This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIM.2021.3123293, IEEE Transactions on Instrumentation and Measurement

LedMapper: Towards efficient and accurate LED mapping for visible light positioning at scale

Qing Liang, Yuxiang Sun, Chengju Liu, Ming Liu, *Senior Member, IEEE*, and Lujia Wang

***Abstract*—Indoor localization of high accuracy has been widely interested. Among competitive solutions, visible light positioning (VLP) is promising due to its ability to deliver high-accuracy 3D position and orientation with low-cost sensors by sharing the LED lighting infrastructure widespread in buildings. Most VLP systems require a prior LED location map for which manual surveys are costly in practical deployment at scale. In this paper, to address this difficulty, we propose a novel system for efficient and accurate offline mapping of LEDs for VLP. With input from visual-inertial sensors and existing or surveyed priors, it builds the map by posing a full-SLAM (simultaneous localization and mapping) problem within a factor graph formulation. Compared to manual surveys, it greatly saves human labor and time while yielding an accurate and workspace-aligned LED map. With real-world experiments in a room-scale testbed and a 15x larger lab office, we extensively evaluate the LED mapping system to verify its efficacy and performance gains.**

***Index Terms*—Factor graph optimization, indoor localization, LED mapping, visible light communication (VLC), visible light positioning (VLP), visual-inertial odometry (VIO).**

I. Introduction

INDOOR localization is needed by many moving platforms indoors, e.g., for robot navigation and a wide variety of location-based services on mobile devices like people way-finding in GPS-denied venues. With the growing adoption of LEDs for energy-efficient lighting in buildings and the advance of visible light communication (VLC), LED lights hold great potential to be a kind of GPS-like ubiquitous infrastructure that allows accurate and efficient indoor localization [1]–[5]. This approach is known as visible light positioning (VLP). Compared to other infrastructure-based approaches of similar high accuracy (cm to dm) that use ultra-wideband (UWB) radio [6] and ultrasound [7], VLP has the advantage of reusing LED lights as infrastructure. This avoids the extra burden and cost of installing specialized positioning hardware.

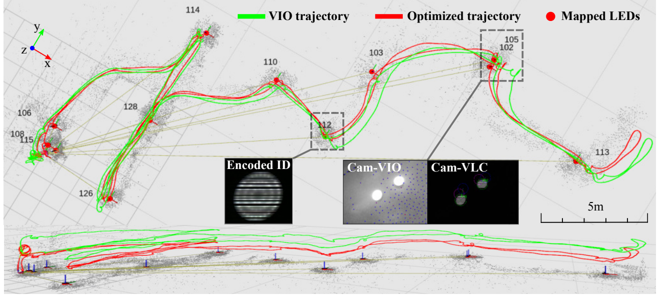


Fig. 1: Results of LedMapper in a lab office (around $20\,\mathrm{m} \times 15\,\mathrm{m}$). It shows the mapped locations of 12 LEDs alongside their IDs decoded through VLC. Also compared are the optimized IMU trajectory after closing loops and the raw VIO input. The example images are from the VIO camera with auto-exposure (bright background) and the VLC camera with a very short exposure (dark background).

In VLC, information is transmitted from modulated LEDs that change intensities quickly beyond human perception and is received by a photodiode (PD) or camera sensor. It carries a unique identifier for each LED, e.g., an identity code (ID) [8]–[12] and a frequency [13]–[16]. In VLP, LEDs act as artificial beacons in the environment, and each allows known data association using its identifier. The sensor takes measurements of LEDs in the VLC range, e.g., bearing, ranging, and received signal strength (RSS). These can be used to infer the senor pose (or position) standalone, such as by geometry-based [15] (e.g., trilateration) and fingerprinting [16] methods, or be combined with other sensors for a fused pose estimate [8]–[12]. To achieve this, most VLP systems require a prior map composed of global LED locations in the environment alongside their identifiers for data association.

If LEDs reside on a precisely assembled frame, the map can be known from the frame geometry and be otherwise by manual surveys using measuring devices (e.g., laser rangefinder, total station, and motion capture system). This can be straightforward for small-scale experiments and is the de facto way in many studies. However, for real applications at scale, manual surveys are difficult and are prone to human errors due to the increase of LEDs and coverage. It entails intensive human labor and time, thereby posing non-negligible deployment costs of VLP systems. To overcome such practical challenges, we seek novel solutions to efficient, accurate LED mapping with less human effort. Yet, this is rarely studied in the VLP field [14]. Our main focus is hence in this direction.

Today, a mobile device has rich sensors onboard, e.g., a MEMS (micro-electro-mechanical system) inertial measurement unit (IMU) and multiple rolling-shutter (RS) cameras. In

*This work was supported in part by the National Natural Science Foundation of China under Grant U1713211, in part by Zhongshan Municipal Science and Technology Bureau Fund under Project ZSST21EG06, in part by the Collaborative Research Fund by Research Grants Council Hong Kong under Project C4063-18G, in part by the Department of Science and Technology of Guangdong Province Fund under Project GDST20EG54, and in part by the Foshan-HKUST Projects University-Enterprise Collaborative R&D Program under Project FSUST20-SHCIRI06C, awarded to Prof. Ming Liu. (*Corresponding author: Ming Liu.*)

Qing Liang, Ming Liu, and Lujia Wang are with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong (email: qliangah@ust.hk; eelium@ust.hk; eewanglj@ust.hk).

Yuxiang Sun is with the Department of Mechanical Engineering, The Hong Kong Polytechnic University, Hung Hom, Hong Kong (e-mail: yx.sun@polyu.edu.hk, sun.yuxiang@outlook.com).

Chengju Liu is with the School of Electronics and Information Engineering, Tongji University, Shanghai, China (e-mail: liuchengju@tongji.edu.cn).

VLP, the RS camera is a ready-to-use receiver, and the IMU is widely used for aided VLP, e.g., in loosely or tightly coupled manners [8]–[11]. Moreover, with camera-IMU sensors, many visual-inertial odometry (VIO) algorithms are well-developed [17]–[19] in robotics, capable of low-drift, accurate 6-degrees-of-freedom (DoF) pose estimates in a local frame. Indeed, VIO has been integrated into recent mobile devices as mature software, e.g., ARKit[1] and ARCore[2]. In applications, it acts as a virtual 6-DoF odometer of low drift (e.g., a few percent of traveled distance or less). To the best of our knowledge, it has not yet been well explored in VLP usages.

