^{k}_{k-1} &= {}^{\mathcal{I}}\mathbf{R}^{k}_{k-1}({}^{\mathcal{N}}\mathbf{p}_{k,0} - {}^{\mathcal{N}}\mathbf{p}_{k-1,0}), \end{aligned} \tag{9}$$

(see box 2 of Fig. 2). Recall we have the pre-calibrated camera-to-IMU rotation matrix ${}^{\mathcal{I}}_{\mathcal{C}}\hat{\mathbf{R}}$ and translation vector ${}^{\mathcal{I}}_{\mathcal{C}}\hat{\mathbf{t}}$. We then have the relative rotation and translation for keyframe $\mathbf{I}_{k-1}$ w.r.t. $\mathbf{I}_k$ through (9) by,

$$\begin{aligned} {}^{\mathcal{C}}\hat{\mathbf{R}}^{k}_{k-1} &= {}^{\mathcal{I}}_{\mathcal{C}}\hat{\mathbf{R}}\, {}^{\mathcal{I}}\mathbf{R}^{k}_{k-1}\, {}^{\mathcal{I}}_{\mathcal{C}}\hat{\mathbf{R}}^{\intercal}, \\ {}^{\mathcal{C}}\hat{\mathbf{t}}^{k}_{k-1} &= -{}^{\mathcal{C}}\mathbf{R}^{k}_{k-1}\, {}^{\mathcal{I}}_{\mathcal{C}}\hat{\mathbf{t}} + {}^{\mathcal{I}}_{\mathcal{C}}\hat{\mathbf{R}}\, {}^{\mathcal{I}}\mathbf{t}^{k}_{k-1} + {}^{\mathcal{I}}_{\mathcal{C}}\hat{\mathbf{t}}. \end{aligned} \tag{10}$$

With the relative camera pose estimated, we construct the fundamental matrix between two keyframes. Recall that $\mathbf{K}$ is the camera intrinsic matrix. Through (10), we construct the fundamental matrix $\mathbf{F}$ for keyframe $\mathbf{I}_{k-1}$ and $\mathbf{I}_k$ to be [32],

$$\mathbf{F} = (\mathbf{K}^{-1})^{\intercal}\, \lfloor {}^{\mathcal{C}}\hat{\mathbf{t}}^{k}_{k-1}\rfloor_{\times}\, {}^{\mathcal{C}}\hat{\mathbf{R}}^{k}_{k-1}\, \mathbf{K}^{-1}, \tag{11}$$

which helps us establish relation for features between image $\mathbf{I}_{k-1}$ and $\mathbf{I}_k$ (see box 3 of Fig. 2). We still need to extract features from image $\mathbf{I}_{k-1}$ and $\mathbf{I}_k$ to quantify the camera-to-IMU extrinsic matrix quality.

B. Two-Step Road Feature Selection in Camera Image

In urban environment, the features extracted from the road (concrete or asphalt) usually have relative smaller parallax and more regular pattern comparing with features located on moving objects, which makes them a better fit to capture the camera-to-IMU calibration quality. To filter out the features that are not located on the road, we propose a two-step approach: we first apply chi-square test to remove features against our hypothesis, and then utilize road normal vector to further select features through the fundamental matrix in (11) (see boxes 4 and 5 of Fig. 2).

1) Step 1: Recall that $\mathbf{p}_{k,j}$ is the j-th feature position in image $\mathbf{I}_k$ at time k. Let $\mathbf{p}_{k-1,j}$ be in the image $\mathbf{I}_{k-1}$ and the corresponding matched feature position for the point $\mathbf{p}_{k,j}$. We test if $\mathbf{p}_{k-1,j}$ and $\mathbf{p}_{k,j}$ are road features through the Chi-squared test [33]:

- H_0: $\mathbf{p}_{k-1,j}$ and $\mathbf{p}_{k,j}$ are road feature points.
- H_1: Otherwise.

