Towards Robust Visible Light Positioning Under LED Shortage by Visual-inertial Fusion

Qing Liang, Jiahui Lin and Ming Liu
Department of Electronic and Computer Engineering
Hong Kong University of Science and Technology
Hong Kong, China
qing.liang@connect.ust.hk, eealbertlin@ust.hk, eelium@ust.hk

***Abstract*—Accurate indoor positioning is urgent for critical location-based services. The approach based on visible light communication (VLC) is promising, as it can deliver high accuracy by sharing the LED lighting infrastructure. In this paper, we propose an EKF-based tightly-coupled visual-inertial fusion method for visible light positioning with an IMU and a rolling-shutter camera, aiming for improved positioning robustness under LED shortage. With the proposed method, we can relax the assumption on the minimum number of concurrently observable LEDs required for positioning from three to one. Meanwhile, we can accurately track the sensor pair's global 3D pose in real-time. We evaluate our method by real-world experiments using a prototyping VLC network. The efficacy for VLC beaconing and 3D pose estimation, as well as the robustness under LED shortage, is verified by extensive experiments.**

***Index Terms*—Visible light positioning, VLC, visual-inertial fusion, EKF, IMU**

I. Introduction

Indoor positioning based on visible light communication (VLC) has been a hot topic in recent years [1], [2]. One of the most appealing features is its capability of providing high-accuracy position estimation by exploiting the ubiquitous LED lights in modern buildings, without resorting to any other specialized infrastructure for location services.

A. Motivation

The VLC-based positioning methods in the literature may broadly fall into two categories [1], namely the camera-based and the photodiode-based, according to the optical receiver in use. In particular, the camera-based solutions [3]–[7] have been in favor with both the academia and the industry, for example, due to the high positioning accuracy achievable by imaging geometry and the good compatibility of user devices. The state-of-the-art commercial systems (e.g., Lumicast [5]) can offer centimeter-level accuracy on a commodity smartphone with an inbuilt front-facing camera. Despite the promising performance of existing systems, there remain some practical challenges in this direction.

A most urgent issue arises from the fact that they normally require multiple observations to known LED landmarks at a time for successful position fixing. This is common for vision-only algorithms in the previous works [3]–[7], which can, without loss of generality, come down to solving a perspective-n-point (PnP) problem under geometry-only constraints. As is well known, we need at least three 2D-3D correspondences, e.g., by observing three point-features in a single camera frame, so as to solve the PnP problem with six unknowns.

However, it is reasonably hard to meet such a demanding requirement in practice. According to our observations, the number of visible LEDs by a camera is subject to many factors, such as the LED's deployment density and the geometry layout, the ceiling height, the camera's field-of-view (FoV) and maximum communication distance, the temporal obstructions of line-of-sight (LOS) views caused by the surroundings, and even the camera's own poses relative to those LEDs. As such, the shortage of LEDs[1] can severely deteriorate the performance of vision-only methods in reality. To address this problem, we are motivated to relax the assumption of at least three LED observations for the camera-based solutions.

B. Contributions

In this work, we propose a tightly-coupled visual-inertial fusion method for visible light positioning using a CMOS camera with the aid of a rigidly connected inertial measurement unit (IMU). Specifically, we employ an extended Kalman filter (EKF) for real-time 3D pose estimation (position and orientation) by fusing the relative motion measurements from the IMU with the camera measurements to fixed LED landmarks of known absolute locations. Our EKF-based positioning method can stably track the global poses of the sensor pair, by observing one LED on average in the camera image. Due to the filtering nature, we are also able to cope with the temporal LED outage, i.e., when not a single observation is available over a short period of time. As a result, our method can efficiently relax the originally demanding assumption on the number of concurrently visible LEDs for the vision-only positioning systems. With improved robustness under LED shortage, our method has a good potential for practical

*This work was supported by National Natural Science Foundation of China No. U1713211, the Shenzhen Science, Technology and Innovation Commission (SZSTI) JCYJ20160428154842603 and JCYJ20160401100022706, the Research Grant Council of Hong Kong SAR Government, China, under Project No. 11210017 and No. 21202816, all awarded to Prof. Ming Liu.

[1] In this work, we consider point-source LEDs for positioning, i.e., without utilizing the LED's geometry properties for positioning purposes. Therefore, one LED can only provide one point-feature measurement. The vision-only methods need at least three such LEDs to solve the PnP problem. We call the situation "LED shortage" when the visible LEDs are less than three.

applications in terms of better usability in various indoor environments with LED shortage problems.

We highlight the following contributions as:

- An EKF-based visual-inertial fusion method is proposed for robust visible light positioning under LED shortage in a tightly-coupled manner. We relax the assumption on the minimum number of simultaneously observable LEDs efficiently from three to one, and meanwhile, we can track the global 3D pose accurately in real-time.
- The method is evaluated in a real-world environment with a prototyping VLC network composed of dozens of customized LEDs. The efficacy for VLC beaconing and accurate 3D pose tracking, as well as the robustness under LED shortage, is verified with extensive experiments.

C. Organization

The remainder of this paper is organized as follows. Section II introduces the related works. Section III explains our methodology. Section IV presents the experimental evaluation results and Section V concludes this paper.

II. Related Works

A. VLC using Rolling-shutter Cameras

As we know, LEDs can transmit data over the air by directly modulating the light intensity at a high frequency. The fast-changing light components are invisible to human eyes but perceivable by a normal CMOS camera with the rolling-shutter effect [8]. The temporally-varying intensity signals (1D) from the LED transmitter are mapped to spatially-varying strip patterns (2D) on the camera image. By image processing and the subsequent VLC decoding, we can recover the embedded messages from the captured strip patterns. For a comprehensive understanding of LED-to-camera communication and the underlying rolling-shutter mechanism of CMOS cameras, we refer readers to previous works [4], [8], [9]. In this work, we employ a distributed VLC broadcasting network with one-way communication from LED transmitters to a rolling-shutter (RS-) camera receiver. And each LED keeps transmitting a unique identity (ID) according to a pre-defined VLC protocol.

B. Visible Light Positioning

Normally, the visible light positioning (VLP) systems employ modulated LED luminaries mounted at known locations (e.g., the ceiling) as artificial landmarks, use cameras [3]–[7] or photodiodes [10]–[12] as light sensors, measure the angle-of-arrival (AOA) or received signal strength (RSS) properties of the incoming light from those visible landmarks, associate each light measurement with a certain landmark by recognizing its unique ID through VLC, and finally, determine the target location using the geometry constraints from these associated measurements. In order to fix the receiver's position in 3D, the traditional geometry-only methods, including both the triangulation with bearing measurements [3]–[7] and the trilateration with ranging measurements [10], normally require observing at least three LEDs at a time. To tackle this problem, we introduce an IMU to the camera-based VLP system, which can provide relative motion measurements for the moving camera. Further, an EKF is applied to correct the IMU states with the visual measurements to LED landmarks.