In this paper, to tackle the difficulty in LED mapping for VLP, we propose LedMapper, a novel system developed for efficient and accurate offline mapping of modulated LEDs in a 3D workspace, leading to much-reduced human effort than manual surveys. We utilize a rig of low-cost visual-inertial (VI) sensors already existing on mobile devices, i.e., an IMU and two RS cameras (one for VLC and the other for VIO), as the mapping device. The mapping process entails data acquisition in the workspace, for which a surveyor wanders around with the handheld device to form looped paths and points the VLC camera to LEDs when passing by. For VLP, the LED map should align with the global reference frame of the workspace. To allow this, the mapper needs a few known global LED locations (so-called anchors or control points from surveys beforehand) as prior input. Nevertheless, the required human effort is much less than a complete manual survey since the control points account for only a minor portion of all LEDs. The mapping task is solved by posing a full-SLAM (simultaneous localization and mapping) problem within a factor graph formulation [20]. In this work, we assume point-source LEDs but explore LED geometry priors for mapping with the benefit of absolute metric scale. We consider that our contributions are mainly in the VLP field and credit the novelty to the proposed LED mapping system itself. We highlight the novel contributions below:

- A novel LED mapping system for efficient and accurate mapping of LEDs offline. With input from visual-inertial sensors and existing or surveyed priors, it builds the map by solving a full-SLAM problem within a factor graph. Compared to manual surveys, it effectively saves human labor and time while yielding an accurate and workspace-aligned LED map for a wide range of VLP systems.
- Extensive evaluations with real-world experiments in a room-scale testbed and a 15x larger lab office. The results show the efficacy and performance gains of our system.

The remainder of this paper is structured as follows. Section II lists the related work. Section III overviews the proposed system. Section IV briefly reviews the VLC front-end. Section V details the mapping approach. Section VI and Section VII shows experimental results and our discussions of limitations, respectively. Section VIII concludes this paper.

II. Related Work

There is a rich body of literature on VLP, among which [1]–[5] provide fundamentals and comprehensive surveys. To our knowledge, however, only a handful of works aim to map LED locations efficiently for VLP. In this section, we review these closely related to our proposed LedMapper in detail.

In [13] and its follow-up [14], a VLP calibration (i.e., LED mapping) method is proposed using a mobile robot equipped with a 2D Lidar and an upward-facing RS camera. The Lidar data is processed by a SLAM algorithm [21] to give drift-less robot poses. The camera takes images of overhead LEDs and decodes identifiers. The robot must approach each LED until it appears in the image center. One can thereby obtain the 2D LED position from the robot pose. Assuming a known height, this yields an LED map with 3D locations and unique identifiers. Evaluated in a small testbed with four LEDs, [14] shows a good map accuracy of centimeters. Yet, the map is expressed in a local SLAM frame, not necessarily aligned with the global workspace [14]. In addition, data collection may be problematic in a non-traversable area by a wheeled robot.

A handheld device has better mobility in complex scenes. Using a PD receiver and a Tango tablet (running VIO), [22] proposes a light registration method to map 2D light locations onto a floor plan (say height is known). This entails a surveyor who holds the device, starts from a known pose (set by a few anchored lights of known locations), and walks across ceiling lights. Upon crossing a lamp, its identifier is decoded, and its 2D location is recorded using the tracked VIO pose. To bound VIO drifts, [22] divides the mapping area into smaller sections and repeats the process. The VIO paths and light locations are manually aligned to the floor plan. It attains successful results in large-scale scenarios. However, the final map accuracy is not clarified due to the lack of evaluation. Like [14], it may face data collection problems when lights are located above an area not traversable by a person. The required human intervention (e.g., manual alignment) can be prone to errors.

In [22], unmodified lights are in place of modulated LEDs for VLP. Please refer to [1] for a review of VLP based on modified and unmodified lights. As in [14], we focus on mapping modulated LEDs and use an RS camera as the VLC receiver, but for flexible data collection in complex venues, we use a handheld device like in [22]. Likewise, we consider offline mapping since LED positions are fixed and only require a one-time registration. The methods of [13], [14], [22] are heuristic and ignore the uncertainties of sensor measurements and prior knowledge. By contrast, our LedMapper follows a principled design based on probabilistic state estimation [23]. The mapping task is formulated as a full-SLAM problem within a factor graph that has been well studied in robotics [20]. The input information can be utilized in a sounder way. Moreover, we expect our system to be way self-contained, not relying on an accurate floor plan or the presence of other SLAM/localization systems in the environment.

Rather than using an LED location map, some VLP methods (e.g., fingerprinting [16]) require a detailed map of location-labeled light fingerprints. A robot or mobile device can assist in the mapping process, as in [24], [25]. Yet, a detailed review

[1] https://developer.apple.com/augmented-reality/
[2] https://developers.google.com/ar/

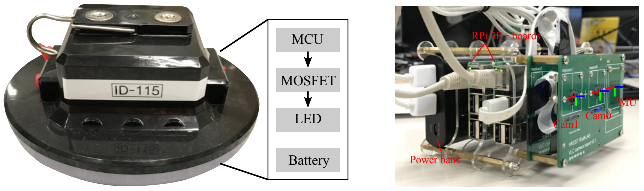
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIM.2021.3123293, IEEE Transactions on Instrumentation and Measurement

of such methods is beyond our scope. Overall, unlike in WiFi-based positioning systems [26], fingerprinting methods attract less attention in VLP studies [3], partly due to the high cost of light fingerprinting [4]. In contrast, due to the line-of-sight (LOS) light propagation, geometry methods allow for high accuracy using LED location maps that are easier to obtain.

III. System Overview

In this section, we describe the hardware setup for our LED mapping task in Section III-A, give an overview of the system workflow in Section III-B, and clarify the notations and reference frames used in this paper in Section III-C.

A. Hardware Setup



(a) (b)

Fig. 2: Photos of our self-assembled hardware. (a) Modulated LEDs. (b) Visual-inertial sensors.

The VLP hardware often consists of modulated LEDs at the infrastructure side and VLC receivers at the user side. Here, an RS camera is chosen as the receiver. We build a few LED prototypes with rechargeable batteries (see Fig. 2a). Each has a round radiation surface of $15.5\,\mathrm{cm}$ in diameter and rating power of $3\,\mathrm{W}$. The portable design allows for a flexible LED deployment for experiments. The LEDs are modified from off-the-shelf products by adding a microcontroller unit (MCU) and reusing most inbuilt components, such as MOSFET, LED beads, batteries, and housings. The MCU runs VLC logic and modulates LEDs via the MOSFET to broadcast LED IDs. Due to the small amount of data required for LED IDs in VLP, a simple, low-speed, and low-cost VLC method is advocated [5]. We follow the same VLC protocol as in our previous work [11], which uses the basic on-off keying (OOK) modulation and Manchester coding scheme due to ease of implementation and DC balance. The OOK modulation frequency is 16 kHz.

The system is evaluated on but not limited to a custom-built VI sensor rig, including two RS cameras (Raspberry Pi camera v2[3]) and a low-cost MEMS IMU (LPMS-ME1[4]). The installation relationship of the two cameras (Cam0&Cam1) and the IMU is illustrated in Fig. 2b. Note the reference frame attached to each sensor is marked by colored axes. The sensors lack hardware synchronization due to hardware limitations. The two cameras are placed in pairs, but not in a stereo setup since they are triggered independently and exposed differently on purpose. This setup is to ease assembly efforts. The rig uses two Raspberry Pi (RPi) 3B+ computers for sensor interfacing. Each RPi has only one CSI (camera serial interface) port that allows connection to the RPi camera. It runs the Ubuntu Mate 16.04 OS with ROS (robot operating system) middleware. The two RPis interconnect by wired Ethernet and communicate through the ROS network. Their system clocks are software synchronized using NTP (network time protocol). The master RPi acts as a local NTP server and provides the clock reference for all sensor timestamps. The IMU connects to the USB port of the master. The sensor streams are recorded as ROS bags for later processing. All hardware is powered by a power bank.