Let $\mathbf{p}^{+}_{k,j}$ be the closest point on the epipolar line $\mathbf{F}\mathbf{p}_{k-1,j}$ to the feature $\mathbf{p}_{k,j}$, namely

$$\mathbf{p}^{+}_{k,j} = \arg\min \|\check{\mathbf{p}}^{\intercal}_{k-1,j}\, \mathbf{F}\, \check{\mathbf{p}}_{k,j}\|, \quad \text{s.t. } \check{\mathbf{p}}^{\intercal}_{k-1,j}\, \mathbf{F}\, \check{\mathbf{p}}^{+}_{k,j} = 0,$$

Therefore the Euclidean distance of the feature point $\mathbf{p}_{k,j}$ to the corresponding epipolar line is $d_j(\mathbf{p}_{k,j}) = \|\mathbf{p}^{+}_{k,j} - \mathbf{p}_{k,j}\|$, where $\|\cdot\|$ is the vector Euclidean norm. Let σ_j be the variance of the distance of road features to its corresponding epipolar line (we manually tuned the variance value and empirically found out that $\sigma_j = 1$ works best for our algorithm). We have $\chi^2 = d_j(\mathbf{p}_{k,j})^2/\sigma^2_j$ and we reject H_0 if $\chi^2 > \chi^2_{1-\beta,2}$ with 2 DoF, where β is the significance level. We then have the road feature set

$$P_k = \{(\mathbf{p}_{k-1,j}, \mathbf{p}_{k,j}) | k = 1, 2, \ldots, n_k, j = 1, 2, \ldots, n_j\}. \tag{12}$$

for feature pairs that pass our testing.

2) Step 2: We then use the road normal vector ${}^{\mathcal{N}}\mathbf{n}$ to further help select the road features in the set P_k. The road normal vector ${}^{\mathcal{N}}\mathbf{n}$ is aligned with the IMU z-axis at time $k = 1$[2]. We then have the road normal vector at time $k-1$ as,

$$ {}^{\mathcal{C}}\mathbf{n}_{k-1} = {}^{\mathcal{I}}_{\mathcal{C}}\hat{\mathbf{R}}^{\intercal}\, \mathbf{R}({}^{\mathcal{I}}_{\mathcal{N}}\mathbf{q}_{k-1,0})\, {}^{\mathcal{N}}\mathbf{n}, \tag{13}$$

expresssed in the camera coordinate.

Here, we derive ${}^{\mathcal{C}}\hat{\mathbf{n}}_{k-1}$, the road normal vector in the camera coordinate when the keyframe $\mathbf{I}_{k-1}$ is taken, if we triangulate the

[2]We assume for autonomous ground vehicles the road surface is mostly even and the flat road surface is approximately perpendicular to the IMU z-axis.

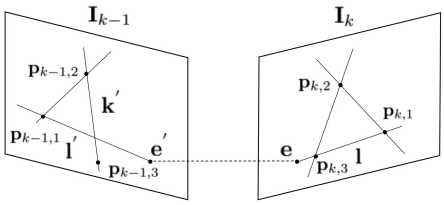


Fig. 3. Road feature selection using epipolar geometry.

road features from P_k and perform road plan fitting in camera coordinate $\{\mathcal{C}\}$. Due to the skew matrix property [34], we have

$$\lfloor \mathbf{F}^{-1}\, {}^{\mathcal{C}}\hat{\mathbf{n}}_{k-1} \rfloor_\times = |\mathbf{F}|\, \mathbf{F}^\intercal\, \lfloor {}^{\mathcal{C}}\hat{\mathbf{n}}_{k-1} \rfloor_\times\, \mathbf{F}, \tag{14}$$

where $|\cdot|$ is the determinant of a matrix. Then we obtain

$$(\mathbf{F}\, \check{\mathbf{p}}_{k-1,j})^\intercal\, \lfloor {}^{\mathcal{C}}\hat{\mathbf{n}}_{k-1} \rfloor_\times\, \mathbf{F}\, \check{\mathbf{p}}_{k-1,j} = 0, \tag{15}$$

by taking feature $\mathbf{p}_{k-1,j}$ into (14) on both sides (see Fig. 3). Let $\mathbf{l} = \mathbf{F}\check{\mathbf{p}}_{k-1,j}$ be the epipolar line on image $\mathbf{I}_k$, and $\mathbf{l}'$ be the corresponding epipolar line of $\mathbf{l}$ on image $\mathbf{I}_{k-1}$, and $\mathbf{k}'$ is any line not passing through the epipole $\mathbf{e}$ of the image $\mathbf{I}_{k-1}$. We further relate $\mathbf{l}$ and $\mathbf{l}'$ by $\mathbf{l} = \mathbf{F}\lfloor \mathbf{k}' \rfloor_\times \mathbf{l}'$. We reorganize (15), and have

$$(\mathbf{l} \times (\mathbf{F}\, \lfloor \mathbf{k}' \rfloor_\times\, \mathbf{l}'))^\intercal\, {}^{\mathcal{C}}\hat{\mathbf{n}}_{k-1} = 0, \tag{16}$$

which helps us obtain the normal vector ${}^{\mathcal{C}}\hat{\mathbf{n}}_{k-1}$ through three lines, and the aforementioned lines can be constructed through image features extracted from keyframe $\mathbf{I}_{k-1}$ and $\mathbf{I}_k$.