C. IMU-aided VLP

There are several VLP methods with IMU assistance. A majority of them fell into the loosely coupled category. In some works [10], [13], the IMU was utilized to provide absolute orientation measurements or simply the roll and pitch measurements around the gravity. As one of the most famous pioneering works in VLP, Epsilon [10] revealed the LED shortage problem in real situations and proposed to solve it by accumulating a continuous period of light measurements from a given location at different camera attitudes measured by the IMU. It was shown that localization results with meter-level accuracy were achievable by observing only one LED. Yet tedious user involvement was required. To improve the robustness of VLP under LED outage, [14] proposed to fuse the geometry-only position estimates with the relative motion estimates from pedestrian dead-reckoning, for example when the IMU was carried by walking persons. Due to the loosely-coupled nature, it may still fail to handle insufficient light observations. [15] was a tightly-coupled fusion method through graph optimization for VLP, with an aim to cope with the LED shortage problem. The proposed method can work with two or more LEDs. Note that the authors employed the AprilTag [16] fiducial markers, instead of real LEDs, for the experiments. However, an AprilTag marker is not equivalent to a point-source LED, since each marker, with four distinctive corners, can provide four point feature measurements.

III. Methodology

We consider the context of absolute 3D pose estimation using an IMU sensor and an RS-camera in a known environment equipped with instrumented LEDs as artificial visual landmarks. The identity of each landmark, as well as its 3D position in the environment, is known a prior and registered in a database, e.g., from an offline lights mapping procedure. Moreover, we are able to obtain the unique ID for each observed landmark by means of VLC, and retrieve the associated 3D location by querying the lights database with the decoded identity. Our goal is thus to estimate the 3D pose of the IMU body frame, $\{I\}$, with respect to a fixed global frame, $\{G\}$, by fusing IMU's inertial measurements with the visual measurements of LED landmarks observed in the camera frame, $\{C\}$. To this end, an EKF-based global pose estimator is proposed.

Here, we choose the z axis of $\{G\}$ to be aligned with gravity and be pointing straight upwards. As such, the gravitational acceleration expressed in $\{G\}$ is ${}^{G}\mathbf{g} = [0\,,0\,,-g]^{\top}$. The spatial transformation, ${}^{C}_{I}\mathbf{T}$, between the IMU frame and the camera frame, is known, e.g., by prior extrinsic calibration, and remains constant. It can be further expressed by a unit quaternion, ${}^{C}_{I}\bar{\mathbf{q}}$, that describes the rotation from the IMU frame to the camera frame, and a translation vector, ${}^{C}\mathbf{p}_{I}$, that represents the IMU's position in the camera frame. We use both the quaternion by following the JPL convention [17] and rotation matrix for the rotation representation, e.g., ${}^C_I\mathbf{R} = \mathbf{R}\left({}^C_I\bar{\mathbf{q}}\right)$. Furthermore, we assume a calibrated pinhole camera model, i.e., with known camera intrinsic parameters.

A. VLC Frontend

To obtain the camera observations to LED landmarks, we should detect the potential landmarks in the camera image, and for each landmark, we need to find its centroid imaging location, recognize its identity, and retrieve its 3D position from the registered lights database. In the following, we first define the VLC protocol in use, and then briefly introduce our image processing pipeline and the VLC decoding scheme.

1) VLC protocol design: We assume an RS-camera that exposes a row of pixels at a time, with a row read-out time, τ_r. The sampling frequency is defined as per $f_s = 1/\tau_r$, where f_s is a few tens of kilohertz for normal RS-cameras [9]. In addition, we consider a circular-shaped LED transmitter that can only be switched fully on and off, under the control of square wave signals. We employ an on-off-keying (OOK) modulation scheme with Manchester coding to encode data messages, for the sake of its simplicity and the DC-balancing nature [12]. The OOK modulation frequency is $f_m = 1/\tau_m$ where τ_m is the sampling interval. In other words, τ_m equals to the minimum duration for a symbol bit. When received by the camera, the modulated pulses are captured as bright or dark strips with varying widths proportional to the pulse durations. The width of the narrowest strip is computed as $\mathrm{w}_0 = \tau_m/\tau_r$, measured in pixels. A data packet of L bits long leads to a strip pattern extending $\mathrm{w}_0 L$ pixels in height. That is, we need at least $\mathrm{w}_0 L$ rows of pixels from the strip pattern to fully recover the information carried by the data packet.

The designed VLC data packet starts with a 4-bit preamble symbol $\mathrm{PS} = \mathrm{b0001}$ indicating the start of a new packet, precedes with a 16-bit data symbol DATA with Manchester coding, and ends with another 4-bit symbol $\mathrm{ES} = \mathrm{b0111}$. This structure results in a complete packet length of 24 bits and balanced DC components in the modulated light intensities. The preamble comprises a 3-bit low-logic and the ending symbol comprises a 3-bit high-logic, both of which never occur in the Manchester-coded symbols. The data symbol encodes one byte of ID messages which can label up to 256 LEDs. The resulting VLC channel capacity is adequate for our existing system implementation and can be extended trivially yet at the cost of a larger packet length. Note that the proposed packet format does not involve any special structure intended for error checking or data recovery. This is because we want to reduce the required pattern size for VLC decoding by minimizing the packet length. By doing so, we are allowed to decode the message from a given LED at a longer distance.

2) Image processing pipeline: The strip patterns induced by modulated LEDs are parallel to the rows in the image and interleaving in the vertical direction, as shown in Fig. 1. We are interested in the image regions that contain spatially-varying strips, as they carry the encoded VLC information. The goal is to extract the regions of interest (ROIs) from the image and obtain the undistorted centroid pixel coordinates as camera measurements. To do so, we first get the gray-scale image and binarize it with a fixed threshold value. We dilate the binary image to fill the banding gaps. Then we detect the ROIs by finding the connected blobs. For each of them, we compute the centroid pixel location and the blob size. For the subsequent VLC decoding, we only keep ROIs that are large enough to contain a complete data packet as candidate regions. Further, we get the undistorted centroid locations using the calibrated camera intrinsics. We crop the gray-scale image with the ROI masks and send the image crops to the VLC decoder.

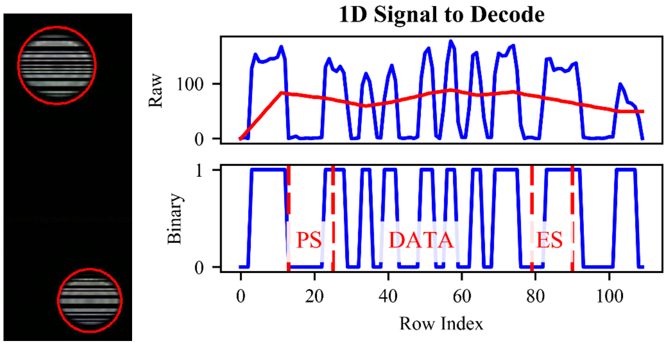


Fig. 1: Example results for ROI extraction and VLC decoding. The left shows a cropped image with two ROIs that contain strip patterns induced by modulated LEDs. The minimum strip width is determined by experiments, e.g., $\mathrm{w}_0 \approx 3$ pixels. The right shows the 1D intensity signals for decoding before (upper) and after (lower) thresholding. In the lower sub-figure, we illustrate a complete data packet containing symbols PS, DATA, and ES.