Cam0 captures images with auto exposure for VIO use at $20\,\mathrm{Hz}$ with a resolution of 640×480. Cam1 works with a very short exposure (e.g., $20\,\mu\mathrm{s}$) for VLC use, collecting images at $10\,\mathrm{Hz}$ with a resolution of 1640×1232. IMU is configured to output data at $400\,\mathrm{Hz}$. We assume the sensor rig is pre-calibrated[5] (e.g., using Kalibr [27]–[29]) with known camera intrinsics, IMU intrinsics, IMU-camera spatiotemporal extrinsics, and the RS frame readout time t_r.

A sufficiently large LED image with a complete data packet is required for VLC decoding. The distance from LED to the camera (Cam1) must be close enough. As shown in [11], the maximum decoding distance d_m is subject to the LED's modulation period and surface size, the camera's row readout time and focal length, and the data packet length in the VLC protocol. Due to our hardware limitations (e.g., small-sized LEDs), we trade the reduction of data packet length for an acceptable maximum decoding distance (e.g., a few meters). This is achieved by sacrificing the payload size and omitting error checking in the protocol [11]. Finally, our VLC implementation gives $d_\mathrm{m} \approx 2.5\,\mathrm{m}$ while allowing for a data payload of one byte. We find it sufficient for this study.

B. System Workflow

Fig. 3 shows the workflow of the proposed LED mapping system. It entails three blocks: VLC front-end, VIO estimator, and LED mapper. The VLC front-end takes the RS images from Cam1 as input and produces feature tracks of LED blobs with IDs by LED detection, tracking, and decoding. The gyroscope measurements are utilized to assist LED tracking. The VIO estimator fuses IMU measurements and the natural visual features from Cam0 and provides 6-DoF VIO poses of the IMU frame. We treat it as a black box and assume a favorable lighting condition for VIO operation. It can be implemented by well-established VIO algorithms such as VINS-Mono [18] and OpenVINS [30]. In our case, we choose VINS-Mono due to its support for RS cameras. Using all historical LED tracks and VIO poses as input, the LED mapper aims to build a globally consistent and accurate LED map by offline batch optimization. Besides sensor inputs, we consider prior map information essential for global workspace alignment or relatively easy to obtain, such as control points, ceiling height, and LED geometry. These priors, if available, can be seamlessly incorporated into the mapper as additional constraints for improved quality. After mapping, the built LED map is assumed accurate (with uncertainties) and can later be used by VLP systems for online localization.

[3] https://www.raspberrypi.org/products/camera-module-v2/

[4] https://lp-research.com/lpms-me1-dk/

[5] Currently, we treat these calibrated parameters as known constants without considering the uncertainties due to possible calibration errors. This gross treatment gives acceptable results in our implementation.

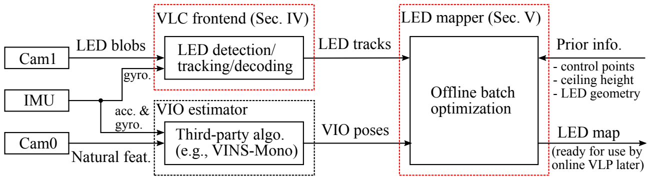


Fig. 3: Block diagram showing the workflow of the proposed system.

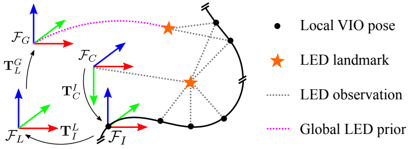


Fig. 4: Illustration of the involved reference frames and their relations. The global LED priors (e.g., control points) are expressed in $\mathcal{F}_G$.

C. Notations and Reference Frames

A transformation matrix $\mathbf{T}_B^A \in SE(3)$ takes a vector $\mathbf{p}^B \in \mathbb{R}^3$ in the frame $\mathcal{F}_B$ to the frame $\mathcal{F}_A$. It can be divided into a rotation matrix $\mathbf{R}_B^A \in SO(3)$ and a translation vector $\mathbf{p}_B^A \in \mathbb{R}^3$, i.e., $\mathbf{T}_B^A = [\mathbf{R}_B^A, \mathbf{p}_B^A]$. The transformed vector $\mathbf{p}^A \in \mathbb{R}^3$ in $\mathcal{F}_A$ is given by $\mathbf{p}^A = \mathbf{R}_B^A\mathbf{p}^B + \mathbf{p}_B^A$. With slight abuse of notation for brevity, we also write $\mathbf{p}^A = \mathbf{T}_B^A\mathbf{p}^B$. Often, we use the unit quaternion under Hamilton convention [31], $\mathbf{q}_B^A$, to represent the rotation. $\mathbf{R}(\cdot)$ converts $\mathbf{q}_B^A$ to the rotation matrix $\mathbf{R}_B^A$. $\otimes$ denotes the quaternion multiplication. For a variable $(\cdot)$, we write its measurement as $\hat{(\cdot)}$. As depicted by Fig. 4, we work with four coordinate frames. The global base frame $\mathcal{F}_G$ is gravity-aligned and fixed in the workspace. It sets the origin for all global measurements. The map to be built is aligned to this frame. The local base frame $\mathcal{F}_L$ is gravity-aligned and sets the origin for VIO pose estimation. $\mathcal{F}_I$ is attached to the IMU body frame, and $\mathcal{F}_C$ is attached to the optical frame of the VLC camera. The IMU-camera extrinsic transformation $\mathbf{T}_C^I$ is a known constant from prior calibration.

IV. VLC Front-end

The blobs from modulated LEDs are detected on incoming images, tracked over consecutive frames, and decoded to have correct LED IDs for long-term data association. The front-end has three modules: blob detection, blob tracking, and VLC decoding. It is mostly inherited from our previous works [10]–[12]. We briefly review it for completeness.

Due to the fast exposure of the VLC camera, bright LEDs have high contrast against the background. After binarization and dilation on grayscale image input, LED blobs are detected with standard techniques. For each blob, we take its centroid with pixel coordinate, $[\hat{u}, \hat{v}]^T$. With the known camera intrinsics, the normalized pixel location, $\hat{\mathbf{z}} \in \mathbb{R}^2$, is also computed.

To achieve tracking, we detect new blobs on every frame and find their best matches [32] in the previous frame. Each blob has a unique tracking ID for short-term data association. We assume the mutual blob distances in an image is greater than the inter-frame pixel displacements. This works in our case due to the sparsity of lights and the non-rapid camera motion. Since camera rotation is more likely to yield large pixel displacements, we compensate for it using the short-term integration of gyroscope measurements before matching.

