Technical Report: A Tightly Coupled VLC-Inertial Localization System by EKF

Qing Liang and Ming Liu

***Abstract*— Lightweight global localization is favorable by many resource-constrained platforms working in GPS-denied indoor environments, such as service robots and mobile devices. In recent years, visible light communication (VLC) has emerged as a promising technology that can support global positioning in buildings with widespread LED luminaries. In this paper, we propose a tightly coupled VLC-inertial localization system based on an extended-Kalman filter (EKF) using modulated LEDs as artificial visual landmarks. By tightly fusing the motion measurements from an inertial measurement unit (IMU) with the camera measurements of known LED markers, our EKF localizer provides lightweight real-time accurate global pose estimates, even in LED-shortage situations. The system is completed by a 2-points global pose initialization method that loosely couples the two sensors. We can hence bootstrap our system with two or more LED features in one camera view. The proposed system and method are verified by extensive field experiments using a considerable quantity of LED prototypes.**

I. Introduction

Localization is essential for many robotic tasks like planning and navigation, as well as for a wide range of location-based services in daily life, such as augmented reality and pedestrian way-finding on mobile devices. We are interested in drift-free global solutions of metric-scale in GPS-denied indoor environments. Visual simultaneous localization and mapping approaches [1] can correct long-term drifts accumulated in local pose estimates by applying loop-closure after place recognition. Yet, the consistency is not guaranteed before major loop closure happens. It is also challenging for appearance-based loop detection [2] to work reliably in some texture-less indoor scenarios. Meanwhile, the state-of-the-art Lidar odometry and mapping systems [3], [4] can continuously provide consistent low-drift pose estimates in a large-scale workspace, by managing a global 3D point-cloud map and correcting odometry drifts. However, the expensive multi-scan Lidar sensor and high onboard computational requirements hinder their wide usage on resource-constrained platforms, such as service robots and mobile devices. Bearing these issues in mind, we are motivated to find such a lightweight localization solution that is accurate, consistent, reliable and more easily affordable by inexpensive devices.

*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.

Qing Liang and Ming Liu are with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology. qing.liang@connect.ust.hk, eelium@ust.hk

In recent years, localization using visible light communication (VLC) [5] has emerged as a competitive lightweight solution that can be deployed at scale in modern buildings. Besides illumination, LED lights can also function as artificial visual landmarks. The modulated LED actively broadcast its unique identity by VLC, which can later be recognized by a normal rolling-shutter camera. The mounting locations of lights can be mapped once for all, as they are normally fixed and not easily vulnerable to environmental changes. As such, conventional methods based on VLC simply come down to solving a localization problem with known data associations and prior a map. One may immediately obtain the global camera pose by solving the well-studied perspective-n-point (PnP) problem if more than three 2D-3D point correspondences are simultaneously available from decoded LEDs. Yet we find that such a requirement is usually demanding, if not impossible, to meet in real situations.

Note that square-shaped fiducial markers (e.g., AprilTag [6]) provide four distinctive corner features in each, which are sufficient to determine a camera pose. By contrast, LED lights, if not specially designed, provide less usable point features in a single observation due to the lack of distinguishable appearance, e.g., one feature for each circular-shaped LED. The number of decodable LEDs in a camera view is further limited by a few practical factors, such as the LEDs' deployment density, the ceiling height, the camera's field-of-view (FoV) and the maximum range for VLC decoding. As a result, the performance of vision-only methods suffers the shortage of decodable LEDs in reality.

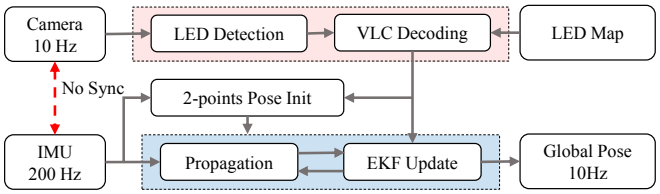


Fig. 1: Overview of the proposed VLC-inertial localization system.

In this work, we aim to overcome the challenge of the LED-shortage problem and thus to promote the adoption of VLC in modern buildings for lightweight global localization on service robots and mobile devices. To this end, we propose an EKF-based VLC-inertial localization system by tightly fusing the inertial measurements with the camera measurements of known LED markers. In particular, we use a low-end rolling-shutter camera and a MEMS-grade IMU sensor without hardware synchronization. As is shown in Fig. 1, a VLC frontend extracts LED features with known

2D-3D correspondences from the built map, by performing LED detection and VLC decoding on camera images. The EKF corrects the propagated IMU states using such absolute measurements and ensures globally consistent pose estimates free of drift. To initialize the filter's global pose, we introduce a 2-points method based on an IMU-aided P2P solution in [7]. By doing so, our EKF localizer can safely bootstrap from at least two point features from LEDs in one camera view, and enable failure recovery in extreme scenarios of long-term LED outages. The main contributions are as follows:

- An EKF-based tightly coupled VLC-inertial localization system is proposed. It provides lightweight real-time accurate global pose estimates in LED-shortage scenarios.
- A 2-points global pose initialization method is integrated to aid system bootstrapping and failure recovery.
- The system and method are verified by extensive field experiments with dozens of self-made LEDs.

The remainder of this paper is organized as follows. Section II introduces the related works. Section III and Section IV explain our VLC frontend and EKF-based localizer, respectively. Section V presents the experimental results. Section VI concludes this paper.

II. Related Works

A. VLC-based Localization

VLC-based systems [8]–[12] employ modulated LEDs of known locations as visual landmarks, use cameras or photodiodes, measure the sensor's bearings or ranges from visible LEDs, associate each measurement with a specific LED using its unique identity from VLC decoding, and solve the location using the measured constraints. Geometry-based methods (e.g., triangulation) need at least three LED features at a time to fix the sensor pose. This is a major cause of their fragile performance in real situations with an insufficient number of usable LEDs. Several methods proposed to address this issue by fusing IMU measurements. Epsilon [11] employed an IMU, as well as a digital compass, to measure the optical sensor's absolute orientation w.r.t. the geomagnetic north. By involving tedious user intervention, it managed to locate the sensor using one LED observation with a meter-level accuracy in non-real time fashion. Epsilon may also suffer from large errors from the compass due to magnetic anomalies. [10] measured the roll and pitch directions around the gravity with an IMU. Due to the loosely-coupled nature, however, it can not handle the case of one LED feature only in one camera view, let alone the complete LED outage. We here propose a tightly-coupled method to fill this gap.

B. Pose Estimation with Fiducial Markers

The paper printable squared fiducials [6], [13] are popular artificial visual landmarks used for lightweight pose estimation in robot applications. Similar to modulated LEDs, the fiducial maker can be uniquely identified by its encoded code patterns from a camera image. Yet, each marker can provide four distinctive corner features. By the integration of inertial measurements, in either a loosely- or tightly-coupled manner, some methods [14]–[16] can provide very accurate and robust pose estimates with fiducials. They have a trivial solution to the pose initialization problem, as it is sufficient to determine the camera pose from a single observation of known fiducials. By contrast, it is more technically difficult to obtain an initial pose guess for our system, especially under the LED shortage. We note that fiducial-based methods are suitable for specialized workspaces for robots, such as warehouses, factories, and laboratories, in which the environmental appearance is of no concern. However, fiducials may look unappealing or even weird in daily environments, such as shopping malls and museums. As an alternative, our LED-based systems can be naturally compatible with most daily scenarios, as well as some specialized workspaces.