We utilize three noncollinear feature points in $\mathbf{I}_{k-1}$ to build the lines $\mathbf{k}$, $\mathbf{l}$, and $\mathbf{l}'$ in (16). Define $\mathbf{p}_{k-1,l}$, $l = 1, 2, 3$, to be non-collinear normalized coordinates of points sampled from the set P_k in (12), and $\mathbf{p}_{k,l}$ be the corresponding feature points in image $\mathbf{I}_k$. We have the unit vector ${}^{\mathcal{C}}\hat{\mathbf{n}}_{k-1}$ by solving the following through singular value decomposition (SVD), eq 17 is shown at the bottom of this page

To segment road features, we define the label function $C({}^{\mathcal{C}}\mathbf{n}_{k-1}, {}^{\mathcal{C}}\hat{\mathbf{n}}_{k-1})$ for feature pairs $\mathbf{p}_{k-1,l}$ and $\mathbf{p}_{k,l}$ to be,

$$C\left({}^{\mathcal{C}}\mathbf{n}_{k-1}, {}^{\mathcal{C}}\hat{\mathbf{n}}_{k-1}\right) = \begin{cases} 1 & \text{if } |{}^{\mathcal{C}}\mathbf{n}_{k-1} \times {}^{\mathcal{C}}\hat{\mathbf{n}}_{k-1}| \le \epsilon, \\ 0 & \text{otherwise.} \end{cases} \tag{18}$$

We claim that $\mathbf{p}_{k-1,1}$, $\mathbf{p}_{k-1,2}$, and $\mathbf{p}_{k-1,3}$ are road features when $C({}^{\mathcal{C}}\mathbf{n}_{k-1}, {}^{\mathcal{C}}\hat{\mathbf{n}}_{k-1})$ is equal to 1; otherwise, at least one of the three points are not located on the road. Here, ϵ is a threshold variable that facilitate road feature selection, and $|\cdot|$ is the vector absolute-value norm. We further refine the matched feature set P_k in (12) by comparing the normal vector ${}^{\mathcal{C}}\mathbf{n}_{k-1}$ with ${}^{\mathcal{C}}\hat{\mathbf{n}}_{k-1}$ in (17), shown at the bottom of this page, and obtain a updated features located on the road as Q_k (see Fig. (4)).

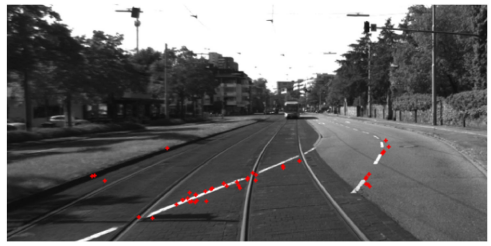


Fig. 4. Road features after the two-step selection (see red points). Best viewed in color.

C. Camera-to-IMU Calibration Quantification

In reality, *Condition* 1 is hard to measure due to the lack of camera-IMU extrinsics ground truth (${}^{\mathcal{I}}_{\mathcal{C}}\mathbf{R}$ and ${}^{\mathcal{I}}_{\mathcal{C}}\mathbf{t}$) when the vehicle is on road. Thus we utilize road features to facilitate online camera-IMU extrinsic calibration monitoring. For now, we have road feature $\mathbf{p}_{k-1,j}$ in image $\mathbf{I}_{k-1}$ and the corresponding matched feature $\mathbf{p}_{k,j}$ in image $\mathbf{I}_k$. We also have the fundamental matrix $\mathbf{F}$ through IMU propagation. To quantify calibration error utilizing the matched feature points, we apply three different commonly-used metrics:

- *Sampson error*:

$$\mu_s = \frac{1}{|Q_k|} \sum_{j=0}^{|Q_k|} \frac{1}{w_j} \left(\check{\mathbf{p}}_{k-1,j}^\intercal\, \mathbf{F}\, \check{\mathbf{p}}_{k,j} \right)^2, \tag{19}$$

  where $w_j = (\mathbf{F}\check{\mathbf{p}}_{k-1,j})_1^2 + (\mathbf{F}\check{\mathbf{p}}_{k-1,j})_2^2 + (\mathbf{F}^\intercal\check{\mathbf{p}}_{k,j})_1^2 + (\mathbf{F}^\intercal\check{\mathbf{p}}_{k,j})_1^2$, and $(\cdot)_m$ denotes the m-th component of a vector.