VLC decoding applies to blobs with barcode-like patterns. The row-parallel strips of varying widths and pixel intensities carry VLC information, e.g., LED ID. Given a blob, we pick up the grayscale pixel values on the centering column of its image region. As the camera's sampling rate is known, these ordered pixel values form a time-varying 1D signal. After binarization, OOK demodulation, and Manchester decoding, the LED ID can be obtained. The blobs that are part of a track have the same LED ID, and we can identify them all if any single is decodable. The blob tracking allows more instances of LED detections for later mapping and VLP.

V. LED Mapping from Batch Optimization

We present the proposed LED mapper based on factor graph optimization in detail, including problem statement in Section V-A, graph construction in Section V-B, factor description in Section V-C, and batch optimization in Section V-D.

A. Problem Statement

The mapping task entails solving a full-SLAM problem that seeks to estimate the entire IMU poses, $\{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$, and the locations of LED landmarks, $\{\mathbf{l}_1, \cdots, \mathbf{l}_M\}$, given all historical sensor measurements and prior knowledge about LEDs. N and M are the total number of poses and landmarks, respectively. It is formulated as a factor graph optimization problem [20]. The structure of the built factor graph is shown in Fig. 5.

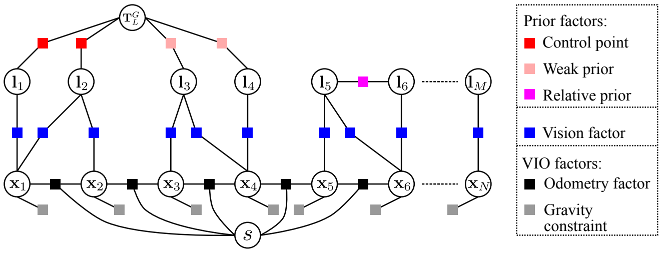


Fig. 5: The factor graph representation of our LED mapping task.

The IMU poses and LED locations are expressed in the local frame $\mathcal{F}_L$ and have the parametrization of $\mathbf{x}_i = \mathbf{T}_{I_i}^L = [\mathbf{p}_{I_i}^L, \mathbf{q}_{I_i}^L], i \in \{1, \cdots, N\}$ and $\mathbf{l}_j = \mathbf{p}_j^L, j \in \{1, \cdots, M\}$, respectively. To align the LED map globally to the workspace, we also estimate the transformation $\mathbf{T}_L^G$. Since $\mathcal{F}_G$ and $\mathcal{F}_L$ are both gravity-aligned, it has 4-DoF with roll and pitch as zeros.

VIO offers 6-DoF odometry measurements, $\hat{\mathbf{T}}_{I_i}^L$, at discrete time instances t_i after initialization. The roll and pitch are accurate, while the yaw and position drift over time. Monocular VIO can suffer from inaccurate metric scale estimates under degenerated motions [33]. Also, low-cost IMUs often have non-negligible systematic errors such as non-unit scale factors and axis misalignment [34]. Without proper compensation by the estimator, these can further affect the VIO scale accuracy. Note that VIO acts as a black box in this work. Dealing with the scale issue from the VIO perspective is beyond our scope. Instead, we estimate in the factor graph the VIO scale factor, s, using additional metric scale sources from priors.

We write the entire state for LED mapping as follows:

$$\mathcal{X}_m = [\mathbf{T}_{I_1}^L, \cdots, \mathbf{T}_{I_N}^L \,|\, \mathbf{p}_1^L, \cdots, \mathbf{p}_M^L \,|\, \mathbf{T}_L^G, s], \qquad (1)$$

where $\{\mathbf{T}_{I_1}^{L}, \cdots, \mathbf{T}_{I_N}^{L}\}$ are the N IMU poses; $\{\mathbf{p}_1^L, \cdots, \mathbf{p}_M^L\}$ are the M LED positions; $\mathbf{T}_L^G$ is the base frame transformation between $\mathcal{F}_G$ and $\mathcal{F}_L$; and s is the VIO scale factor.

B. Graph Construction

We explain how to construct this graph. A new pose node, $\mathbf{x}_i$, is added as the latest VIO pose measurement, $\hat{\mathbf{T}}_{I_i}^L$, arrives (at 10 Hz in our case). This VIO pose gives the initial estimate of $\mathbf{x}_i$. Also, we can apply a simple keyframe selection strategy based on translation and rotation changes to further sparsify the graph. The LED detections are available at maximally 10 Hz from the VLC front-end. Yet, in general, we do not assume the same update rate for the VIO poses and LED detections. When a new LED blob first arrives (as per the unique tracking ID), we create a temporary landmark node for it. As each LED has been tracked over frames, its subsequent detections are associated with this landmark node. After the LED track is complete, we check all the detections to find whether one or more blobs have a valid LED ID (say successfully decoded). If not, the landmark node is immediately dropped due to decoding failure. Note this ID is critical for long-term data association when loop-closure occurs later on. Otherwise, we proceed to check if a landmark of the same LED ID exists in the graph. If so, we re-associate all related LED detections to the existing node and drop the temporal one. Otherwise, we will triangulate the LED position using the involved LED detections and IMU poses alongside the known $\mathbf{T}_C^I$. After this, the landmark node is marked matured and added to the graph as $\mathbf{l}_j$, where j counts the number of unique LED IDs obtained so far. The associated LED ID is denoted as ID_j.

In general, the timestamps of LED detections are not aligned with those of VIO poses, due to the lack of hardware synchronization between the VLC and VIO sensors. We assume the VIO time is based on the IMU clock as in VINS-Mono. Like in [35], we match the LED detections to a past VIO pose with the closest timestamp and point them virtually to the related pose node in the graph. The true pose from which the LED is detected is in between the two bounding IMU poses, which can be obtained by linear interpolation. To facilitate this, we expect the sensors to move smoothly, as in normal walking.

C. Factor Description

As illustrated in Fig. 5, the graph incorporates three sources of factors posed by sensor measurements and prior knowledge. We will describe them in detail below.