III. VLC with a Rolling-shutter Camera

The time-varying light signals from LEDs are perceived by the rolling-shutter camera as spatially-varying strip patterns. We intend to retrieve the encoded VLC messages from such barcode-like patterns. To do so, we first extract candidate regions from the image that may contain possible LEDs. For each of them, we try to decode its unique identity (ID) and find its normalized centroid pixel as feature measurements. After that, we can obtain its absolute 3D position from the registered LED map.

A. VLC Preliminaries

We consider a rolling-shutter camera with row exposure time τ_e and row read-out time τ_r. The effective sampling rate, also known as the rolling-shutter frequency [17], is $f_s = 1/\tau_r$. We assume that the LED transmitter can switch on and off under the control of binary signals. We use an on-off-keying (OOK) modulation scheme with Manchester coding for data packaging. The OOK modulation frequency is $f_m = 1/\tau_m$ with τ_m as the sampling interval. That is, τ_m is the minimum pulse duration in the modulated binary signals. The upper bound of the square wave fundamental frequency is $f_h = f_m/2$. To recover the signals, the Nyquist sampling theorem must apply[1], i.e., $f_h < f_s/2$ and hence $f_m < f_s$. The modulated pulses are captured by the camera as bright or dark strips with varying widths proportional to the pulse durations. The minimum strip width, measured in pixels, is computed as $\mathrm{w}_0 = \tau_m/\tau_r$. An L-bit long data packet yields a strip pattern extending $\mathrm{w}_0 L$ pixels in height. That is, to recover the complete information carried by the data packet, we need a strip pattern with at least $\mathrm{w}_0 L$ rows of pixels. The pattern size is bounded by the image height H, i.e., $\mathrm{w}_0 L \leq H$. It follows $\tau_r < \tau_m \leq \tau_r H/L$.

We further consider a circular-shaped LED of diameter A. The maximum image size S of the LED radiation surface at a given distance d is described by $S = Af/d$, where f is the

[1]We consider the fundamental frequency components for analysis convenience. In a more strict sense, we should consider high-order harmonics of the square wave signals. For example to recover its third-order harmonics, we should have $3f_h < f_s/2$ and $f_m < f_s/3$.

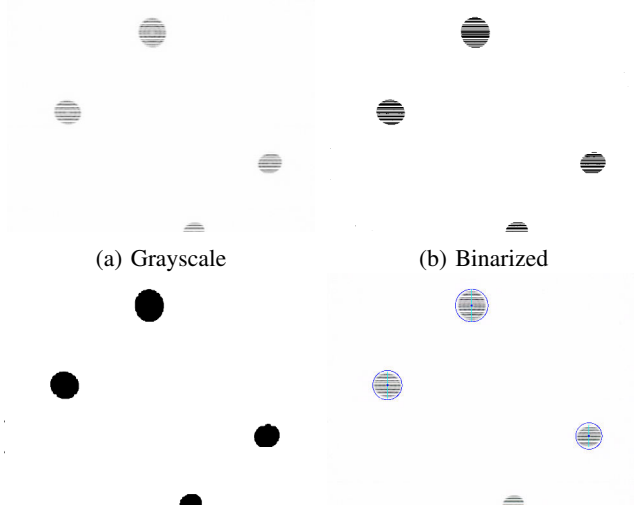
camera focal length in pixels. The data packet is decodable only if the condition $S \geq \mathrm{w}_0 L$ holds:

$$d \leq d_{\mathrm{m}} = \frac{Af}{\mathrm{w}_0 L} = \frac{\tau_r A f}{\tau_m L} \tag{1}$$

where d_{m} is the maximum range for VLC decoding. The determining factors include the focal length f and row read-out time τ_r of the rolling-shutter camera, the radiation surface size A and the OOK modulation interval τ_m of the LED transmitter, and also the data packet size L in use.

B. Protocol Definition

The designed data packet begins with a 4-bit preamble $\mathrm{PS} = \mathrm{b0001}$, precedes with a 16-bit Manchester-coded data payload DATA, and ends with another 4-bit symbol $\mathrm{ES} = \mathrm{b0111}$. This format yields a 24-bit long data packet with balanced DC-components to circumvent the LED flicker issue. The payload carries one byte of IDs (e.g., labeling 256 LEDs). The channel capacity can be extended by a larger payload. Yet, we are motivated to improve the maximum VLC decoding distance d_{m} by using a smaller packet size L, as is suggested by Eq. (1), due to hardware limitations in our implementation. To do so, we further omit any special packet section for error checking or data recovery.



(a) Grayscale (b) Binarized (c) Dilated (d) ROI detected

Fig. 2: An illustration of the image processing pipeline. We show inverted images for better visualization on paper.

C. LED Detection

Rolling-shutter cameras can capture strip patterns from a flashing LED during underexposure. Natural features are not observable, while bright objects (e.g., LEDs) can be easily distinguished. Normally, the strips are parallel to image rows and interleaving in the column direction. We are interested in those regions as they carry VLC information. To locate the bright blobs in the image and extract such regions of interest (ROI), we first binarize the grayscale image by thresholding. We then dilate the binary image in the column direction to fill strip gaps. After that, the bright strips from a given LED can join together as a connected blob. We detect blobs and retain large ones as ROIs for subsequent VLC decoding, as they are more likely to carry a complete data packet. We crop the grayscale image using the ROI masks and send the cropped images to the VLC decoder. Fig. 2 illustrates the key steps of our image processing pipeline. In addition, the centroid pixel for each ROI is undistorted and normalized with the calibrated camera intrinsics, as image measurements of the LED feature. Note that the perspective projection of the LED (e.g., a circle) centroid, in general, do not squarely coincide with the centroid of the LED image (e.g., an oval). Yet in practice, such an approximation error is acceptable for small objects and can be accommodated by the image noise.

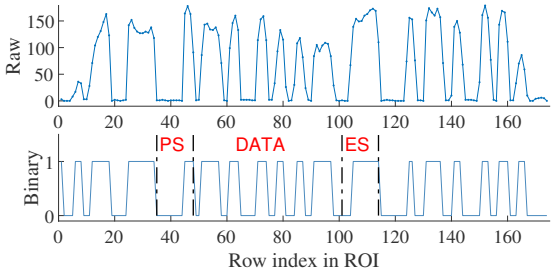


Fig. 3: An example of the 1D intensity signal for VLC decoding. The raw signal comprises the grayscale pixels in the center column of the top ROI, as is shown in Fig. 2. The segments of the binary signal, marked with the following symbols: PS, DATA and ES, constitute a complete data packet defined by our protocol.

D. VLC Decoding

VLC information is encoded by strips of varying widths. As is shown in Fig. 3, we pick up the grayscale pixels in the centering column for each ROI. We consider the column pixels as 1D time-varying intensity signals, as the camera's sampling frequency is fixed and known. The binary versions are used for OOK demodulation and Manchester decoding. We adopt adaptive thresholding to counter the nonuniform illumination artifacts of LEDs [18]. Now we can obtain the LED's ID from the decoding result. The data packet may start at a random location in an ROI due to the asynchronous communication mechanism. Sometimes, only shifted packet versions are available in some ROIs. To address this problem, we adopt a bidirectional decoding scheme [17] to improve decoding success rates. Note that decoding mistakes may happen due to the lack of a special data integrity checking mechanism in our protocol. Therefore, the pose estimator should be resilient to possible data association errors.