- *Symmetric epipolar distance*:

$$\mu_d = \frac{1}{|Q_k|} \sum_{j=0}^{|Q_k|} d\left(\mathbf{p}_{k,j}, \mathbf{p}_{k-1,j}\right) + d\left(\bar{\mathbf{p}}_{k,j}, \bar{\mathbf{p}}'_{k-1,j}\right), \tag{20}$$

  assuming the image point measurement satisfies Gaussian distribution, and $\bar{\mathbf{p}}_{k,j}$ and $\bar{\mathbf{p}}'_{k-1,j}$ to be the true correspondences that satisfy $\check{\mathbf{p}}_{k-1,j}^\intercal \mathbf{F} \check{\mathbf{p}}_{k,j} = 0$. Here, $d(\mathbf{p}, \mathbf{q})$ represents the Euclidean distance between the inhomogeneous points represented by $\mathbf{p}$ and $\mathbf{q}$.

- *Residual error*:

$$\mu_r = \frac{1}{|Q_k|} \sum_{j=0}^{|Q_k|} d\left(\mathbf{p}_{k,j}, \mathbf{F}\check{\mathbf{p}}_{k-1,j}\right) + d\left(\mathbf{p}_{k-1,j}, \mathbf{F}^\intercal\check{\mathbf{p}}_{k,j}\right), \tag{21}$$

  between image $\mathbf{I}_{k-1}$ and image $\mathbf{I}_k$. The error is the squared distance between a point's epipolar line and the matching point in the other image (computed for both points of the

$$\begin{aligned}
&((\check{\mathbf{p}}_{k,1} \times \check{\mathbf{p}}_{k,3}) \times (\mathbf{F}\lfloor \check{\mathbf{p}}_{k-1,1} \times \check{\mathbf{p}}_{k-1,2} \rfloor_\times (\check{\mathbf{p}}_{k-1,1} \times \check{\mathbf{p}}_{k-1,3}))^\intercal\, {}^{\mathcal{C}}\hat{\mathbf{n}}_{k-1} = 0 \\
&((\check{\mathbf{p}}_{k,2} \times \check{\mathbf{p}}_{k,3}) \times (\mathbf{F}\lfloor \check{\mathbf{p}}_{k-1,3} \times \check{\mathbf{p}}_{k-1,1} \rfloor_\times (\check{\mathbf{p}}_{k-1,2} \times \check{\mathbf{p}}_{k-1,3}))^\intercal\, {}^{\mathcal{C}}\hat{\mathbf{n}}_{k-1} = 0 \\
&((\check{\mathbf{p}}_{k,1} \times \check{\mathbf{p}}_{k,2}) \times (\mathbf{F}\lfloor \check{\mathbf{p}}_{k-1,3} \times \check{\mathbf{p}}_{k-1,2} \rfloor_\times (\check{\mathbf{p}}_{k-1,1} \times \check{\mathbf{p}}_{k-1,2}))^\intercal\, {}^{\mathcal{C}}\hat{\mathbf{n}}_{k-1} = 0,
\end{aligned} \tag{17}$$

match), averaged over all $|Q_k|$ matches. Here, $d(\mathbf{p}, \mathbf{l})$ is the distance of a point $\mathbf{p}$ to the line $\mathbf{l}$.

We apply an averaging filter of n_w frame sliding window ($n_w = 10$ in our experiments) to filter out high-frequency noises. If the filtered error values are larger than the predetermined threshold value, we will report the camera-to-IMU extrinsic matrix is expected to be recalibrated. Here, ${\mu_+}^2 = \sigma^2 F_{n_w}^{-1}(1-\alpha)$, where F_{n_w} denotes the cumulative χ^2 distribution under n_w DoF and $\alpha = 0.05$ is the significance level.