1) VIO factors: Given input VIO poses, $\{\hat{\mathbf{T}}_{I_i}^L\}$, the relative transformation measurement between consecutive poses, $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$, is given by $\hat{\mathbf{T}}_{I_{i+1}}^{I_i} = \hat{\mathbf{T}}_{I_i}^{L^{-1}} \hat{\mathbf{T}}_{I_{i+1}}^{L}$. It is written as $[\hat{\mathbf{p}}_{i+1}^i, \hat{\mathbf{q}}_{i+1}^i]$. The predicted motion is $\mathbf{T}_{I_{i+1}}^{I_i} = \mathbf{T}_{I_i}^{L^{-1}} \mathbf{T}_{I_{i+1}}^{L} = [\mathbf{p}_{i+1}^i, \mathbf{q}_{i+1}^i]$. This yields the odometry residue:

$$\mathbf{r}_i^{\mathcal{O}}(\mathbf{x}_i, \mathbf{x}_{i+1}, s) = \begin{bmatrix} s\mathbf{p}_{i+1}^i - \hat{\mathbf{p}}_{i+1}^i \\ 2 \cdot \mathrm{vec3}(\mathbf{q}_{i+1}^i \otimes \hat{\mathbf{q}}_{i+1}^{i^{-1}}) \end{bmatrix}, \tag{2}$$

where s is the VIO scale factor, and $\mathrm{vec3}(\cdot)$ returns the vector part $(q_x, q_y, q_z)^T$ of a quaternion $\mathbf{q}$. We write the covariance of this relative measurement as $\mathbf{\Sigma}_i^{\mathcal{O}} = \mathrm{diag}(\sigma_{pos}^2 \mathbf{I}_3, \sigma_{rot}^2 \mathbf{I}_3)$, where σ_{pos}^2 and σ_{rot}^2 describe the uncertainties in translation and rotation, respectively, and are set empirically in this work.

The roll and pitch in VIO are accurate without drift in $\mathcal{F}_L$. To ensure this property, we exploit them as absolute measurements. For each pose $\mathbf{x}_i$, we apply a rotational constraint due to gravity with the residue:

$$\mathbf{r}_i^{\mathcal{G}}(\mathbf{x}_i) = 2 \cdot \mathrm{vec2}(\mathbf{q}_{I_i}^L \otimes \hat{\mathbf{q}}_{I_i}^{L^{-1}}), \tag{3}$$

where $\mathrm{vec2}(\cdot)$ returns the vector part $(q_x, q_y)^T$ of $\mathbf{q}$. The covariance is written as $\mathbf{\Sigma}_i^{\mathcal{G}} = \sigma_g^2 \mathbf{I}_2$, where σ_g^2 describes the small uncertainty in the absolute roll and pitch measurements.

2) Vision factor: We assume an undistorted, pinhole RS camera model for the VLC camera. An RS camera captures image rows sequentially at varying times. In the case of general motions, this leads to different camera poses for each row. Following the convention in [34], we assume the image timestamp corresponds to the middle image row. For an image of K rows in total with timestamp t, the sampling time of the kth row away from the middle is $t^k = t + \frac{k}{K} t_r, k \in (-\frac{K}{2}, \frac{K}{2}]$. t_r is the RS frame readout time which is assumed known from pre-calibration or the sensor's datasheet if available.

Consider the landmark $\mathbf{l}_j$ associated with the pose $\mathbf{x}_i$ in the graph, for which the VLC front-end gives the normalized image measurement $\hat{\mathbf{z}}_{ij}$ with timestamp t_{ij}. The corresponding pixel coordinate $[\hat{u}, \hat{v}]^T$ lies in row $k = \hat{v} - \frac{K}{2}, \hat{v} \in [1, K]$. Due to the time misalignment between VIO poses and LED detections, $\hat{\mathbf{z}}_{ij}$ is indeed taken from an intermediate pose, $\mathbf{x}_{ij}$, between $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$ with timestamps t_i and t_{i+1}, respectively. With RS imaging, the exact timestamp is given by $t_{ij}^k = t_{ij} + \frac{k}{K} t_r$. This is depicted by the diagram in Fig. 6.

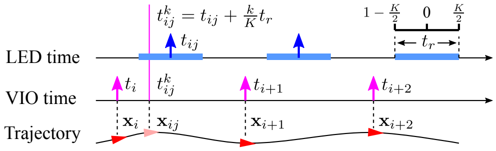


Fig. 6: Illustration of the time misalignment alongside the RS imaging time used for pose interpolation. Note the time interval of VIO poses and that of LED detections are not necessarily the same.

We can obtain $\mathbf{x}_{ij}$ by interpolating $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$, i.e., using spherical linear interpolation (Slerp) for the rotation and linear interpolation for the translation [36]. Having $\mathbf{x}_i = [\mathbf{p}_{I_i}^L, \mathbf{q}_{I_i}^L]$ and $\mathbf{x}_{i+1} = [\mathbf{p}_{I_{i+1}}^L, \mathbf{q}_{I_{i+1}}^L]$, we write $\mathbf{x}_{ij} = [\mathbf{p}_{I_{ij}}^L, \mathbf{q}_{I_{ij}}^L]$ with

$$\mathbf{q}_{I_{ij}}^L = \mathrm{Slerp}(\mathbf{q}_{I_i}^L, \mathbf{q}_{I_{i+1}}^L, \tau) \tag{4}$$

$$\mathbf{p}_{I_{ij}}^L = (1 - \tau)\mathbf{p}_{I_i}^L + \tau \mathbf{p}_{I_{i+1}}^L \tag{5}$$

$$\tau = \frac{t_{ij}^k - t_i}{t_{i+1} - t_i} = \frac{t_{ij} + (\frac{\hat{v}}{K} - \frac{1}{2})t_r - t_i}{t_{i+1} - t_i} \tag{6}$$

where $\mathrm{Slerp}(\mathbf{q}_0, \mathbf{q}_1, \tau)$ is the Slerp function [31] that linearly interpolates from $\mathbf{q}_0$ to $\mathbf{q}_1$ as τ evolves from 0 to 1 ($\tau \in [0, 1]$).

The re-projection residue relating to the landmark $\mathbf{l}_j = \mathbf{p}_j^L$ and the interpolated pose $\mathbf{x}_{ij} = \mathbf{T}_{I_{ij}}^L$ is given by

$$\mathbf{r}_{ij}^{\mathcal{V}}(\mathbf{x}_i, \mathbf{x}_{i+1}, \mathbf{l}_j) = \hat{\mathbf{z}}_{ij} - \pi\left(\mathbf{T}_C^{I^{-1}} \mathbf{T}_{I_{ij}}^{L^{-1}} \mathbf{p}_j^L\right) \tag{7}$$

where $\pi(\cdot)$ projects a 3D point $\mathbf{p}^C$ onto the normalized image plane as a 2D point $\mathbf{z}$ according to $\mathbf{z} = [p_x^C/p_z^C, p_y^C/p_z^C]^T$.

The covariance matrix is given by $\mathbf{\Sigma}_{ij}^{\mathcal{V}} = \sigma_n^2 \mathbf{I}_2$, where σ_n^2 describes the normalized pixel noise for LED observations.

3) Prior factors: We consider prior map information that is deemed essential or readily available to our mapping task. It can be absolute or relative depending on if it is given in $\mathcal{F}_G$. The absolute prior on $\mathbf{l}_i$ is specified with a known LED position $\hat{\mathbf{p}}_i^G$. It provides absolute geometric constraints for the system states. The introduced metric scale information can help correct VIO scale errors. With the transformation $\mathbf{T}_L^G$, this leads to the absolute prior residue below:

$$\mathbf{r}_i^{\mathcal{P}_a}(\mathbf{l}_i, \mathbf{T}_L^G) = \hat{\mathbf{p}}_i^G - \mathbf{T}_L^G \mathbf{p}_i^L \tag{8}$$

with covariance matrix $\mathbf{\Sigma}_i^{\mathcal{P}_a} = \mathrm{diag}(\sigma_x^2, \sigma_y^2, \sigma_z^2)$ that describes the measurement uncertainty on each axis.