E. Implementation Details

We customize dozens of battery-powered LEDs as VLC transmitters. The LED has a circular radiation surface of diameter $15.5\mathrm{cm}$. The rating power is around 3 watts. We employ a cheap microcontroller to run the VLC protocol on its firmware and use a MOSFET transistor for driving the LED current. The modulation frequency f_m is set to $16\mathrm{kHz}$. We use a Raspberry Pi rolling-shutter camera (Sony IMX219 with a vertical FoV of $48.8\,\mathrm{deg}$) as the VLC receiver which

has a focal length of 1284 pixels under the image resolution of 1640 by 1232. We manually adjust the camera exposure time to capture sharp patterns. Further, we can derive the maximum decoding range of our VLC system from Eq. 1. The following quantities are known: $\tau_m = 1/16\text{kHz} = 62.5\mu\text{s}$, $A = 0.155\text{m}$, $f = 1284$, and $L = 24$. As we fail to find τ_r from the sensor datasheet, we determine the minimum strip width by experiments, e.g., $\text{w}_0 = \tau_m/\tau_r \approx 3$. It follows $\tau_r \approx 20.8\mu\text{s}$ and $f_s \approx 48\text{kHz}$. Now, we can compute the upper bound distance as $d_\text{m} = \tau_r A f/\tau_m L \approx 2.76\text{m}$.

IV. Global Localization by EKF

We consider an indoor environment installed with modulated LED lights at fixed locations (e.g., on the ceiling) that can be registered in a LED map. The EKF localizer uses the camera observations to known LED features extracted by the VLC frontend to correct its state estimates, after bootstrapping from 2-points global pose initialization.

A. Notations

We define a gravity-aligned global reference frame $\{G\}$ with its z-axis pointing upwards to the ceiling. The gravity vector expressed in $\{G\}$ is ${}^G\mathbf{g} = [0,\ 0,\ -g]$. The IMU frame $\{I\}$ and camera frame $\{C\}$ are rigidly connected. Yet, the IMU and camera sensors are not hardware synchronized. The IMU-camera spatial transformation can be obtained from offline calibration or manual measurements. To account for calibration inaccuracy, we further include these extrinsic parameters in the filter state for refinement by joint estimation. Besides this, the time offset t_d between the two sensors is assumed unknown. We use the IMU time as time reference (i.e., $t_{imu} = t_{cam} + t_d$) by following the convention in [19]. For a camera image timestamped at t, its actual sampling time instance is $t + t_d$. We use the unit quaternion ${}^A_B\bar{\mathbf{q}}$ under JPL convention [20] to represent the rotation ${}^A_B\mathbf{R}$ from frame $\{B\}$ to $\{A\}$, i.e., ${}^A_B\mathbf{R} = \mathbf{R}\left({}^A_B\bar{\mathbf{q}}\right)$. $\otimes$ denotes the quaternion multiplication. $\lfloor \cdot_\times \rfloor$ denotes the skew-symmetric matrix. For a quantity $\mathbf{a}$, we use $\hat{\mathbf{a}}$ for its estimate and $\tilde{\mathbf{a}}$ for the residue.

B. Filter State Definition

The IMU state $\mathbf{x}_I \in \mathbb{R}^{24}$ is defined as follows [19]:

$$\mathbf{x}_I = \left[{}^I_G\bar{\mathbf{q}}^\top\ {}^G\mathbf{p}_I^\top\ {}^G\mathbf{v}_I^\top\ \mathbf{b}_g^\top\ \mathbf{b}_a^\top\ {}^C_I\bar{\mathbf{q}}^\top\ {}^C\mathbf{p}_I^\top\ t_d\right]^\top \tag{2}$$

where ${}^I_G\bar{\mathbf{q}}$ is the unit quaternion that describes the rotation ${}^I_G\mathbf{R}$ from $\{G\}$ to $\{I\}$, i.e., ${}^I_G\mathbf{R} = \mathbf{R}\left({}^I_G\bar{\mathbf{q}}\right)$; ${}^G\mathbf{p}_I$ and ${}^G\mathbf{v}_I$ are the global IMU position and velocity, respectively; $\mathbf{b}_g$ and $\mathbf{b}_a$ are the gyroscope and accelerometer biases; ${}^C_I\bar{\mathbf{q}}$ is the unit quaternion that represents the rotation ${}^C_I\mathbf{R}$ from the IMU frame $\{I\}$ to the camera frame $\{C\}$; ${}^C\mathbf{p}_I$ denotes the IMU position in $\{C\}$; and t_d is the time offset.

The error state $\tilde{\mathbf{x}}_I \in \mathbb{R}^{22}$ is then given by:

$$\tilde{\mathbf{x}}_I = \left[{}^I\tilde{\boldsymbol{\theta}}^\top\ {}^G\tilde{\mathbf{p}}_I^\top\ {}^G\tilde{\mathbf{v}}_I^\top\ \tilde{\mathbf{b}}_g^\top\ \tilde{\mathbf{b}}_a^\top\ {}^I\tilde{\boldsymbol{\phi}}^\top\ {}^C\tilde{\mathbf{p}}_I^\top\ \tilde{t}_d\right]^\top \tag{3}$$

where for quaternions, we employ the multiplicative error definition with local perturbations in the IMU frame. That is, we have ${}^I_G\bar{\mathbf{q}} \simeq \begin{bmatrix}\frac{1}{2}{}^I\tilde{\boldsymbol{\theta}}\\ 1\end{bmatrix} \otimes {}^I_G\hat{\bar{\mathbf{q}}}$ and ${}^C_I\bar{\mathbf{q}} \simeq {}^C_I\hat{\bar{\mathbf{q}}} \otimes \begin{bmatrix}\frac{1}{2}{}^I\tilde{\boldsymbol{\phi}}\\ 1\end{bmatrix}$ where ${}^I\tilde{\boldsymbol{\theta}}$ and ${}^I\tilde{\boldsymbol{\phi}}$ are the 3×1 small angle rotation error vectors expressed in $\{I\}$. The standard additive errors apply to other quantities, e.g., ${}^G\mathbf{p}_I = {}^G\hat{\mathbf{p}}_I + {}^G\tilde{\mathbf{p}}_I$.

C. IMU Propagation

The IMU measures the true angular velocity ${}^I\boldsymbol{\omega}$ and linear acceleration ${}^I\mathbf{a}$ in its local frame $\{I\}$ as [19]:

$$\boldsymbol{\omega}_m = {}^I\boldsymbol{\omega} + \mathbf{b}_g + \mathbf{n}_g, \quad \mathbf{a}_m = {}^I\mathbf{a} - {}^I_G\mathbf{R}\,{}^G\mathbf{g} + \mathbf{b}_a + \mathbf{n}_a \tag{4}$$

where the measurements $\boldsymbol{\omega}_m$ and $\mathbf{a}_m$ are corrupted by zero-mean white Gaussian noises, $\mathbf{n}_g \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}_g^2)$ and $\mathbf{n}_a \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}_a^2)$, respectively. The continuous-time dynamics of the IMU state $\mathbf{x}_I$ is given by:

$$\begin{aligned}&{}^I_G\dot{\bar{\mathbf{q}}} = \frac{1}{2}\boldsymbol{\Omega}\left({}^I\boldsymbol{\omega}\right){}^I_G\bar{\mathbf{q}},\ {}^G\dot{\mathbf{p}}_I = {}^G\mathbf{v}_I,\ {}^G\dot{\mathbf{v}}_I = {}^I_G\mathbf{R}^\top\,{}^I\mathbf{a},\\ &\dot{\mathbf{b}}_g = \mathbf{n}_{wg},\ \dot{\mathbf{b}}_a = \mathbf{n}_{wa},\ {}^C_I\dot{\bar{\mathbf{q}}} = \mathbf{0},\ {}^C\dot{\mathbf{p}}_I = \mathbf{0},\ \dot{t}_d = 0\end{aligned} \tag{5}$$

where $\boldsymbol{\Omega}({}^I\boldsymbol{\omega}) = \begin{bmatrix}-\lfloor {}^I\boldsymbol{\omega}_\times \rfloor & {}^I\boldsymbol{\omega}\\ -{}^I\boldsymbol{\omega}^\top & 0\end{bmatrix}$; and $\mathbf{n}_{wg}$ and $\mathbf{n}_{wa}$ are the underlying noise processes that drive the IMU biases, with $\mathbf{n}_{wg} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}_{wg}^2)$ and $\mathbf{n}_{wa} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}_{wa}^2)$. The propagation of nominal state $\hat{\mathbf{x}}_I$ derives from the expectation of Eq. (5):

$$\begin{aligned}&{}^I_G\dot{\hat{\bar{\mathbf{q}}}} = \frac{1}{2}\boldsymbol{\Omega}\left(\hat{\omega}\right){}^I_G\hat{\bar{\mathbf{q}}},\ {}^G\dot{\hat{\mathbf{p}}}_I = {}^G\hat{\mathbf{v}}_I,\ {}^G\dot{\hat{\mathbf{v}}}_I = {}^I_G\hat{\mathbf{R}}^\top\hat{\mathbf{a}} + {}^G\mathbf{g},\\ &\dot{\hat{\mathbf{b}}}_g = \mathbf{0},\ \dot{\hat{\mathbf{b}}}_a = \mathbf{0},\ {}^C_I\dot{\hat{\bar{\mathbf{q}}}} = \mathbf{0},\ {}^C\dot{\hat{\mathbf{p}}}_I = \mathbf{0},\ \dot{\hat{t}}_d = 0\end{aligned} \tag{6}$$

where $\hat{\boldsymbol{\omega}} = \boldsymbol{\omega}_m - \hat{\mathbf{b}}_g$, $\hat{\mathbf{a}} = \mathbf{a}_m - \hat{\mathbf{b}}_a$, and ${}^I_G\hat{\mathbf{R}} = \mathbf{R}\left({}^I_G\hat{\bar{\mathbf{q}}}\right)$. We can now predict $\hat{\mathbf{x}}_I$ in discrete-time by numerical integration.

The linearized continuous-time error state equation is written as follows:

$$\dot{\tilde{\mathbf{x}}}_I = \mathbf{F}\,\tilde{\mathbf{x}}_I + \mathbf{G}\,\mathbf{n}_I \tag{7}$$

with

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

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

where $\mathbf{F}$ is the system matrix, $\mathbf{G}$ is the noise input matrix, and $\mathbf{n}_I = \left[\mathbf{n}_g^\top\ \mathbf{n}_{wg}^\top\ \mathbf{n}_a^\top\ \mathbf{n}_{wa}^\top\right]^\top$ is the continuous-time IMU noise with its covariance as $\mathbf{Q}_c = \mathbf{diag}\{\boldsymbol{\sigma}_g^2, \boldsymbol{\sigma}_{wg}^2, \boldsymbol{\sigma}_a^2, \boldsymbol{\sigma}_{wa}^2\}$.

Then, we compute the discrete-time error state transition matrix $\boldsymbol{\Phi}_k$ and the discrete-time noise covariance matrix $\mathbf{Q}_d$

as: $\mathbf{\Phi}_k \simeq \mathbf{I}+\mathbf{F}(t_k)\Delta t$ and $\mathbf{Q}_d \simeq \mathbf{G}(t_k)\mathbf{Q}_c\mathbf{G}^\top(t_k)\Delta t$. Here, $\mathbf{F}(t)$ is a constant over a small time interval $\Delta t = t_{k+1} - t_k$. The EKF then propagates the state covariance $\mathbf{P}$ from t_k to t_{k+1} according to $\mathbf{P}_{k+1|k} = \mathbf{\Phi}_k \mathbf{P}_{k|k} \mathbf{\Phi}_k^\top + \mathbf{Q}_d$.

D. Camera Measurement Update

We assume a calibrated pinhole camera model[2]. For an image timestamped at t, we consider the ith feature f_i of the decoded LEDs from the VLC frontend. Its measurement $\{\mathbf{z}_i, {}^G\mathbf{p}_{f_i}\}$ is known, where $\mathbf{z}_i$ is the normalized pixel of the LED centroid and ${}^G\mathbf{p}_{f_i}$ is the global LED position. The feature observation $\mathbf{z}_i$ taken at camera time t is given by:

$$\mathbf{z}_i(t) = \mathbf{h}\left({}^C\mathbf{p}_{f_i}(t+t_d)\right) + \mathbf{n}_{im}(t+t_d) \tag{8}$$
$${}^C\mathbf{p}_{f_i}(t+t_d) = {}^C_I\mathbf{R}\,{}^I_G\mathbf{R}(t+t_d)\left({}^G\mathbf{p}_{f_i} - {}^G\mathbf{p}_I(t+t_d)\right) + {}^C\mathbf{p}_I$$

where $\mathbf{n}_{im} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}_{im}^2)$ is the image noise; $\mathbf{h}(\cdot)$ is the perspective projection function, i.e., $\mathbf{h}\left([x,y,z]^\top\right) = [x/z, y/z]^\top$; and ${}^C\mathbf{p}_{f_i}$ is the feature position with respect to the current camera frame at IMU time $t+t_d$.

With the latest state estimate $\hat{\mathbf{x}}_I(t+\hat{t}_d)$ from the IMU propagation, we can now derive the expected measurement as $\hat{\mathbf{z}}_i(t) = \mathbf{h}\left({}^C\hat{\mathbf{p}}_{f_i}(t+\hat{t}_d)\right)$, and computed its residue term $\tilde{\mathbf{z}}_i = \mathbf{z}_i - \hat{\mathbf{z}}_i$ by first-order approximation:

$$\tilde{\mathbf{z}}_i \simeq \mathbf{H}_{\mathbf{x},i}\tilde{\mathbf{x}}_I + \mathbf{H}_{f_i}\,{}^G\tilde{\mathbf{p}}_{f_i} + \mathbf{n}_{im} \tag{9}$$

where the LED position error ${}^G\tilde{\mathbf{p}}_{f_i}$ is modeled as zero-mean white Gaussian noise with covariance $\boldsymbol{\sigma}_f^2$; the measurement Jacobian w.r.t. the IMU state $\mathbf{H}_{\mathbf{x},i}$ and the Jacobian w.r.t. the LED feature position $\mathbf{H}_{f_i}$ are given by:

$$\begin{aligned}
&\mathbf{H}_{\mathbf{x},i} = [\mathbf{H}_{\boldsymbol{\theta},i} \quad \mathbf{H}_{\mathbf{p},i} \quad \mathbf{0}_{2\times 9} \quad \mathbf{H}_{\boldsymbol{\phi},i} \quad \mathbf{H}_{\mathbf{p}_c,i} \quad \mathbf{H}_{t_d,i}] \\
&\mathbf{H}_{\boldsymbol{\theta},i} = \mathbf{J}_i\,{}^C_I\hat{\mathbf{R}}\,\lfloor {}^I_G\hat{\mathbf{R}}\left({}^G\hat{\mathbf{p}}_{f_i} - {}^G\hat{\mathbf{p}}_I\right)_\times \rfloor \\
&\mathbf{H}_{\mathbf{p},i} = -\mathbf{J}_i\,{}^C_I\hat{\mathbf{R}}\,{}^I_G\hat{\mathbf{R}},\ \mathbf{H}_{\boldsymbol{\phi},i} = \mathbf{H}_{\boldsymbol{\theta},i},\ \mathbf{H}_{\mathbf{p}_c,i} = \mathbf{J}_i \\
&\mathbf{H}_{t_d,i} = \mathbf{H}_{\boldsymbol{\theta},i}\,\hat{\boldsymbol{\omega}} + \mathbf{H}_{\mathbf{p},i}\,{}^G\hat{\mathbf{v}}_I,\ \mathbf{H}_{f_i} = -\mathbf{H}_{\mathbf{p},i}
\end{aligned} \tag{10}$$

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 in the camera frame ${}^C\hat{\mathbf{p}}_{f_i} = [\hat{x},\hat{y},\hat{z}]^\top$, i.e., $\mathbf{J}_i = \frac{1}{\hat{z}}\begin{bmatrix}1 & 0 & -\hat{x}/\hat{z}\\ 0 & 1 & -\hat{y}/\hat{z}\end{bmatrix}$.

The filter state and covariance estimates can be updated by following the general EKF equations [20]. To account for false data associations from VLC decoding errors, we perform the Mahalanobis gating test for each observation before the measurement update. The EKF can naturally process multiple LED observations in a single image if more LEDs are successfully decoded.

E. 2-points Pose Initialization

For global localization, we need to initialize the EKF with a six degrees-of-freedom (DoF) pose and its velocity w.r.t. the global frame. Since vision-only PnP methods easily suffer from large errors or failure in LED-shortage scenarios, we steer to an IMU-aided P2P solution that can work more reliably with two point-correspondences [7]. IMU measures the roll and pitch directions accurately by monitoring gravity, leaving four unknowns in the camera pose. It has been proved that there are two closed-form solutions to this problem with two feature measurements. In our applications, moreover, we can obtain a unique solution by checking its z-direction as the sensor-suite is alway beneath the ceiling. We refine the camera pose by minimizing camera re-projection errors once more than two LEDs are decoded in the image. Specially, we use the P2P solution as an initial guess and optimize the pose in 4-DoF by fixing the roll and pitch directions. The IMU-centric pose can be resolved given the sensor extrinsic parameters. The velocity computed from two consecutive poses is noisy and unreliable to use, especially when the sensor moves slowly, e.g., for handheld cases and low-speed robots. Alternatively, we provide the filter with zero velocity and a large velocity variance. So far, our system can bootstrap with two or more point features from the LEDs decoded in a single image.

[2]In this section, we assume a simplified global-shutter camera measurement model without considering the rolling-shutter effect on feature measurements. We leave this issue for our future work.

V. Experiments

We evaluate our system through real-world experiments. We use the absolute trajectory error (ATE) for global position accuracy and use the axis-angle error for orientation accuracy assessment. We set up a room-sized (5*4*2.3 m^3) test field with 25 LEDs evenly mounted on the ceiling (see Fig. 4). The spacing is around 1-1.5m. We use a customized sensor suite for data collection, as is shown by the right side of Fig. 4. It comprises a Raspberry Pi camera (IMX219, 1640*1232 @10Hz) and a MicroStrain IMU (3DM-GX3-25 @200Hz) without hardware synchronization. The motion capture system (OptiTrack Mocap @120Hz) provides ground truth poses for our experiments. We set the Mocap world frame to coincide with the global frame $\{G\}$. The extrinsic parameters between the camera and the Mocap rigid body (i.e., reflective markers on the sensor suite) is known from hand-eye calibration. We measured the global 3D location of LEDs using Mocap as well as a commodity laser ranger for height compensation. The algorithm runs on a desktop computer (Intel i7-7700K CPU @4.20GHz, 16G RAM).

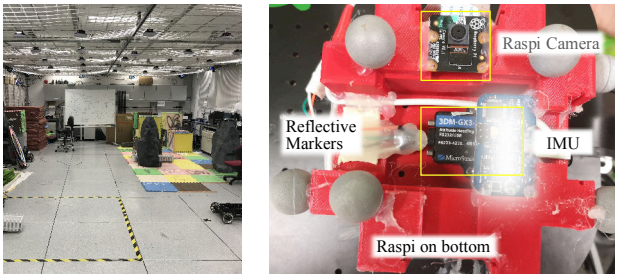


Fig. 4: Test field (left) and self-assembled sensor suite (right).

A. VLC Decoding Performance

We aim to study the VLC performance delivered by our hardware setup under varying LED-to-camera distances. To do so, we first define the decoding success rate, for a given

LED, as the ratio of the number of images with correct decoding results to the total number of images captured in a certain period. We orient the camera against the out-coming direction of the LED's surface normal, vary the distance from 1m to 3m with a step length of 0.5m, and record a 60s long image stream at 10Hz at each distance. The respective decoding rates are computed and shown in Fig. 5. The success rate decreases with distance and drops quickly after 2m. This is because the captured LED pattern contains a less number of complete data packets at some large distances. The rate drops to 0 at 3m in which case the LED pattern is too small in size to be decoded. The maximum decoding range of our system is larger than yet close to 2.5m, which coincides with our previous reasoning (e.g., $d_m \approx 2.76\text{m}$).

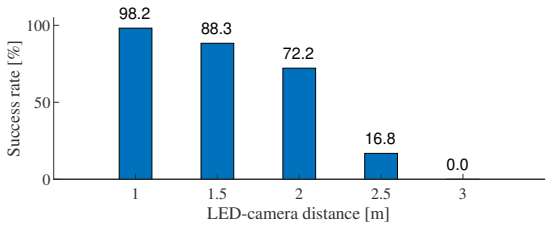


Fig. 5: The decoding performance of our VLC frontend.

B. Localization Performance

To assess the localization performance[3], we have collected a few datasets in eight trials [see Table I]. Specifically, we move the handheld sensor suite smoothly by walking in the test field. We orient the camera upwards facing the ceiling lights. For the ease of filter initialization, we put the sensor on the ground and keep it still for a few seconds at the start of each run. Unless otherwise specified, the global pose in EKF is initialized by the 2-points initialization method, which will later be evaluated in section V-D. The extrinsic parameters $\{{}^C_I\bar{\mathbf{q}}, {}^C\mathbf{p}_I\}$ are initialized with coarse manual measurements. The remaining parameters in the filter (e.g., the IMU biases and time offset) are simply set to zeros.

TABLE I: Description of the eight datasets in use.