V. Algorithm

We utilize a KD-tree to store the road feature set P_k. We iterate the tree and select a set of closest points for a feature to check if they are road feature points. It takes $O(|P_k|)$ to check all the feature points in the worst case. However, we use them as seeds to help verify other unchecked features to speed up the process, once a set of three non-collinear road feature points are located. It takes linear time to check the remaining feature points. We then remove paired features in set P_k that do not satisfy (18). We summarize our online camera-to-IMU extrinsic calibration quality monitoring algorithm in Algorithm 1. The computational complexity of our algorithm is,

Lemma 1: Our online camera-to-IMU extrinsic calibration quality monitoring algorithm runs in $O(|Q_k| \log |Q_k|)$.

Comparing with running online camera-to-IMU recalibration algorithm [35]–[37] at a fixed frequency, our light-weight and memory efficient approach simply monitors the camera-IMU extrinsics and triggers the recalibration process only when necessary.

Algorithm 1: Camera-IMU Extrinsic Quality Monitoring.

Input: $\mathbf{I}_{1:k}, {}^{\mathcal{I}}\mathbf{w}_{1:k,1:c}, {}^{\mathcal{I}}\mathbf{a}_{1:k,1:c}, {}^{\mathcal{I}}_{\mathcal{C}}\hat{\mathbf{T}}$
Output: ${}^{\mathcal{I}}_{\mathcal{C}}\hat{\mathbf{T}}$ should be recalibrated or not

```
V = ∅, Q = ∅, and i = 1;                                   O(1)
for k ∈ {2, 3, ..., t} do                                  O(k)
    Obtain ᴵR^k_{k-1} and ᴵt^k_{k-1} using (4) and (7);    O(c)
    Get F in (11) throguh (9) and (10);                    O(1)
    Generate Q_k through the two-step approach;            O(|Q_k|)
    Select (p_{k-1,l}, p_{k,l}) pairs from Q_k;            O(log |Q_k|)
    while (p_{k-1,l}, p_{k,l}) ∉ V and i ≤ |Q_k| do        O(|Q_k|)
        Obtain ᶜn_{k-1} according to (13);                 O(1)
        Compute ᶜn̂_{k-1} (17) by p_{k-1,l} and p_{k,l};    O(1)
        if C(ᶜn_{k-1}, ᶜn̂_{k-1}) = 1 then
            U = {(p_{k-1,l}, p_{k,l}) | l = 1, 2, 3};      O(1)
            Break;                                         O(1)
        V ← V ⋃ {(p_{k-1,l}, p_{k,l}) | l = 1, 2, 3};      O(1)
        Q_k ← Q_k \ (p_{k-1,l}, p_{k,l});                  O(1)
        Select p_{k-1,l} and p_{k,l} pairs from Q_k;       O(log |Q_k|)
        i ← i + 1;                                         O(1)
    i ← 1;                                                 O(1)
    if U ≠ ∅ then
        for (p_{k-1,l}, p_{k,l}) ∈ Q_k \ U do
            Check (18) using p_{k-1,l} and p_{k,l};        O(1)
            if C(ᶜn_{k-1}, ᶜn̂_{k-1}) ≠ 0 then
                U ← U ⋃ {(p_{k-1,l}, p_{k,l}) | l = 1,2,3};   O(1)
        if U ≠ ∅ then
            Obtain error metrics μ (μ_s/μ_d/μ_r);          O(1)
        Use an average filter to computer mean μ̄;          O(1)
        if μ̄ ≥ μ_+ then
            Report "Recalibration is required";            O(1)
            Break;                                         O(1)
        else
            if continue monitoring then
                Continue;                                  O(1)
            else
                Report "Extrinsics are Accurate";          O(1)
                Exit;                                      O(1)
```

VI. Experiments

We implement our algorithm and perform experiments using simulation and the KITTI dataset [14]. The purpose of the experiments is to show that our method can (1) detect when displacement between camera and IMU when occurs, and (2) report the magnitude of such displacement.

A. Simulated Experiments

Our simulation experiments aim to show that our method can detect when displacement between camera and IMU occurs and when such displacement disappears. We use a monocular camera of 1024×720 resolution and 680 pixels focal length. An IMU sensor is right below the camera with a fixed pose relative to the camera. The camera runs at 20 Hz and the IMU outputs 100 Hz high-resolution sensor measurement (see Table I for detailed parameters settings, whose units are aligned with the work in [20]).