For absolute priors, we distinguish between *control points* and *weak priors*. The former is precisely known 3D positions in $\mathcal{F}_G$ for a few selected LEDs. These absolute locations can be obtained by manual surveys using measuring equipment (e.g., laser rangefinder, total station) and are necessary if we need to align the built map with the workspace. The latter is partial knowledge about the LED position (only certain on one axis), e.g., the common height of ceiling LEDs. This can be known from one control point on the ceiling or the 3D architectural plan when available. For control points, we set σ_x^2, σ_y^2, and σ_z^2 to small values. For weak priors with known height, we set small values for σ_z^2 and large values for others.

An initial guess of $\mathbf{T}_L^G$ is needed for its optimization. Given a known pose $\hat{\mathbf{T}}_I^G$ in $\mathcal{F}_G$ and the related VIO pose $\hat{\mathbf{T}}_I^L$, it is given by $\hat{\mathbf{T}}_L^G = \hat{\mathbf{T}}_I^G {\hat{\mathbf{T}}_I^L}^{-1}$. To achieve this, we currently need at least two control points that are closely located. Using such two LEDs co-visible in a single frame, we can compute $\hat{\mathbf{T}}_I^G$ by the 2-point pose initialization [11] based on a closed-form P2P solution [37]. With initialization success, we estimate $\mathbf{T}_L^G$ while zeroing the roll and pitch. Otherwise[6], we fix it to an identity matrix and ignore absolute priors in mapping.

The relative priors come from the known shape and size of LED geometry without effort. Note in this work, we evaluate the system using small-sized circular LEDs and assume point landmarks. In reality, squared panels or linear tubes are often used, and each can be, e.g., represented by a set of corner (end) points. Like in square fiducial markers [38], the side length of LED panels (tubes) provides extra distance measurements of metric scale. Consider a relative prior between two corners on an LED landmark, $\mathbf{l}_i$ and $\mathbf{l}_j$, with the known distance $\hat{d}_{ij}$. The residue is simply written as

$$\mathbf{r}_{ij}^{\mathcal{P}_r}(\mathbf{l}_i, \mathbf{l}_j) = \hat{d}_{ij} - \|\mathbf{p}_i^L - \mathbf{p}_j^L\| \tag{9}$$

with covariance $\mathbf{\Sigma}_{ij}^{\mathcal{P}_r} = [\sigma_d^2]$, where σ_d^2 is the noise uncertainty.

D. Batch Optimization

To obtain the maximum-a-posteriori estimate for the entire state $\mathcal{X}_m$, we minimize a cost function $f(\mathcal{X}_m)$ that sums over the Mahalanobis norm of all measurement residues as follows:

$$\begin{aligned} f(\mathcal{X}_m) = & \sum_{i\in\mathcal{O}} \|\mathbf{r}_i^{\mathcal{O}}(\mathcal{X}_m)\|^2_{\mathbf{\Sigma}_i^{\mathcal{O}}} + \sum_{i\in\mathcal{G}} \|\mathbf{r}_i^{\mathcal{G}}(\mathcal{X}_m)\|^2_{\mathbf{\Sigma}_i^{\mathcal{G}}} \\ & + \sum_{(i,j)\in\mathcal{V}} \rho\left(\|\mathbf{r}_{ij}^{\mathcal{V}}(\mathcal{X}_m)\|^2_{\mathbf{\Sigma}_{ij}^{\mathcal{V}}}\right) \\ & + \sum_{i\in\mathcal{P}_a} \|\mathbf{r}_i^{\mathcal{P}_a}(\mathcal{X}_m)\|^2_{\mathbf{\Sigma}_i^{\mathcal{P}_a}} + \sum_{(i,j)\in\mathcal{P}_r} \|\mathbf{r}_{ij}^{\mathcal{P}_r}(\mathcal{X}_m)\|^2_{\mathbf{\Sigma}_{ij}^{\mathcal{P}_r}}, \end{aligned} \tag{10}$$

where $\rho(\cdot)$ is a robust loss function [23] to reduce the effect of LED outliers. $\mathcal{O}$ and $\mathcal{G}$ are sets of the relative odometry measurements and the absolute rotation measurements around the gravity, respectively, derived from VIO input. $\mathcal{V}$ is the set of visual measurements of LEDs that are successfully decoded. $\mathcal{P}_a$ is the set of absolute map priors, including control points and weak priors, while $\mathcal{P}_r$ is the set of relative map priors. The residuals and their covariances are defined in Section V-C.

This nonlinear problem is solved using Ceres solver [39]. After optimization, we obtain the global position of each LED, $\mathbf{l}_j$, as per $\mathbf{p}_j^G = \mathbf{T}_L^G \mathbf{p}_j^L$. The set of $\{(\mathrm{ID}_j, \mathbf{p}_j^G)\}, j \in [1, M]$ constitutes the final LED map anchored to the workspace. For later VLP use, we empirically set a fixed uncertainty, based on the experimental evaluation, for the mapped locations.

Remark: Without map priors, the VIO scale factor cannot be determined and is hence fixed in optimization (i.e., $s = 1$). As such, the optimized LED positions and the IMU poses can be subject to an inaccurate metric scale estimate from the VIO input. If control points are not available or when $\mathbf{T}_L^G$ fails in initialization, as explained previously, the built LED map can not align to $\mathcal{F}_G$ in the workspace. Yet, it is acceptable to use if we allow localization solutions within a local frame $\mathcal{F}_L$.

VI. Evaluation

In this section, we evaluate the proposed LED mapping system by real-world experiments. We first introduce the experiment setup in Section VI-A. We assess the LED mapping accuracy in a controlled testbed with ground truth LED locations in Section VI-B. We show the influence of VIO scale errors on results and describe how they are compensated for by using various priors if available. In Section VI-C, we study the possible impact of LED sparsity on map accuracy. In Section VI-D, we show the influence of the inhomogeneity of LED layout. Finally, in Section VI-E, we evaluate the mapping system at a 15x larger office area of more realistic settings.

A. Experiment Setup

To verify the efficacy of our LedMapper, we compare its four variants: M1–M4. The difference lies in the constraints needed for optimizing the cost function $f(\mathcal{X}_m)$, including sensor measurements $\{\mathcal{O}, \mathcal{G}, \mathcal{V}\}$ and prior knowledge $\{\mathcal{P}_a, \mathcal{P}_r\}$, as explained in Section V-D.