3) VLC decoding scheme: We hereafter consider a set of separate ROI candidates for VLC decoding. Obviously, the VLC information is carried by the vertically-varying strip widths within ROIs. For each candidate, we pick the gray-scale pixels in the image column crossing the region center and arrange them in a 1D array indexed by rows. Knowing the camera's sampling frequency f_s, we treat these pixel values as time-varying 1D signals. We convert the intensity signals to a binary waveform by adaptive thresholding [18], aiming to counter the artifacts caused by the nonuniform illumination on the LED radiation surface. Following our protocol, we perform OOK demodulation and Manchester decoding in the time domain to recover the ID. We show an example of the 1D signals prior to VLC decoding on the right side of Fig. 1.

B. EKF State Formulation

We define the IMU state as follows [19],

$$\mathbf{x}_I = \begin{bmatrix} {}^I_G\bar{\mathbf{q}}^\top & {}^G\mathbf{p}_I^\top & {}^G\mathbf{v}_I^\top & \mathbf{b}_g^\top & \mathbf{b}_a^\top \end{bmatrix}^\top \in \mathbb{R}^{16} \tag{1}$$

where the unit quaternion ${}^I_G\bar{\mathbf{q}}$ represents the rotation from the global frame to the IMU frame. The vectors ${}^G\mathbf{p}_I$ and ${}^G\mathbf{v}_I$ are the position and velocity of the IMU origin in the global frame. The gyroscope bias $\mathbf{b}_g$ and the accelerometer bias $\mathbf{b}_a$ are expressed in the local IMU frame. They are modeled as random walk processes driven by white Gaussian noise, $\mathbf{n}_{wg} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}^2_{wg})$ and $\mathbf{n}_{wa} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}^2_{wa})$, respectively.

Further, we define the IMU error state vector as,

$$\tilde{\mathbf{x}}_I = \begin{bmatrix} {}^I\tilde{\boldsymbol{\theta}}^\top & {}^G\tilde{\mathbf{p}}_I^\top & {}^G\tilde{\mathbf{v}}_I^\top & \tilde{\mathbf{b}}_g^\top & \tilde{\mathbf{b}}_a^\top \end{bmatrix}^\top \in \mathbb{R}^{15} \tag{2}$$

where we use the standard additive error definition for the position, velocity, and bias terms, e.g., ${}^G\mathbf{p}_I = {}^G\hat{\mathbf{p}}_I + {}^G\tilde{\mathbf{p}}_I$. For the rotation quaternion errors, we employ the classic local perturbation of quaternions, as defined in the following,

$${}^I_G\bar{\mathbf{q}} = \delta\bar{\mathbf{q}} \otimes {}^I_G\hat{\bar{\mathbf{q}}} \Leftrightarrow \delta\bar{\mathbf{q}} = {}^I_G\bar{\mathbf{q}} \otimes {}^I_G\hat{\bar{\mathbf{q}}}^{-1} \tag{3}$$

By applying the small angle approximation [17], we have

$$\delta\bar{\mathbf{q}} \simeq \begin{bmatrix} \frac{1}{2}{}^I\tilde{\boldsymbol{\theta}} \\ 1 \end{bmatrix} \quad \text{and} \quad \mathbf{R}\left(\delta\bar{\mathbf{q}}\right) \simeq \mathbf{I}_3 - \lfloor {}^I\tilde{\boldsymbol{\theta}}_\times \rfloor \tag{4}$$

The error quaternion $\delta\bar{\mathbf{q}}$ as well as its rotation matrix $\mathbf{R}\left(\delta\bar{\mathbf{q}}\right)$ describes the infinitesimal rotation that can align the estimated IMU frame with the true one. Following the standard practice, we use the minimal 3×1 rotation error representation, ${}^I\tilde{\boldsymbol{\theta}}$, expressed in the local IMU frame for the error state formulation.

C. IMU Propagation

The continuous-time IMU measurements of angular velocity, $\boldsymbol{\omega}_m$, and acceleration, $\mathbf{a}_m$, are given by [20]:

$$\boldsymbol{\omega}_m = \boldsymbol{\omega} + \mathbf{b}_g + \mathbf{n}_g \tag{5}$$

$$\mathbf{a}_m = {}^I_G\mathbf{R}({}^G\mathbf{a} - {}^G\mathbf{g}) + \mathbf{b}_a + \mathbf{n}_a \tag{6}$$

where $\boldsymbol{\omega}$ is the angular velocity of IMU in the local frame and ${}^G\mathbf{a}$ is the acceleration of IMU expressed in the global frame. The measurements are corrupted by additive white Gaussian noise, $\mathbf{n}_g \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}_g^2)$, and $\mathbf{n}_a \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}_a^2)$, respectively. For brevity, we denote ${}^I_G\mathbf{R} = \mathbf{R}\left({}^I_G\bar{\mathbf{q}}\right)$.

The continuous-time dynamics of the evolving IMU state can be described by the following equations:

$$\begin{aligned} {}^I_G\dot{\bar{\mathbf{q}}}(t) &= \frac{1}{2}\boldsymbol{\Omega}\left(\boldsymbol{\omega}(t)\right){}^I_G\bar{\mathbf{q}}(t) \\ {}^G\dot{\mathbf{p}}_I(t) &= {}^G\mathbf{v}_I(t), \quad {}^G\dot{\mathbf{v}}_I(t) = {}^G\mathbf{a}(t) \\ \dot{\mathbf{b}}_g(t) &= \mathbf{n}_{wg}(t), \qquad \dot{\mathbf{b}}_a(t) = \mathbf{n}_{wa}(t) \end{aligned} \tag{7}$$

with $\boldsymbol{\Omega}(\cdot)$ as the matrix operator,

$$\boldsymbol{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} -\lfloor \boldsymbol{\omega}_\times \rfloor & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^\top & 0 \end{bmatrix}, \lfloor \boldsymbol{\omega}_\times \rfloor = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

where $\lfloor \cdot_\times \rfloor$ is the skew-symmetric matrix for cross product.

To propagate the state estimate, $\hat{\mathbf{x}}_I$, we obtain the nominal IMU state equations by taking the expectation of Eq. 7:

$$\begin{aligned} {}^I_G\dot{\hat{\bar{\mathbf{q}}}} &= \frac{1}{2}\boldsymbol{\Omega}\left(\hat{\boldsymbol{\omega}}\right){}^I_G\hat{\bar{\mathbf{q}}}, \quad \dot{\hat{\mathbf{b}}}_g = \mathbf{0}, \quad \dot{\hat{\mathbf{b}}}_a = \mathbf{0} \\ {}^G\dot{\hat{\mathbf{p}}}_I &= {}^G\hat{\mathbf{v}}_I, \quad {}^G\dot{\hat{\mathbf{v}}}_I = {}^G\hat{\mathbf{a}} = {}^I_G\hat{\mathbf{R}}^\top\hat{\mathbf{a}} + {}^G\mathbf{g} \end{aligned} \tag{8}$$

with $\hat{\boldsymbol{\omega}} = \boldsymbol{\omega}_m - \hat{\mathbf{b}}_g$, $\hat{\mathbf{a}} = \mathbf{a}_m - \hat{\mathbf{b}}_a$, and ${}^I_G\hat{\mathbf{R}} = \mathbf{R}\left({}^I_G\hat{\bar{\mathbf{q}}}\right)$. Then we employ the fourth order Runge-Kutta numerical integration method for the discrete-time implementation.