We manually inject constant camera-IMU displacement to the originally fixed extrinsics for two separate periods of time and enable the system running in a rectangular city block environment. One short segment appears at frames 70-80, and the other at frame 130–150. We inject displacement of 0.09° for angular error and 0.9 cm for translation error. Fig. 5 illustrates how the three error metrics change in response to the artificially injected

TABLE I
Simulation Settings

$\Delta\theta$	Δt	$\mathbf{n}_{wa}$	$\mathbf{n}_{wg}$	$\mathbf{n}_a$	$\mathbf{n}_g$
0.1	10^{-2}	10^{-3}	10^{-4}	0.1	0.01

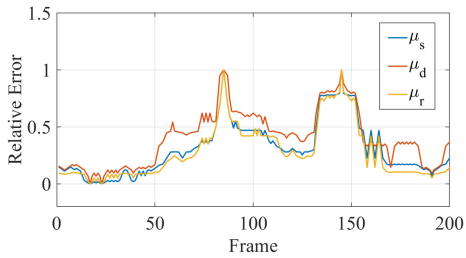


Fig. 5. Camera-IMU error quantification: All three metrics are able to reflect the sensor displacement error (best viewed in color).

camera-to-IMU displacement. All three error metrics significantly increase when camera-IMU displacement is injected and immediately return to normal when the displacement is removed (see the two peaks in Fig. 5). In most cases, Sampson error (blue) is almost indistinguishable from the residual error (yellow). Both

Fig. 6. Real-world driving scenario from the KITTI dataset. Figure labels correspond Table II from the top to the bottom. (a) Road with curbs. (b) Strong sunlight on road. (c) Shadows on roads. (d) Intersections and vehicles with low speed. (e) Parked vehicles. (f) Crowded vehicles with high speed.

TABLE II
EXPERIMENTS ON KITTI DATASET: SAMPSON ERROR μ_s, SYMMETRIC EPIPOLAR DISTANCE μ_d, AND RESIDUAL ERROR μ_r

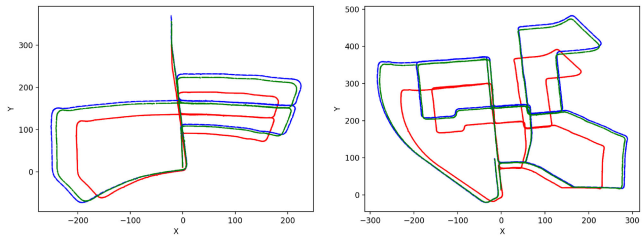
Seq.	Duration	Length	μ_s			μ_d			μ_r		
			Zero	Small	Large	Zero	Small	Large	Zero	Small	Large
(a) 2011_09_26_drive_0002	8 s	81.74 m	0.470	0.674	0.951	0.482	0.820	1.121	0.239	0.481	0.571
(b) 2011_09_26_drive_0005	16 s	69.43 m	0.322	0.412	0.503	0.114	0.533	0.835	0.054	0.121	0.361
(c) 2011_09_26_drive_0056	30 s	419.95 m	0.493	0.659	0.732	0.132	0.582	0.652	0.122	0.327	0.470
(d) 2011_09_26_drive_0104	31 s	246.75 m	0.159	0.353	0.559	0.109	0.325	0.501	0.081	0.115	0.227
(e) 2011_09_26_drive_0106	23 s	114.33 m	0.306	0.429	0.982	0.196	0.353	0.943	0.196	0.237	0.351
(f) 2011_09_29_drive_0004	34 s	254.98 m	0.781	0.811	1.031	0.413	0.589	0.749	0.356	0.466	0.501

have noticeably smaller magnitude than the symmetric epipolar distance (orange), but all three are able to properly reflect the injected extrinsic error.

B. Experiments on the KITTI Dataset

We also use the KITTI dataset [14], which contains camera images and IMU readings with a variety of street scenes captured from a vehicle driving around the city of Karlsruhe (Fig. 6), to evaluate the performance of our camera-to-IMU extrinsic monitoring algorithm in a real-world setting.

We utilize six different sequences of two categories from KITTI dataset, including city trail and road data. Because the KITTI dataset uses relatively accurate camera-to-IMU pre-calibration and does not contain significant displacement between the sensors during data collection, we artificially introduce displacement to the pre-calibrated extrinsics to simulate the scenario where camera-IMU displacement occurs after pre-calibration. We test our system's performance under "zero," "small," and "large" displacement to the pre-calibrated extrinsics to show that the magnitude of the three error metrics is proportional to that of the displacement. In Table II, "zero" means we directly utilize the highly accurate pre-calibrated camera-IMU extrinsics (Euler angles and translation vector) from KITTI (which remain precise during runtime), while we artificially introduce "small" or "large" displacement to the pre-calibration as if the true extrinsics are changed right after pre-calibration. Here, the magnitude of displacement is measured in terms of Euler angles and translation vectors: we randomly sample a constant Euler angle error from $[0.1^\circ, 0.4^\circ)$ and a constant translation error from $[0.01m, 0.03m)$ as "small" displacement, and $[0.4^\circ, 0.8^\circ)$ for Euler angles and $[0.03m, 0.06m)$ for translation vector as "large" displacement. As shown in Table II, all three error metrics directly reflect the magnitude of the artificially introduced camera-IMU displacement, regardless of the sequence duration and length.