- M1: The baseline method use sensor measurements $\{\mathcal{O}, \mathcal{G}, \mathcal{V}\}$ only without prior. The mapping results can suffer from a non-unit metric scale due to VIO.
- M2: Besides $\{\mathcal{O}, \mathcal{G}, \mathcal{V}\}$ as in M1, it uses absolute priors $\mathcal{P}_a$ from control points. We aim to study their effects

[6] E.g., there are less than two control points; control points are far separated, not allowing for 2-point pose initialization; and fail due to other reasons.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIM.2021.3123293, IEEE Transactions on Instrumentation and Measurement

on scale estimates and mapping accuracy, alongside their usage for global map alignment.

- M3: Like M2, it also exploits $\{\mathcal{O}, \mathcal{G}, \mathcal{V}, \mathcal{P}_a\}$. However, $\mathcal{P}_a$ now includes both control points and weak priors (e.g., ceiling height). Comparing M3 to M2 allows us to see the impact of weak priors.
- M4: Besides $\{\mathcal{O}, \mathcal{G}, \mathcal{V}\}$ as in M1, it uses relative priors $\mathcal{P}_r$ from LED geometry. We aim to test their efficacy in improving VIO scale estimates and mapping accuracy.

For comparison, we align the results to the ground truth by SE(3) or Sim3 transformation using [40]. To assess the map accuracy, we compute the root-mean-square error (RMSE) of estimated LED positions. The metric scale error in LED positions is based on the scale computed during Sim3 alignment. The trajectory accuracy is evaluated by the absolute trajectory error (ATE) [41].

All experimental data are collected using the self-assembled sensor rig (cf. Section III-A) and are processed on a desktop computer (Intel i7-7700K CPU@4.2 GHz, 16 GB RAM).

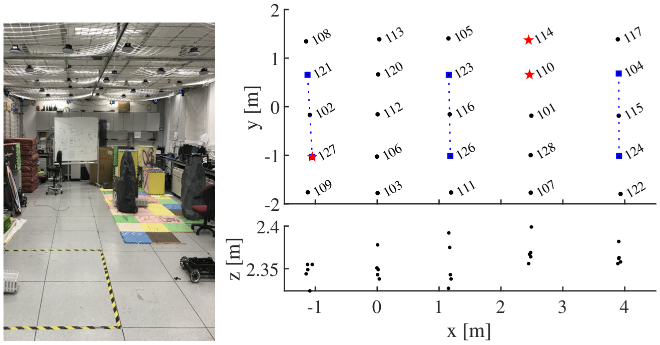
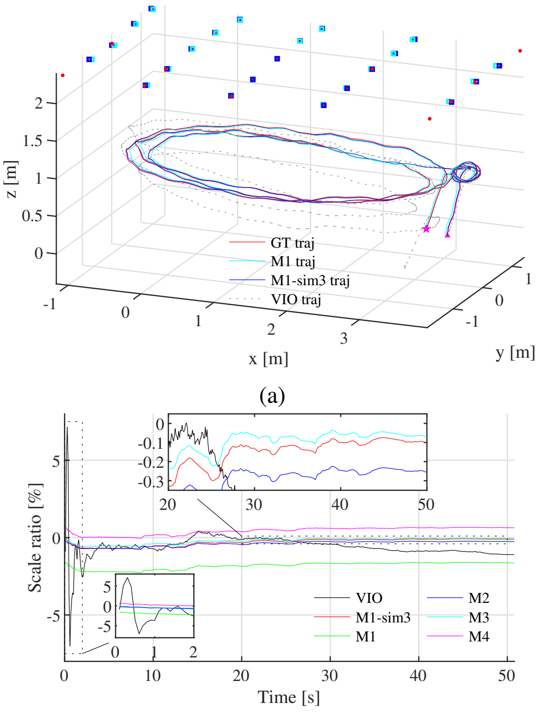


Fig. 7: The photo (left) and illustration (right) of the Mocap testbed for experiments, showing the locations and IDs of mounted LEDs. The black dots show the ground truth locations. The red pentagons are for control points. The blue squares are for LED geometry prior.

B. Mapping Accuracy

To study the mapping accuracy, we conduct experiments in a room-sized testbed ($5\,\mathrm{m} \times 4\,\mathrm{m} \times 2.35\,\mathrm{m}$) instrumented with a precise OptiTrack[7] motion capture system (Mocap). There are 25 modulated LEDs evenly distributed on the ceiling, as shown in Fig. 7. We set $\mathcal{F}_G$ the same as the world frame of Mocap. High-accuracy 3D LED positions are available as ground truth by a tedious manual calibration procedure using Mocap and a laser rangefinder for height compensation[8]. The ground truth IMU poses (trajectory) are computed from offline batch optimization, using all available sensor inputs (VIO poses and LED detections) and taking all ground truth LED positions as control points[9]. To collect data in the testbed, we carry the handheld sensor, point the VLC camera to ceiling LEDs and walk around normally to close loops. While walking, we do not require the camera to face upright to the ceiling (say it can tilt forward). The sensor height is kept relatively constant ($1\,\mathrm{m}$ above the floor) in the experiment. Five datasets with different motion profiles are collected, and each lasts about one minute[10]. At the start of each run, the sensor is put on the ground still for a few seconds and then moved with sufficient motion excitation to aid VIO initialization. The start point sits beneath LED-110 and LED-114, as shown in Fig.7.

We now detail the test settings for the four mapper variants (i.e., M1–M4). As indicated in Fig.7, three LEDs of known positions in $\mathcal{F}_G$ (red pentagons) are chosen as control points for M2/M3. With LED-110 and LED-114, an initial estimate for $\mathbf{T}_L^G$ can be obtained by 2-point pose initialization [11]. We use the rough ceiling height of $2.35\,\mathrm{m}$ as weak priors for M3. The affected standard deviation is set as $\sigma_z = 0.2\,\mathrm{m}$. For M4, we select three pairs of LEDs (blue squares) with known pairwise distances (dashed blue lines) and treat this knowledge as a simulated source of relative priors from LED geometry.



(a)

(b)

Fig. 8: Mapping results on dataset #1 using four methods (M1–M4). (a) visualizes the mapped LED positions and the optimized trajectory. Cyan shows results after SE(3) alignment, blue shows results after Sim3 alignment, and red shows the ground truth. (b) plots the scale ratio of optimized trajectories and VIO input over time.

Fig.8 shows the results on dataset #1 of M1–M4 after SE(3) alignment. 22 LEDs are mapped successfully. The Sim3 results of M1 are also included (denoted as M1-sim3). We align the VIO trajectory using its first 50 poses. Fig.8a compares M1's results with the ground truth and VIO input. The results of

[7] https://optitrack.com/

[8] The ceiling LED locations are beyond the operation scope of our Mocap system. We do calibration in two steps using Mocap and a leveler-mounted laser rangefinder. We first measure the 3D orthogonal projection position of the LED on the floor using Mocap and then obtain the truth LED location by compensating for the height difference using the laser distance measurement.

[9] The ground truth trajectory data provided by Mocap were not recorded at the time of data collection due to some reason. Yet, this does not hamper the goal of assessing the LED mapping accuracy in this work.