The continuous-time dynamics of the IMU error state $\tilde{\mathbf{x}}_I$ linearized at its nominal state $\hat{\mathbf{x}}_I$ is written as

$$\dot{\tilde{\mathbf{x}}}_I = \mathbf{F}\tilde{\mathbf{x}}_I + \mathbf{G}\mathbf{n}_I \tag{9}$$

with $\mathbf{n}_I = \begin{bmatrix} \mathbf{n}_g^\top & \mathbf{n}_{wg}^\top & \mathbf{n}_a^\top & \mathbf{n}_{wa}^\top \end{bmatrix}^\top$, and

$$\mathbf{F} = \begin{bmatrix} -\lfloor \hat{\boldsymbol{\omega}}_\times \rfloor & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ -{}^I_G\hat{\mathbf{R}}^\top\lfloor \hat{\mathbf{a}}_\times \rfloor & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -{}^I_G\hat{\mathbf{R}}^\top \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix} -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -{}^I_G\hat{\mathbf{R}}^\top & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 \end{bmatrix}.$$

More specifically, for the IMU noise, $\mathbf{n}_I$, we have $\mathbf{Q}_c = \mathbb{E}\left[\mathbf{n}_I\mathbf{n}_I^\top\right] = \mathbf{diag}\{\boldsymbol{\sigma}_g^2, \boldsymbol{\sigma}_{wg}^2, \boldsymbol{\sigma}_a^2, \boldsymbol{\sigma}_{wa}^2\}$. $\mathbf{Q}_c$ is the continuous-time noise covariance matrix which, for example, can be obtained by sensor calibration or from the IMU specification.

For the state covariance propagation of the EKF, we need a discrete-time form of the IMU error state dynamics in Eq. 9. The discrete-time error state transition matrix, $\boldsymbol{\Phi}_k$, and the discrete-time noise covariance matrix, $\mathbf{Q}_d$, are computed as

$$\boldsymbol{\Phi}_k = \boldsymbol{\Phi}\left(t_{k+1}, t_k\right) = \exp\left(\int_{t_k}^{t_{k+1}} \mathbf{F}(\tau)d\tau\right) \tag{10}$$

$$\mathbf{Q}_d = \int_{t_k}^{t_{k+1}} \boldsymbol{\Phi}\left(t_{k+1}, \tau\right)\mathbf{G}(\tau)\mathbf{Q}_c\mathbf{G}^\top(\tau)\boldsymbol{\Phi}^\top\left(t_{k+1}, \tau\right)d\tau \tag{11}$$

We assume $\mathbf{F}(t)$ is constant over a small time interval, $\Delta t_k = t_{k+1} - t_k$, such that $\boldsymbol{\Phi}_k \simeq \exp\left(\mathbf{F}\left(t_k\right)\Delta t_k\right)$. By further applying the first-order approximation, we have $\boldsymbol{\Phi}_k \simeq \mathbf{I} + \mathbf{F}(t_k)\Delta t_k$. Accordingly, we can approximate the discrete-time noise covariance matrix by $\mathbf{Q}_d \simeq \mathbf{G}(t_k)\mathbf{Q}_c\mathbf{G}^\top(t_k)\Delta t_k$. Then, the state covariance matrix $\mathbf{P}$ can be propagated as:

$$\mathbf{P}_{k+1|k} = \boldsymbol{\Phi}_k\mathbf{P}_{k|k}\boldsymbol{\Phi}_k^\top + \mathbf{Q}_d \tag{12}$$

D. Camera Measurement Update

The EKF employs the camera measurements of known LED landmarks to correct its state estimate. We assume a fully calibrated pinhole camera with a perspective projection model. The VLC frontend is responsible for image processing and VLC message interpretation. Upon successful detection and decoding for a given landmark feature f_i, we can obtain its normalized image measurement $\mathbf{z}_i = [u_i \, v_i]^\top$, and its unique identity code, ID$_i$, together with the absolute 3D position in the global frame, ${}^G\mathbf{p}_{f_i}$, which is known but with some uncertainties. It happens that some LEDs within the FoV may fail to decode by the VLC frontend, e.g., when they are too far away from the camera. Here, we are interested in the observations to the decodable LEDs. For any given image taken at time t, we hereafter consider a set of features, $\{f_i\}$, from the observed LED landmarks with successful VLC decoding results.

The position of the ith feature in the camera frame, ${}^C\mathbf{p}_{f_i}$, can be computed as

$${}^C\mathbf{p}_{f_i} = {}^C_I\mathbf{R}\,{}^I_G\mathbf{R}\left({}^G\mathbf{p}_{f_i} - {}^G\mathbf{p}_I\right) + {}^C\mathbf{p}_I \tag{13}$$

where ${}^C_I\mathbf{R} = \mathbf{R}\left({}^C_I\bar{\mathbf{q}}\right)$ and ${}^C\mathbf{p}_I$ represent the known spatial transformation with some uncertainties between the two sensor frames. The camera observation to this feature is described by

$$\mathbf{z}_i = \mathbf{h}\left({}^C\mathbf{p}_{f_i}\right) + \mathbf{n}_{im} \tag{14}$$

where $\mathbf{h}(\cdot)$ is the perspective projection function, for example, $\mathbf{h}\left([x,y,z]^\top\right) = [x/z, y/z]^\top$, and $\mathbf{n}_{im} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}_{im}^2)$ is the image measurement noise expressed in normalized pixels.

In order to model the potential errors in the "known" spatial transformation, $\{{}^C_I\bar{\mathbf{q}}, {}^C\mathbf{p}_I\}$, for example due to imperfect extrinsic calibration, we define the transformation error as

$${}^C\mathbf{p}_I = {}^C\hat{\mathbf{p}}_I + {}^C\tilde{\mathbf{p}}_I \quad \text{and} \quad {}^C_I\bar{\mathbf{q}} \simeq {}^C_I\hat{\bar{\mathbf{q}}} \otimes \begin{bmatrix} \frac{1}{2}{}^I\tilde{\boldsymbol{\phi}} \\ 1 \end{bmatrix}$$

where, similar to the orientation error in the filter state, the rotation error ${}^I\tilde{\boldsymbol{\phi}}$ is defined in the IMU frame. Moreover, we consider the case when the locations of LED landmarks are subject to some mapping errors. As such, we define the feature position error vector, ${}^G\tilde{\mathbf{p}}_{f_i} = {}^G\mathbf{p}_{f_i} - {}^G\hat{\mathbf{p}}_{f_i}$. To account for the uncertainties induced by the aforementioned error sources, we model them as zero-mean white Gaussian noises, i.e., ${}^G\tilde{\mathbf{p}}_f \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}_f^2)$, ${}^C\tilde{\mathbf{p}}_I \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}_{pc}^2)$, and ${}^I\tilde{\boldsymbol{\phi}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}_{qc}^2)$.