(a) 2011_09_30_drive_0018 (b) 2011_10_03_drive_0027

Fig. 7. Camera trajectory on two KITTI trials. Here, the blue curves are the camera ground-truth trajectory, the red ones are the camera trajectory with fixed pose error of extrinsics, and the green ones are the camera trajectory generated from the VIO system with extrinsics re-calibration triggered by our approach (best viewed in color).

We incorporate our online monitoring approach into a full SLAM system [38] and perform testing on two different KITTI trials using the Sampson error (Fig. 7). Similarly, we artificially introduce displacement to the pre-calibrated extrinsics. Our monitoring system continuously outputs high Sampson error value and immediately triggers at the beginning of the SLAM stage on both cases to alert the system to re-calibrate the camera-IMU extrinsic. To simulate re-calibration, the displacement is gradually decreased to 0 within 80 keyframes (totally 1371 keyframes) for KITTI sequence 2011_09_30_drive_0018 until extrinsics are fully re-calibrated, and kept close to zero for

the rest of the SLAM process. The same occurs for KITTI sequence `2011_10_03_drive_0027`. Comparing with camera trajectory with constant "large" displacement, our approach helps generate trajectories that are closer to the ground-truth given initial calibration errors. We also find out that in our experiments all three metrics achieve equivalent status monitoring performance. Due to space limit, we only present camera trajectories of applying Sampson error in Fig. 7. In all cases, our approach can quickly catch the abnormalities of camera-IMU extrinsics, trigger re-calibration process, and monitor metric changes to determine extrinsic parameter accuracy status.

VII. Conclusion and Future Work

In this letter, we introduce an online method to monitor the camera-IMU extrinsic calibration quality to determine when recalibration is necessary. We develop an efficient algorithm to identify the set of feature points on the road using geometric properties. We further characterize sensor calibration error in terms of mismatches of road features in the image space. The effectiveness of our approach is demonstrated in both simulated and real-world dataset under three commonly-used error metrics. Our future work includes extending the algorithm to other types of sensors such as LiDAR-to-camera, and deploying our algorithm on real vehicles.

References

[1] X. Xiao, J. Dufek, T. Woodbury, and R. Murphy, "UAV assisted USV visual navigation for marine mass casualty incident response," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 6105–6110.

[2] X. Xiao, J. Dufek, and R. Murphy, "Visual servoing for teleoperation using a tethered UAV," in *Proc. IEEE Int. Symp. Saf., Secur. Rescue Robot.*, 2017, pp. 147–152.

[3] H. Li, D. Song, Y. Liu, and B. Li, "Automatic pavement crack detection by multi-scale image fusion," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 6, pp. 2025–2036, Jun. 2019.

[4] B. Li, D. Song, H. Li, A. Pike, and P. Carlson, "Lane marking quality assessment for autonomous driving," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 1–9.

[5] B. Li, D. Song, A. Kingery, D. Zheng, Y. Xu, and H. Guo, "Lane marking verification for high definition map maintenance using crowdsourced images," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 2324–2329.

[6] B. Li *et al.*, "Virtual lane boundary generation for human-compatible autonomous driving: A tight coupling between perception and planning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 3733–3739.

[7] X. Xiao, J. Biswas, and P. Stone, "Learning inverse kinodynamics for accurate high-speed off-road navigation on unstructured terrain," *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 6054–6060, Jul. 2021.

[8] H. Li, C. Chou, L. Fan, B. Li, D. Wang, and D. Song, "Toward automatic subsurface pipeline mapping by fusing a ground-penetrating radar and a camera," *IEEE Trans. Automat. Sci. Eng.*, vol. 17, no. 2, pp. 722–734, Apr. 2020.

[9] X. Xiao and S. Zarar, "Machine learning for placement-insensitive inertial motion capture," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 6716–6721.