Trials	1	2	3	4	5	6	7	8
Time [s]	39.5	33.4	40.7	34.6	66.8	43.4	67.7	133.5
Dist. [m]	30.2	37.1	35.0	27.8	67.6	42.0	69.0	158.6
MaxVel [m/s]	1.40	1.99	1.49	1.36	1.48	1.55	1.51	1.67
Shape	circle		square		eight		random	

Fig. 6 shows the results for trial 7 as we walk randomly in the field for 68s. We use Mocap to denote the ground truth and use EKF for the estimates. As is shown in Fig. 6a, the estimated trajectory well matches the ground truth. The global position, orientation and velocity estimates for this trial, as well as the respective errors compared against the ground truth, are shown in Fig. 7. We illustrate the number

[3]Online demonstrations can be found in our supplementary video.

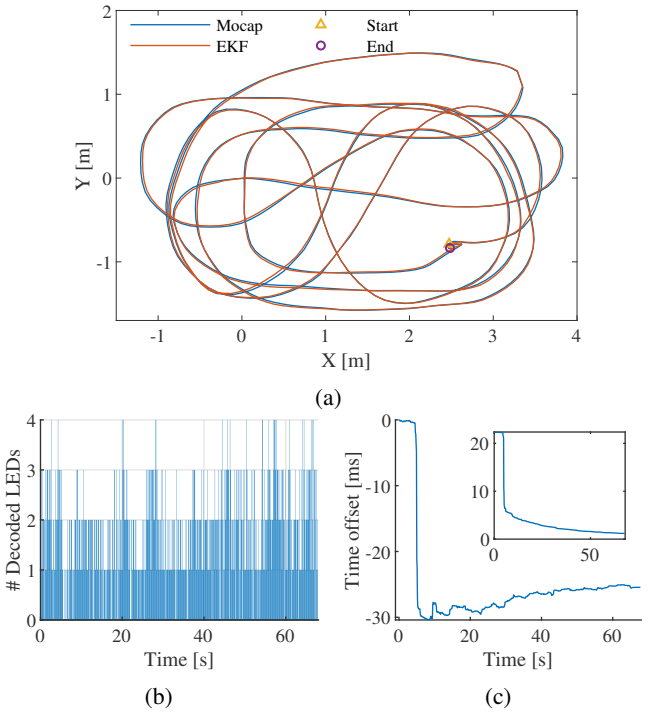
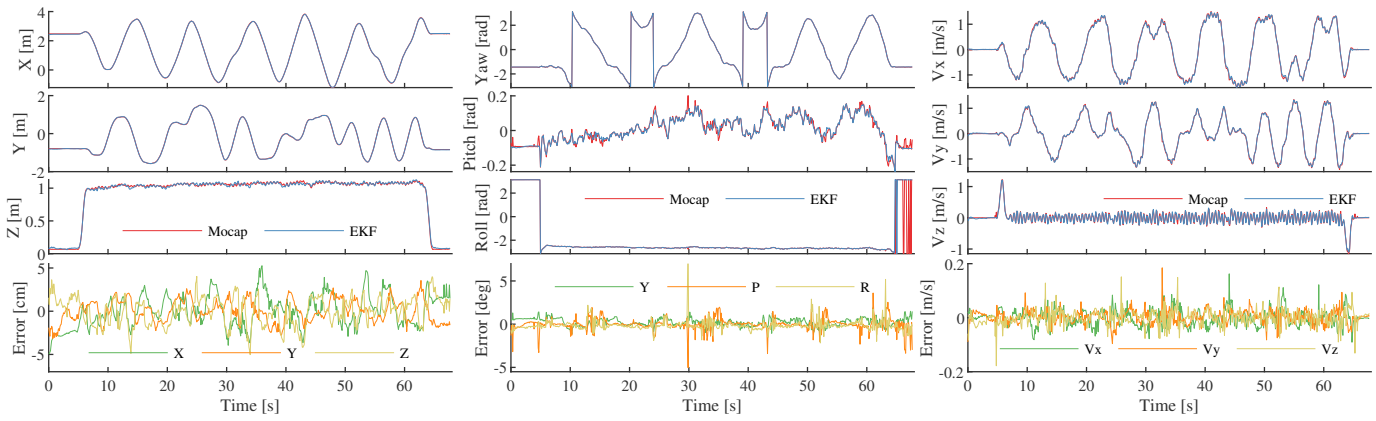


Fig. 6: The random trajectory (a) travels approximately 69m in 68s. The EKF estimates are very close to the ground truth by visual comparison. (b) shows the number of LEDs that are successfully decoded from each camera frame by the VLC frontend. (c) shows the time offset estimate as well as its uncertainty described by standard deviation, as is shown by the inner subplot.

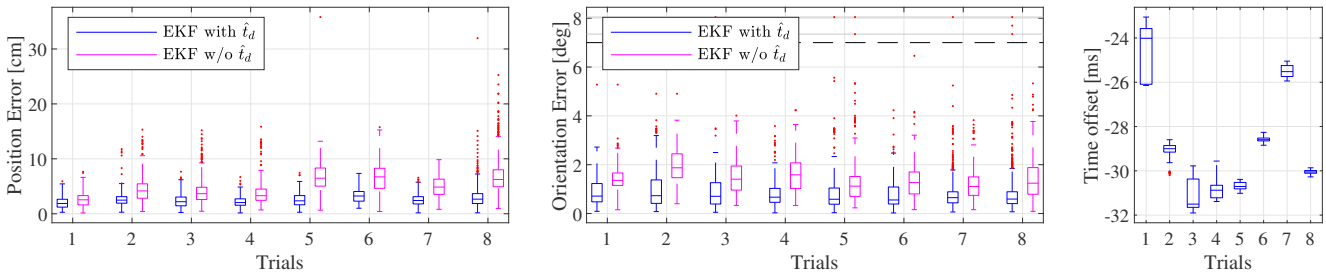
of decodable LEDs in each camera frame in Fig. 6b. On the one hand, we have a very low chance to decode three or more LEDs in one image despite the dense LED deployment. As such, vision-only methods can rarely be used in our setting. On the other hand, we can concurrently decode two LEDs at a much higher possibility, and thus, bootstrap the proposed system more easily by 2-points initialization. Furthermore, we show the time offset estimate $\hat{t}_d$ in Fig. 6c, as well as its standard deviation from the filter covariance matrix. It converges soon after the sensor starts moving.

The absolute pose errors for the eight trials are shown in Fig. 8, where the position error is evaluated by ATE and the orientation error is based on the axis-angle representation. Fig. 8c shows the time offset estimation results for the last 20s in each run. We can observe that most of these estimates are consistent, e.g., lying between -24ms and -32ms. There is no ground truth time offset for our sensor suite. It may even vary slightly in different runs due to the lack of hardware synchronization. Further, we study the impact of temporal calibration on our localization performance. With online time offset estimation activated, the proposed method significantly outperforms its counterpart without such a calibration, say on the eight trials in terms of both the global position accuracy and orientation accuracy. We note that the extreme outliers in orientation, as is shown in Fig. 8b, are most probably caused by the occasional Mocap tracking errors (especially the rotation) at certain places, e.g., due to the blockage of reflective markers by the experimenter. By revisiting the



(a) Position (b) Orientation (c) Velocity

Fig. 7: Global position (a), orientation (b), and velocity (c), as well as their respective errors in trial 7 compared with the ground truth.



(a) Position error (b) Orientation error (c) Time offset

Fig. 8: Absolute position (a) and orientation (b) errors on eight trials. We show the consistency of time offset estimates in (c) by using results over the last 20s, and compare the performance of the proposed method both with and without online temporal calibration.