Given the latest state estimate for $\hat{\mathbf{x}}_I$ from the IMU propagation, as well as the expected feature measurement $\hat{\mathbf{z}}_i = \mathbf{h}\left({}^C\hat{\mathbf{p}}_{f_i}\right)$, we can now compute the measurement residue $\mathbf{r}_i = \mathbf{z}_i - \hat{\mathbf{z}}_i$ by first-order approximation:

$$\begin{aligned} \mathbf{r}_i &\simeq \mathbf{H}_{\mathbf{x},i}\tilde{\mathbf{x}}_I + \mathbf{H}_{\boldsymbol{\phi},i}{}^I\tilde{\boldsymbol{\phi}} + \mathbf{H}_{\mathbf{p}_c,i}{}^C\tilde{\mathbf{p}}_I + \mathbf{H}_{f_i}{}^G\tilde{\mathbf{p}}_{f_i} + \mathbf{n}_{im} \\ &= \mathbf{H}_{\mathbf{x},i}\tilde{\mathbf{x}}_I + \mathbf{n}_{o,i} \end{aligned} \tag{15}$$

where $\mathbf{H}_{\mathbf{x},i}$ is the measurement Jacobian w.r.t. the IMU state. $\mathbf{H}_{\boldsymbol{\phi},i}$ and $\mathbf{H}_{\mathbf{p}_c,i}$ are the Jacobians w.r.t. the IMU-camera rotation and translation, respectively. $\mathbf{H}_{f_i}$ is the Jacobian w.r.t. the ith feature position. They can be computed as

$$\begin{aligned} \mathbf{H}_{\mathbf{x},i} &= [\mathbf{H}_{\boldsymbol{\theta},i} \quad \mathbf{H}_{\mathbf{p},i} \quad \mathbf{0}_{2\times 9}] \\ \mathbf{H}_{\boldsymbol{\theta},i} &= \mathbf{J}_i\, {}^C_I\hat{\mathbf{R}} \lfloor {}^I_G\hat{\mathbf{R}} \left({}^G\hat{\mathbf{p}}_{f_i} - {}^G\hat{\mathbf{p}}_I\right)_\times \rfloor \\ \mathbf{H}_{\mathbf{p},i} &= -\mathbf{J}_i\, {}^C_I\hat{\mathbf{R}}\, {}^I_G\hat{\mathbf{R}} \\ \mathbf{H}_{f_i} &= -\mathbf{H}_{\mathbf{p},i} \quad \mathbf{H}_{\boldsymbol{\phi},i} = \mathbf{H}_{\boldsymbol{\theta},i} \quad \mathbf{H}_{\mathbf{p}_c,i} = \mathbf{J}_i \end{aligned}$$

where $\mathbf{J}_i = \partial\mathbf{h}(\mathbf{f})/\partial\mathbf{f}$ is the Jacobian of $\mathbf{h}(\cdot)$ evaluated at the expected feature position, ${}^C\hat{\mathbf{p}}_{f_i} = [\hat{x}, \hat{y}, \hat{z}]^\top$, in the camera frame, i.e., $\mathbf{J}_i = \frac{1}{\hat{z}}\begin{bmatrix} 1 & 0 & -\hat{x}/\hat{z} \\ 0 & 1 & -\hat{y}/\hat{z} \end{bmatrix}$.

To deal with the uncertainties in feature locations as well as in the IMU-camera extrinsic parameters, we combine these modeled error terms into an observation noise vector, $\mathbf{n}_{o,i}$, and accordingly inflate the measurement noise covariance as

$$\begin{aligned} \mathbf{R}_i =& \mathbb{E}\left[\mathbf{n}_{o,i}\mathbf{n}_{o,i}^\top\right] \\ =& \mathbf{H}_{\boldsymbol{\phi},i}\mathbb{E}\left[{}^I\tilde{\boldsymbol{\phi}}\,{}^I\tilde{\boldsymbol{\phi}}^\top\right]\mathbf{H}_{\boldsymbol{\phi},i}^\top + \mathbf{H}_{\mathbf{p}_c,i}\mathbb{E}\left[{}^C\tilde{\mathbf{p}}_I{}^C\,\tilde{\mathbf{p}}_I^\top\right]\mathbf{H}_{\mathbf{p}_c,i}^\top \\ &+ \mathbf{H}_{f_i}\mathbb{E}\left[{}^G\tilde{\mathbf{p}}_{f_i}\,{}^G\tilde{\mathbf{p}}_{f_i}^\top\right]\mathbf{H}_{f_i}^\top + \mathbb{E}\left[\mathbf{n}_{im}\,\mathbf{n}_{im}^\top\right] \\ =& \mathbf{H}_{\boldsymbol{\phi},i}\mathbf{H}_{\boldsymbol{\phi},i}^\top\boldsymbol{\sigma}_{qc}^2 + \mathbf{H}_{\mathbf{p}_c,i}\mathbf{H}_{\mathbf{p}_c,i}^\top\boldsymbol{\sigma}_{pc}^2 + \mathbf{H}_{f_i}\mathbf{H}_{f_i}^\top\boldsymbol{\sigma}_f^2 + \boldsymbol{\sigma}_{im}^2 \end{aligned} \tag{16}$$

Following the general EKF equations [17], we can now update the state and covariance estimates according to

$$\mathbf{S}_i = \mathbf{H}_{\mathbf{x},i}\,\mathbf{P}_{k+1|k}\,\mathbf{H}_{\mathbf{x},i}^\top + \mathbf{R}_i \tag{17}$$

$$\mathbf{K}_i = \mathbf{P}_{k+1|k}\,\mathbf{H}_{\mathbf{x},i}^\top\,\mathbf{S}_i^{-1} \tag{18}$$

$$\hat{\mathbf{x}}_{k+1|k+1} \leftarrow \hat{\mathbf{x}}_{k+1|k} \oplus \mathbf{K}_i\,\mathbf{r}_i \tag{19}$$

$$\mathbf{P}_{k+1|k+1} \leftarrow \mathbf{P}_{k+1|k} - \mathbf{K}_i\,\mathbf{S}_i\,\mathbf{K}_i^\top \tag{20}$$

where $\hat{\mathbf{x}}_{k+1|k}$ and $\mathbf{P}_{k+1|k}$ denote the latest filter state and covariance estimates by IMU propagation using all the inertial measurements between the current image timestamped at t_{k+1} and the last processed image timestamped at t_k. $\mathbf{S}_i$ is the covariance matrix of the measurement residual, and $\mathbf{K}_i$ is the computed Kalman gain. To carry out the state correction, we employ a compound addition operator, $\oplus$, where the quaternion multiplication is used for the IMU orientation, as well as the standard addition for other quantities.