[10] P. Zhu and W. Ren, "Multi-robot joint visual-inertial localization and 3-d moving object tracking," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 11573–11580.

[11] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2014, pp. 15–22.

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

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

[14] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.

[15] L. Jinyu, Y. Bangbang, C. Danpeng, W. Nan, Z. Guofeng, and B. Hujun, "Survey and evaluation of monocular visual-inertial slam algorithms for augmented reality," *Virtual Reality Intell. Hardware*, vol. 1, no. 4, pp. 386–410, 2019.

[16] S.-P. Li, T. Zhang, X. Gao, D. Wang, and Y. Xian, "Semi-direct monocular visual and visual-inertial SLAM with loop closure detection," *Robot. Auton. Syst.*, vol. 112, pp. 201–210, 2019.

[17] J. Rehder and R. Siegwart, "Camera/IMU calibration revisited," *IEEE Sensors J.*, vol. 17, no. 11, pp. 3257–3268, Jun. 2017.

[18] J. Rehder, J. Nikolic, T. Schneider, T. Hinzmann, and R. Siegwart, "Extending Kalibr: Calibrating the extrinsics of multiple IMUs and of individual axes," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2016, pp. 4304–4311.

[19] B. Fu *et al.*, "High-precision multicamera-assisted camera-IMU calibration: Theory and method," in *IEEE Trans. Instrum. Meas.*, vol. 70, Jan. 2021, Art. no. 1004117, doi: 10.1109/TIM.2021.3051726.

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

[21] R. Voges and B. Wagner, "Timestamp offset calibration for an IMU-Camera system under interval uncertainty," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 377–384.

[22] C. Walters, O. Mendez, S. Hadfield, and R. Bowden, "A robust extrinsic calibration framework for vehicles with unscaled sensors," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 36–42.

[23] L. Heng, B. Li, and M. Pollefeys, "Camodocal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry," in *Proc. IEEE/RSJ Int. Conf. on Intell. Robots Syst.*, 2013, pp. 1793–1800.

[24] M. Fleps, E. Mair, O. Ruepp, M. Suppa, and D. Burschka, "Optimization based IMU camera calibration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2011, pp. 3297–3304.

[25] H. Li, C. Chou, L. Fan, B. Li, D. Wang, and D. Song, "Toward automatic subsurface pipeline mapping by fusing a ground-penetrating radar and a camera," *IEEE Trans. Automat. Sci. Eng.*, vol. 17, no. 2, pp. 722–734, Apr. 2020.

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

[27] Y. Ling, M. Kuse, and S. Shen, "Edge alignment-based visual–inertial fusion for tracking of aggressive motions," *Auton. Robots*, vol. 42, no. 3, pp. 513–528, 2018.

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

[29] G. Younes, D. Asmar, E. Shammas, and J. Zelek, "Keyframe-based monocular SLAM: Design, survey, and future directions," *Robot. Auton. Syst.*, vol. 98, pp. 67–88, 2017.

[30] J. Sola, "Quaternion kinematics for the error-state Kalman filter," *CoRR*, vol. abs/1711.02508, 2017.

[31] M. Kok, J. D. Hol, and T. B. Schön, "Using inertial sensors for position and orientation estimation," *Found. Trends Signal Process.*, vol. 11, no. 1/2, pp. 1–153, 2017.

[32] R Hartley and A. Zisserman, "Multiple view geometry in computer vision," Cambridge Univ. Press, 2003.

[33] S. L. K. Pond and S. V. Muse, "Hyphy: Hypothesis testing using phylogenies," in *Statistical Methods in Molecular Evolution*. Berlin, Germany: Springer, 2005, pp. 125–181.

[34] N. Trawny and S. I. Roumeliotis, "Indirect Kalman filter for 3D attitude estimation," Univ. Minnesota, Dept. Comput. Sci. Eng., Report 2005-002, Jan. 2005.

[35] L. M. Paz, J. D. Tardós, and J. Neira, "Divide and conquer: EKF SLAM in $O(n)$," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1107–1120, Oct. 2008.

[36] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "A quadratic-complexity observability-constrained unscented Kalman filter for SLAM," *IEEE Trans. Robot.*, vol. 29, no. 5, pp. 1226–1243, Oct. 2013.

[37] K. M. Frey, T. J. Steiner, and J. P. How, "Complexity analysis and efficient measurement selection primitives for high-rate graph SLAM," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 6505–6512.

[38] T. Qin, P. Li, and S. Shen, "VINS-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.