TABLE II: Statistics on localization errors and counts of decoded LEDs in eight trials using both dense and sparse feature maps.

Trial		1	2	3	4	5	6	7	8
Position Error [cm]	RMSE	2.20 / 2.91	3.02 / 3.91	2.67 / 3.22	2.41 / 3.29	2.80 / 2.99	3.59 / 3.97	2.75 / 3.00	3.47 / **4.00**
	std	0.96 / 1.43	1.40 / 1.88	1.23 / 1.47	0.97 / 1.43	1.20 / 1.39	1.30 / 1.63	1.05 / 1.46	1.79 / 2.11
Rotation Error [deg]	RMSE	1.07 / 1.09	**1.27** / 1.25	1.12 / 1.15	1.00 / 0.97	1.22 / 1.21	0.99 / 1.00	1.10 / 1.12	1.04 / 1.06
	std	0.59 / 0.61	0.80 / 0.74	0.71 / 0.73	0.57 / 0.58	0.91 / 0.90	0.59 / 0.57	0.73 / 0.73	0.68 / 0.74
#LED	mean	1.59 / 0.86	1.38 / 0.69	1.21 / 0.67	1.21 / 0.69	1.20 / 0.65	1.60 / 0.91	1.79 / 0.95	1.16 / 0.59
Pct. of #LED	≥ 1	0.86 / 0.69	0.83 / 0.58	0.77 / 0.59	0.78 / 0.57	0.83 / 0.61	0.90 / 0.73	0.92 / 0.76	0.77 / 0.54
	≥ 2	0.52 / 0.15	0.42 / 0.11	0.35 / 0.08	0.36 / 0.12	0.32 / 0.05	0.52 / 0.17	0.61 / 0.18	0.33 / 0.05
	≥ 3	0.16 / 0.01	0.11 / 0.00	0.08 / 0.00	0.07 / 0.00	0.05 / 0.00	0.15 / 0.01	0.22 / 0.01	0.05 / 0.00

orientation plots in Fig. 7b, we observe that the yaw direction is consistently smooth while the roll and pitch directions have a few spikes (e.g., at the 30s). Since EKF estimates are normally smooth after converging, those spikes are most probably caused by the Mocap system.

C. Robustness Test under LED Shortage/Outage

We aim to explore the robustness of our system in more challenging scenarios, e.g., with less decodable LEDs in a single view (say LED shortage) or with the complete absence of LEDs in a certain period (say LED outage). These problems may arise from many practical factors, such as the lights deployment density and the maximum VLC decoding range supported by the hardware setup. We here look into the LED shortage problem by altering the deployment density. To do so, we uniformly remove half of the 25 LEDs from the original dense map. This results in a sparse map with 12 LEDs. We simply discard measurements from those removed ones. Unlike commercial lights for illumination, our prototyping LEDs have a much smaller radiation surface (e.g., 15cm in diameter), and thus have a reduced VLC decoding range. The 12 circular LEDs are reasonably sparse for localization in the test field. As a comparison, 10 pairs of standard fluorescent tubes are deployed in the same area.

Table. II summarizes the statistics on absolute position and rotation errors, along with the counts of decodable LEDs in the camera view. The results from the dense map are shown before that from the sparse map side by side. We show the root-mean-squared error (RMSE) and the respective standard

deviation for the estimated poses. We notice that the position errors increase as the map density decreases, e.g., with larger RMSE errors and standard deviations. Yet, we do not find any substantial variation in rotation errors. The maximum RMSE position error (e.g., 4 cm in trial 8) arise from the sparse map, while the maximum RMSE rotation error (e.g., 1.27 deg in trial 2) comes from the dense map. The average number of decodable LEDs in the sparse map is almost half of that in the dense map, indicating a substantial loss of usable LED features. In the meantime, the performance degradation in positioning accuracy is relatively marginal.

Also, we show the percentage of frames that can decode a certain number of LEDs in Table. II. The percentage of decoding three or more LEDs is extremely low, especially in the sparse map. Meanwhile, we have a much higher possibility to decode one or more LEDs. As we know, EKF can still keep correcting its estimates with one observation only. The chance remains to observe two decodable LEDs simultaneously in the sparse map. As such, our system can still bootstrap from 2-points initialization. Therefore, our method has better usability than those vision-only counterparts.

Further, we explore the system performance in situations with an intermediate LED outage. Specifically, we study the short-term outage problem by dropping different quantities of camera frames in a given period. For example, we can simulate an effective camera rate of 1Hz by dropping 9 out of 10 frames for every 1s. We choose five different camera rates from 10Hz to 0.5Hz. The respective pose errors are shown in Fig. 9. The system can bootstrap on its own at the camera rate of 1Hz. Aided by its normal initialization at 10Hz, the system can finally sustain at 0.5Hz without diverging. In other words, the system is tolerant to a certain short-term LED outage, e.g., less than 2s, during normal walking.

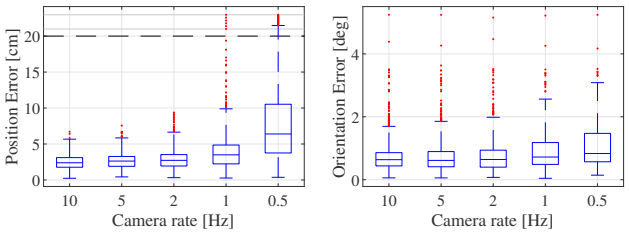


Fig. 9: Pose errors in trial 7 at different camera rates. The maximum position errors are 27cm at 1Hz and 37cm at 0.5Hz. Note that we manually remove an extreme rotation outlier at the 30s (around 10 deg) caused by Mocap tracking errors, as is illustrated by Fig. 7b.

D. 2-points Initialization & Failure Recovery

The 2-points global pose initialization is important for the filter bootstrapping and recovery from failure, e.g., due to the long-term LED outage. We first evaluate the accuracy of our IMU-aided P2P solution enabled with 4-DoF pose refinement, say by comparing with the ground truth and the EKF estimates. Aiming to observe more decodable LEDs in the camera view, we mount the sensor suite on a low-speed wheeled robot and orient the camera facing the ceiling lights straight upwards. The robot follows a square figure

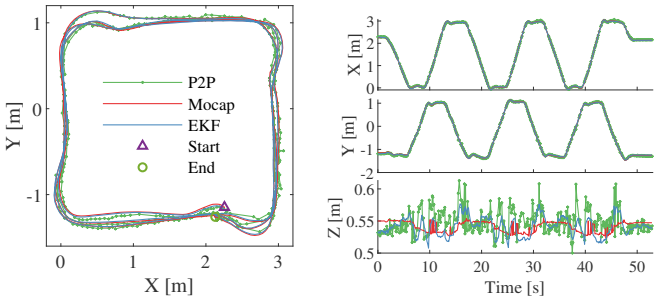


Fig. 10: Trajectory (left) and position estimates (right) of a wheeled robot by our IMU-aided P2P solution, in comparison with the results from our EKF localizer and the Mocap ground truth.