When multiple LED landmarks are successfully decoded in the current image, the EKF can sequentially update its state estimate $\hat{\mathbf{x}}_I$ and state covariance estimate $\mathbf{P}$ using these observations one by one. Note that, for each feature update, we compute all the above-mentioned Jacobians using the same IMU state estimate $\hat{\mathbf{x}}_{k+1|k}$ that is available from the latest IMU propagation stage, for example according to Eq. 8.

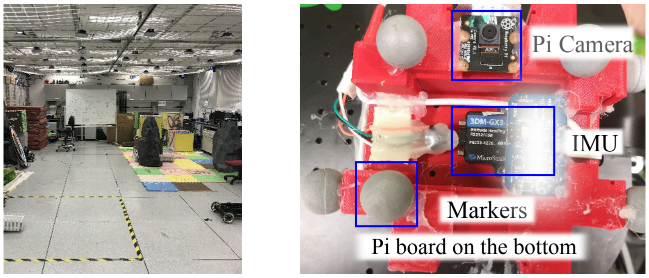
In the context of VLC, we are probably expected to recognize each LED landmark exactly without any false detection, e.g., by exploiting some sophisticated communication algorithms with proper information redundancy. In such situations, we may safely use all the available measurements for the EKF update. In this work, however, due to hardware limitations, we resort to a simple communication protocol with reduced checking mechanisms for the data integrity. As a result, VLC decoding errors may happen. Consider the following situation where the ID for a given LED is wrongly reported, and meanwhile, the wrong ID code happens to refer to another LED registered in the lights map. The resulting observation will be inconsistent with the state estimate and should be rejected as an outlier. To this end, we perform a gating test by computing the Mahalanobis distance of the measurement residue for each new LED observation. And we only use those measurements passing the test for the filter update.

IV. Evaluation

In this section, we evaluate the performance of our method in a real-world environment with dozens of customized LEDs and verify the robustness under LED shortage. To measure the positioning performance, we compute the absolute trajectory error (ATE) of the trajectory estimates by comparing with the ground truth [21]. Also, we employ the root-mean-square error (RMSE) of the estimated trajectory as an indication of the overall positioning performance in terms of accuracy.

A. Experimental Settings

We set up a room-scaled (around 5m × 4m × 2.3m) testing environment with 23 LED prototypes mounted on the ceiling. These LEDs are almost evenly distributed with spacing around



(a) Test field

(b) Sensor rig

Fig. 2: The test field (a) with 23 home-made LEDs mounted on the ceiling, and our self-assembled sensor rig (b) for data collection.

0.8-$1.5\mathrm{m}$. We employ a motion capture system[2] (a.k.a., Mocap) to provide ground truth poses for our experiments. We define the world frame $\{W\}$ for Mocap to fit the room layout, i.e., with the x-axis aligned to the length direction and the z-axis pointing upwards. The reference origin is affixed onto the ground. For the sake of convenience, we set the global frame $\{G\}$ to coincide with the world frame $\{W\}$ defined by Mocap. We measure the 3D positions of these LEDs in $\{G\}$ during a manual site survey using a commodity laser range finder with centimeter-level accuracy.

The LED's radiation surface has a circular shape of size $15.5\mathrm{cm}$ in diameter. Our customized LED driver is composed of a cheap microcontroller running our VLC protocol and a MOSFET transistor modulating the driving current. The OOK modulation frequency is set to $f_m = 16\mathrm{kHz}$. The rating power for each LED is around $3\mathrm{W}$. We have assembled a customized sensor rig for data collection, which includes a Raspberry Pi[3](a.k.a., Pi) single board computer (model 3B) with its onboard RS-camera (Sony IMX219) and a MicroStrain[4] IMU (3DM-GX3-25). The Raspberry Pi runs a Ubuntu Mate 16.04 OS, together with a robot operating system[5] (ROS kinetic). We control the camera settings by minimizing the exposure time, so as to see clear strip patterns from the modulated LEDs. The image stream is captured at $10\mathrm{Hz}$ with a 1640×1232 resolution and recorded as ROS bags. The camera has a vertical FoV of $48.8\,\mathrm{deg}$ and a focal length of 1284 pixels under our resolution setting. The IMU measurements are sampled at $200\mathrm{Hz}$ and the Mocap ground truth poses are recorded at $120\mathrm{Hz}$. It is worth mentioning that the Mocap poses could be lost from time to time. This is probably due to the temporal LOS blockage of the reflective markers caused by the moving human body. When computing the trajectory errors, we only use the poses with valid ground truth.

We obtain the camera intrinsic parameters, as well as the camera-to-IMU extrinsic parameters with the Kalibr calibration toolbox[6]. We run our algorithm on a desktop computer (CPU Intel i7-7700K) using the recorded bags from Pi.

B. VLC Decoding Performance

We aim to investigate the decoding performance of the proposed VLC frontend under the existing hardware setup. We define the VLC decoding rate, for a given LED, as the ratio of the accumulated number of image frames with successful decoding results to the total number of captured frames over a certain period of time.

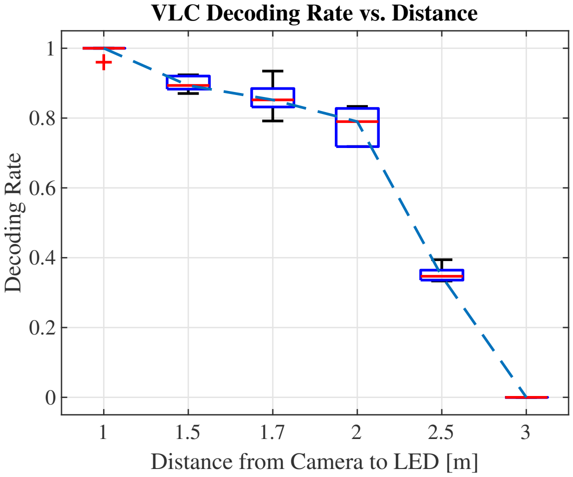


Fig. 3: The decoding performance of our VLC frontend.

We attach a LED to the surface of a vertical wall so that it can point forwards horizontally. We orient the camera to squarely face against the out-coming direction of the LED's normal vector. We vary the distance in between ranging from $1\mathrm{m}$ to $3\mathrm{m}$. Then we collect a $60\mathrm{s}$ long image stream at $10\mathrm{Hz}$ for each distance and compute the decoding rate for a subsequence of images lasting every $5\mathrm{s}$. The results are shown by the boxplots in Fig. 3. Obviously, the decoding rate will decrease with distance. We observe that it drops quickly at distances larger than $2\mathrm{m}$. This is probably because the captured light pattern can only contain one complete data packet at some large distances. We can achieve a median rate of around 0.8 at $2\mathrm{m}$. It drops to around 0.35 at $2.5\mathrm{m}$, and further drops to 0 at $3\mathrm{m}$ in which case the light pattern is too small to decode. Therefore, the maximum decoding distance achieved by the existing hardware setup is larger than yet close to $2.5\mathrm{m}$.

C. Real-time Pose Estimates

To test the real-time localization performance, we collect data with the proposed sensor rig in the testing environment. We orient the camera upwards to face the ceiling lights. To initialize the EKF, we compute the initial camera position and orientation by solving a normal PnP problem with at least four LED observations using OpenCV[7]. Yet from our experience, we can safely decode four LEDs only when the camera is on the ground, e.g., due to the small radiation surface of our LEDs. To help with the initialization, we first put the rig on the ground and keep it still for a few seconds. In this way, we can simply set the initial pose using PnP and set the initial

[2] https://www.vicon.com/
[3] https://www.raspberrypi.org/
[4] https://www.microstrain.com/
[5] https://www.ros.org/
[6] https://github.com/ethz-asl/kalibr
[7] https://opencv.org/

velocity to zero. Then we hold the sensor by hand and walk continuously along an eight-figured trajectory five times.

The final trajectory travels approximately 78m long in 93s. Fig. 4 shows the pose estimation results including the estimated 3D trajectory, the position estimates over time, and the orientation estimates expressed by Euler angles. We compare our results with the ground truth from Mocap. The EKF estimates are very close to the ground truth by visual inspection. We have achieved an RMSE around 4.9cm by observing two LEDs on average in each camera frame.

The evolving number of decodable LEDs in each frame over time is shown in Fig. 5. The magenta line marks the position of three LEDs. Note that the vision-only VLP methods normally require at least three observations in a single image. Yet, we find it is rather difficult to decode more than three LEDs at a time throughout this experiment, in spite of the dense LED deployment. In the meantime, the proposed tightly-coupled visual-inertial fusion method can run smoothly and achieve accurate estimation results.

D. Robustness Test Under LED Shortage

To further explore the robustness of our method under the shortage of available LEDs, we perform two sets of tests under the dense and sparse LED deployment, respectively. We have previously registered a total number of 23 LEDs in the original lights map (i.e., dense map) with a coverage of around 20m^2. To approach the sparse deployment case, we intentionally remove half of those registered LEDs from the dense map in a uniform manner. As such, we build a sparse lights map containing 12 LEDs but covering the same testing area. In this scenario, we may simply discard a camera measurement if the decoded ID is no longer in the original map.

We collect datasets over four walking trials in the testing area with all lights on. For each dataset, we run our positioning algorithm two times, i.e., one time with the dense lights map and the other with the sparse map. Using the dense map, we can initialize the filter by means of PnP with 4+ observations. However, we can rarely observe four registered LEDs in the sparse deployment case. Thus we are unable to initialize the filter by PnP. To circumvent this situation, we resort to the Mocap ground truth for the pose initialization. In this work, we are mainly interested in the localization performance over run-time upon the successful filter initialization.

TABLE I: Statistics for robustness tests over four trials.

Trials	1	2	3	4
Shape of traj.	circle	eight	square	square
Duration [s]	40.0	93.3	80.0	37.3
Length [m]	40.5	78.7	41.5	34.6
Max speed [m/s]	1.38	1.43	1.08	1.92
RMSE [cm]	4.7/**6.2**	4.9/**5.9**	4.3/**4.8**	3.9/**4.0**
Mean # LEDs	2.2/**1.0**	2.0/**0.9**	2.3/**1.1**	2.1/**1.1**
Pct. 1+ LEDs [%]	96.7/-	95.2/-	97.6/-	98.9/-
Pct. 3+ LEDs [%]	33.3/**0.2**	25.6/**1.2**	34.9/**0.7**	25.3/**3.2**

"-" indicates a duplicate of the former value before "/".

We show some key features of the four datasets in Table I, such as the trajectory shape, the duration, the total length, and

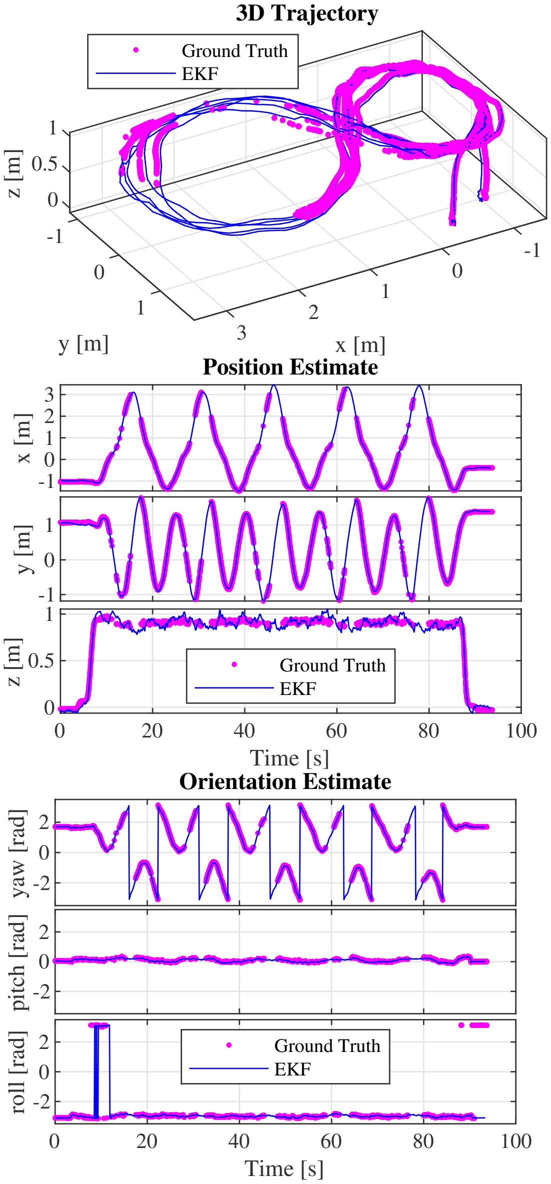


Fig. 4: The trajectory estimates of our EKF-based method compared with the ground truth from the motion capture system. The trajectory travels around 78m long over 93s following an eight figure. A few ground truth poses are missing at certain locations due to the human blockage. We thus only show the available ones.

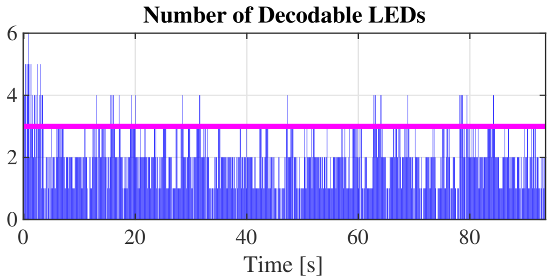


Fig. 5: The number of decodable LEDs in each frame over time.

the maximum walking speed. The statistics on the positioning accuracies and the number of observable LED landmarks are also summarized in this table. To be specific, we show the results for both the dense and sparse deployment cases in the bottom four rows, and the quantities for the latter case are boldfaced for comparisons. The RMSE errors are close to 5cm over the four trials, with both the dense and sparse deployment. When comparing the RMSE results in the sparse deployment case with that in the dense deployment case, we only observe marginal RMSE increases. Meanwhile, the average number of LEDs observed in each frame is decreased from around 2 to 1, indicating the substantial loss of usable LED measurements. Nevertheless, our method performs reasonably well over the four trials and can achieve a centimeter-level accuracy by observing one light on average in each camera frame. Therefore, the robustness of our method under LED shortage is verified.