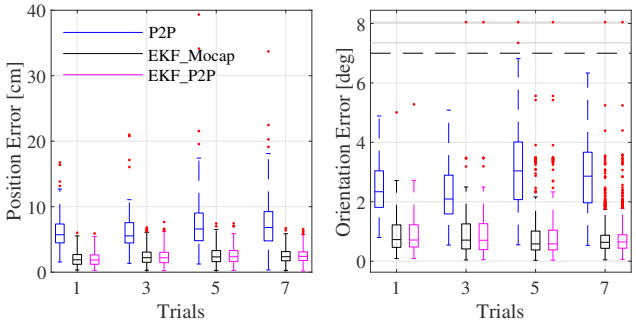


Fig. 11: Pose errors evaluated on trial 1, 3, 5, and 7. We compared the results from P2P, the EKF initialized by the Mocap ground truth, and the EKF initialized by P2P. There is no statistically significant difference in performance between the latter two cases.

by teleoperation. The trajectory and position estimates are shown in Fig. 10. The results of our IMU-aided P2P solution (i.e., denoted by P2P) can match the ground truth. Yet, the position estimates of P2P are noisier than the EKF results and the respective trajectory is less smoother.

Indeed, we are more interested in investigating the impact of the initial pose guess on the overall localization performance. To do so, we initialize the filter using both the P2P-based solution and the ground truth. Fig. 11 shows the pose errors on trial 1, 3, 5, and 7. We use EKF-Mocap for indicating the results from the Mocap-initialized EKF while using EKF-P2P for the P2P-initialized EKF. The P2P acts as a baseline for comparison. We notice that P2P suffers from larger pose errors. Even though, we can achieve a median position error around 5 cm and a median orientation error around 3 degrees. The pose estimation results from both EKF-Mocap and EKF-P2P are almost the same. We can not find any statistically significant difference. So far, we may safely prove the efficacy of the proposed 2-points initialization method for filter bootstrapping.

Fig. 12 illustrates the case of failure recovery under the long-term outage, where we take trial 7 as an example. We manually introduce four outage periods: T1 and T2 last for 5s, while T3 and T4 last for 10s. In the first two periods, the filter begins to diverge after losing LED observations but can converge again once new features are available. The increasing position error is well estimated by EKF, as is indicated by the error plot of Fig. 12. In the latter two cases, the filter uncertainty grows too high that the failure recovery

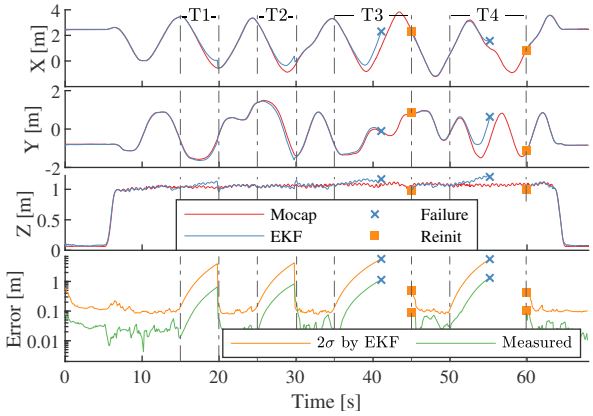


Fig. 12: Position estimates in trial 7 under longer periods of outage: T1=[15s 20s], T2=[25s 30s], T3=[35s 45s], T4=[50s 60s]. Besides this, we show the two times error standard deviation 2σ estimated by EKF, as well as the measured position error in the error plot. It is shown in log-scale for better visualization.

mechanism is triggered, preventing the output of erroneous estimates. The filter can be reinitialized soon after the camera observes two or more decodable LEDs.

E. Runtime Analysis

To evaluate the runtime efficiency, we also run the proposed algorithm on a Raspberry Pi 3B single-board computer (Cortex-A53 @1.2GHz, 1G RAM). We implement two threads: one for VLC decoding and the other for EKF estimation. We summarize the average runtime to process an image taken by each thread in Table. III. The runtime is dominated by the VLC thread. The algorithm efficiency can be improved by optimizing the image processing pipeline for VLC decoding. Nevertheless, we can achieve real-time performance on Raspberry Pi 3B without any code optimization for ARM processors, considering a camera rate of 10Hz. The proposed VLC-inertial localization system is hence lightweight to use on resource-constrained computational platforms.

TABLE III: Runtime statistics.

Module	VLC (Thread 1)	EKF (Thread 2)
Desktop PC	2.3 ms	0.7 ms
Raspberry Pi 3B	40.5 ms	9.7 ms

F. Discussions

The current prototyping system suffers a few limitations. We use only circular LEDs of the same form factor for experiments. This is due to the difficulty in preparing more types of LEDs, with customized hardware modifications, for VLC functionality. The size of workspace supported by our system is limited by the usable VLC channel capacity (i.e., the maximum number of encoded LEDs), which is decided by the data payload. By using LEDs of a larger radiation diameter, we can safely choose a larger payload. As for the camera measurement model, we resort to a "global-shutter" model for the sake of simplicity. We plan to integrate the rolling-shutter measurement model to our system in the future work for better performance.

VI. Conclusion

This paper presented an EKF-based tightly coupled VLC-inertial localization system by using modulated LED lights in modern buildings as artificial visual landmarks, especially for achieving lightweight global localization on resource-constrained platforms. Our system employed an inexpensive rolling-shutter camera and an unsynchronized IMU sensor. The EKF localizer tightly fused IMU measurements with the camera measurements of known LED features from VLC decoding. We further completed our system by the 2-points global pose initialization for filter bootstrapping and failure recovery. Our system managed to be bootstrapped from two and more LED features in a single image, and then sustained by EKF. The system and methods were verified by extensive field experiments in a Mocap-room mounted with dozens of LED prototypes. It was shown that our system can reliably provide lightweight, real-time and accurate global pose estimates in LED-shortage situations. The robustness under short-term LED outage, as well as the failure recovery behavior under long-term outage, was also demonstrated.

References

[1] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, 2017.

[2] D. Gálvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Trans. Robot.*, vol. 28, no. 5, pp. 1188–1197, 2012.

[3] J. Zhang and S. Singh, "LOAM: Lidar Odometry and Mapping in Real-time," in *Robotics: Science and Systems*, vol. 2, 2014, p. 9.

[4] H. Ye, Y. Chen, and M. Liu, "Tightly Coupled 3D Lidar Inertial Odometry and Mapping," in *Proc. ICRA*. IEEE, 2019.

[5] 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.

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

[7] Z. Kukelova, M. Bujnak, and T. Pajdla, "Closed-form solutions to minimal absolute pose problems with known vertical direction," in *Proc. ACCV*. Springer, 2010, pp. 216–229.

[8] 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.

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

[10] 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.

[11] 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.

[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] R. Munoz-Salinas, M. J. Marin-Jimenez, and R. Medina-Carnicer, "SPM-SLAM: Simultaneous localization and mapping with squared planar markers," *Pattern Recognition*, vol. 86, pp. 156–171, 2019.

[14] L. Meier, P. Tanskanen, L. Heng, G. H. Lee, F. Fraundorfer, and M. Pollefeys, "Pixhawk: A micro aerial vehicle design for autonomous flight using onboard computer vision," *Autonomous Robots*, vol. 33, no. 1-2, pp. 21–39, 2012.

[15] M. Neunert, M. Bloesch, and J. Buchli, "An open source, fiducial based, visual-inertial motion capture system," in *Proc. FUSION*. IEEE, 2016, pp. 1523–1530.

[16] G. He, S. Zhong, and J. Guo, "A lightweight and scalable visual-inertial motion capture system using fiducial markers," *Autonomous Robots*, pp. 1–21, 2019.
[17] 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.
[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] 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.
[20] 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.