Further, we compute the percentage of captured images that can decode 1+ registered LEDs, as well as the percentage of images decoding 3+ LEDs. We can successfully decode 1+ LEDs at a very high probability of over 95% in all the testing cases. This is a good proof of the usability of our method in practice. In contrast, we can rarely see 3+ LEDs under the sparse deployment, revealing that the vision-only method cannot be applied in such situations.

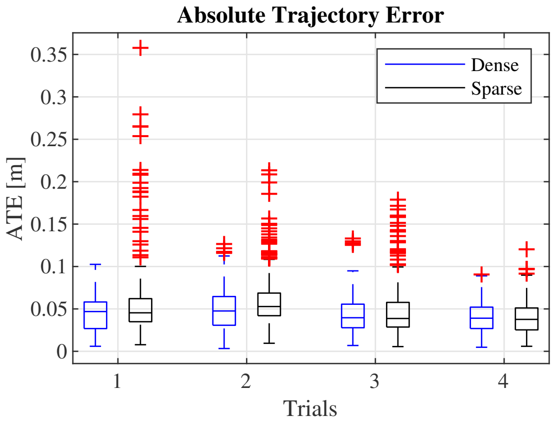


Fig. 6: The ATE of trajectory estimates over four trials in both the dense and sparse LED deployment cases.

Fig. 6 plots the ATE results of the trajectory estimates. We can achieve a median ATE around 5cm over all the four datasets in all the testing cases. The ATE distributions are more scattered under the sparse deployment, e.g., with more outliers in the boxplots and with larger maximum errors. However, the maximum errors are well constrained to a few decimeters, i.e., less than 0.4m in the worst case.

V. Conclusion

We presented an EKF-based fusion method by tight coupling for robust visible light positioning under LED shortage with an IMU and a rolling-shutter camera. We relaxed the assumption on the minimum number of concurrently observable LEDs from three to one for the camera-based visible light positioning systems. The method was evaluated by real-world experiments using a prototyping VLC network. The results showed that our method can robustly estimate the global 3D pose of the sensor pair with centimeter-level accuracy, by observing one LED on average in each camera frame. However, the existing system required good initialization, e.g., by solving a normal PnP problem with 4+ LED observations, or by a motion capture system. For our future study, we will explore more flexible yet accurate initialization methods.

References

[1] Y. Zhuang, L. Hua, L. Qi, J. Yang, P. Cao, Y. Cao, Y. Wu, J. Thompson, and H. Haas, "A survey of positioning systems using visible LED lights," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 1963–1988, 2018.

[2] J. Armstrong, Y. Sekercioglu, and A. Neild, "Visible light positioning: a roadmap for international standardization," *IEEE Commun. Mag.*, vol. 51, no. 12, pp. 68–73, 2013.

[3] Y. Nakazawa, H. Makino, K. Nishimori, D. Wakatsuki, and H. Komagata, "Indoor positioning using a high-speed, fish-eye lens-equipped camera in visible light communication," in *Proc. IPIN*. IEEE, 2013, pp. 1–8.

[4] Y.-S. Kuo, P. Pannuto, K.-J. Hsiao, and P. Dutta, "Luxapose: Indoor positioning with mobile phones and visible light," in *Proc. MobiCom'14*. ACM, 2014, pp. 447–458.

[5] A. Jovicic, "Qualcomm Lumicast: A high accuracy indoor positioning system based on visible light communication," 2016.

[6] Y. Li, Z. Ghassemlooy, X. Tang, B. Lin, and Y. Zhang, "A VLC smartphone camera based indoor positioning system," *IEEE Photon. Technol. Lett.*, vol. 30, no. 13, pp. 1171–1174, 2018.

[7] M. Rátosi and G. Simon, "Real-Time Localization and Tracking Using Visible Light Communication," in *Proc. IPIN*. IEEE, 2018, pp. 1–8.

[8] H.-Y. Lee, H.-M. Lin, Y.-L. Wei, H.-I. Wu, H.-M. Tsai, and K. C.-J. Lin, "Rollinglight: Enabling line-of-sight light-to-camera communications," in *Proc. MobiSys'15*. ACM, 2015, pp. 167–180.

[9] Y. Yang, J. Hao, and J. Luo, "CeilingTalk: Lightweight indoor broadcast through LED-camera communication," *IEEE Trans. Mobile Comput.*, vol. 16, no. 12, pp. 3308–3319, 2017.

[10] L. Li, P. Hu, C. Peng, G. Shen, and F. Zhao, "Epsilon: A visible light based positioning system," in *Proc. NSDI'14*, 2014, pp. 331–343.

[11] K. Qiu, F. Zhang, and M. Liu, "Let the light guide us: VLC-based localization," *IEEE Robot. Autom. Mag.*, vol. 23, no. 4, pp. 174–183, 2016.

[12] Q. Liang, L. Wang, Y. Li, and M. Liu, "Plugo: a Scalable Visible Light Communication System towards Low-cost Indoor Localization," in *Proc. IROS*. IEEE, 2018, pp. 3709–3714.

[13] G. Simon, G. Zachár, and G. Vakulya, "Lookup: Robust and accurate indoor localization using visible light communication," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 9, pp. 2337–2348, 2017.

[14] Z. Li, L. Feng, and A. Yang, "Fusion based on visible light positioning and inertial navigation using extended Kalman filters," *Sensors*, vol. 17, no. 5, p. 1093, 2017.

[15] C. Qin and X. Zhan, "VLIP: Tightly Coupled Visible-Light/Inertial Positioning System to Cope With Intermittent Outage," *IEEE Photon. Technol. Lett.*, vol. 31, no. 2, pp. 129–132, 2018.

[16] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *Proc. ICRA*. IEEE, 2011, pp. 3400–3407.

[17] N. Trawny and S. I. Roumeliotis, "Indirect kalman filter for 3d attitude estimation," *University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep*, vol. 2, p. 2005, 2005.

[18] M. Liu, K. Qiu, F. Che, S. Li, B. Hussain, L. Wu, and C. P. Yue, "Towards indoor localization using Visible Light Communication for consumer electronic devices," in *Proc. IROS*. IEEE, 2014, pp. 143–148.

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

[20] M. Li and A. I. Mourikis, "Online temporal calibration for camera–IMU systems: Theory and algorithms," *Int. J. Rob. Res.*, vol. 33, no. 7, pp. 947–964, 2014.

[21] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IROS*. IEEE, 2012, pp. 573–580.