[10] Running on these datasets, batch optimization for mapping 25 LEDs can finish within a fraction of seconds.

TABLE I: Mapping results by M1, M2, M3, and M4 on five datasets. Each column reports the metric scale error (in %) and the LED position RMSE (in cm) after Sim3 and SE(3) alignment. The bold figures highlight the best results (i.e., smallest errors) in each row.

#	M1 [%, cm, cm]			M2 [%, cm, cm]			M3 [%, cm, cm]			M4 [%, cm, cm]		
1	1.61	**1.5**	3.4	0.26	1.6	1.6	**0.07**	**1.5**	**1.5**	0.59	**1.5**	1.9
2	2.10	2.4	5.0	0.21	2.3	2.3	**0.14**	**2.0**	**2.0**	0.49	2.3	2.6
3	3.49	2.2	7.6	0.20	2.1	2.1	0.12	**1.9**	**1.9**	**0.08**	2.2	2.2
4	2.91	2.6	6.7	0.15	2.1	2.2	**0.14**	**2.0**	**2.0**	0.84	2.6	3.2
5	2.72	1.5	5.7	0.28	1.5	1.6	**0.17**	**1.4**	**1.5**	0.18	1.5	**1.5**

TABLE II: Map alignment errors by M2 and M3 on five datasets. The best results are shown in bold.

#	M2 [cm, deg]		M3 [cm, deg]	
1	2.5	**0.44**	**2.2**	0.53
2	**0.7**	0.36	1.0	**0.27**
3	3.4	0.85	**1.0**	**0.23**
4	3.6	0.72	**1.4**	**0.25**
5	1.4	0.31	**1.0**	**0.21**

M2–M4 are very close to M1-sim3 and are omitted here for clarity. The Sim3 results (blue) well fit the ground truth, while the SE(3) results (cyan) show larger mismatches (see outer rings). This suggests an inaccurate scale estimate in M1. To examine the scale errors in optimized trajectories, we compare the scale ratio [33] among different methods, as shown in Fig.8b. The scale ratio is computed as the traveled distance of the trajectory estimate divided by the ground truth and subtracted by one. A ratio closer to zero means a better metric scale. For M1 and VIO, the scale ratio drifts away evidently from zero. This confirms the non-unit metric scale in the VIO input, which, without correction, will later translate into scale errors in M1. For M2, M3, and M1-sim3, the scale ratio is close to zero (the absolute value is less than $0.3\,\%$). Also, M4 has an improved scale estimate than M1, despite remaining drifts. As a result, the priors used by M2–M4 can help correct VIO scale errors due to their absolute scale information.

The quantitative results by M1–M4 running on five datasets are reported in Table I. In the column of each method, from left to right, we present the scale errors in LED positions, the RMSE of LED positions after Sim3 alignment, and that after SE(3) alignment. The total number of mapped LEDs on five datasets is among $\{22, 23, 24, 22, 25\}$. Overall, M1 yields more significant errors in scale estimates and the SE(3)-aligned LED positions, while M3 achieves the least errors. With Sim3 alignment, however, the map accuracy of M1 is very close to that of M3 (about $2\,\mathrm{cm}$), and no big difference appears among M1–M4. That is, M1 can yield decent mapping results, despite a non-unit metric scale (e.g., a few percent of errors). We can thereby say that the majority of M1's mapping errors are due to its inaccurate metric scale and, in our case, is from the VIO input. If not fixed, it can undermine the mapping accuracy.

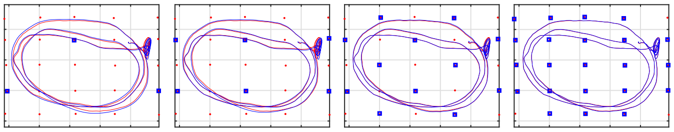
Compared to M1, the metric scale errors of M2 and M3 are of a few thousandths, reduced by one order of magnitude; and despite being less remarkable, that of M4 is three times smaller. For M2–M4, we observe a reduction in LED position errors by a factor of 2 to 3. The LED position RMSE is almost within $3\,\mathrm{cm}$ across five datasets. This shows the advantage of using priors over the baseline method. Due to added geometric constraints, the priors from a few control points (e.g., M2/M3) or the LED geometry (e.g., M4) can help correct the scale estimate and maintain a good map accuracy. To see how weak priors contribute, we compare M3 to M2 and find that M3 has smaller errors in both the estimated scale and LED positions. Note M3 takes the ceiling height as extra priors while M2 does not. This shows the gain for better mapping accuracy of using some weak priors, which are easy to obtain, by our mapper.

As per the design, control points enable the built LED map to be aligned with the global workspace. To evaluate this, we report in Table II the map alignment errors of M2 and M3 on five datasets computed during SE(3) alignment. M2 yields a translation error of within $4\,\mathrm{cm}$ and a rotation error of within one degree. M3 performs even better, with smaller errors in translation ($\leq 2.2\,\mathrm{cm}$) and rotation ($\leq 0.53\,\mathrm{deg}$). This accuracy gain is due to the weak priors about LED height.

C. Impact of LED Sparsity on Mapping Accuracy

In what follows, we study the impact of the sparsity of LED observations on mapping accuracy. The previous setup of 25 LEDs in a $5\,\mathrm{m} \times 4\,\mathrm{m}$ area means a dense LED placement and provides rich LED observations. In reality, due to the variances of LED deployment density and ceiling height in complex indoor settings, LED observations valid for mapping can be much sparser. It will be helpful to assess the mapping performance under sparse LED setups. Explicitly, we consider four LED setups of different sparsity levels with 3/6/12/25 LEDs, as shown in Fig. 9. Rather than altering the physical setup, for evaluation convenience, we selectively use measurements from the chosen LEDs (see blue squares in Fig. 9a-c). These are evenly scattered over the test area.

We test M1 on the five testbed datasets as used previously. To minimize the influence of VIO scale errors, we pre-calibrate the scale factor s by batch optimization using all sensor measurements and the ground truth LED map and keep it fixed in the experiments. That is, we here assume a scale-correct VIO input. As the VIO scale can change with motion profiles across datasets, we calibrate the scale factor individually for each but apply it to all four LED setups. For comparison, the mapping results are SE(3)-aligned to the ground truth.



(a) (b) (c) (d)

Fig. 9: Mapping results (blue) on dataset #1 by M1 under LED setups of varying sparsities, compared to the ground truth (red). From (a) to (d) are with 3, 6, 12, and 25 LEDs. 3 LEDs are not mapped in (d).

In Fig. 9, we show the qualitative results (blue) obtained on dataset #1 for four LED setups, compared to the ground truth (red). We report the quantitative results on five datasets in the boxplots of Fig. 10, including LED position errors and trajectory RMSE. With the four LED setups, the mapper can recover LED positions and the IMU trajectory on all datasets. As shown in Fig. 9, these LED positions match the ground

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIM.2021.3123293, IEEE Transactions on Instrumentation and Measurement

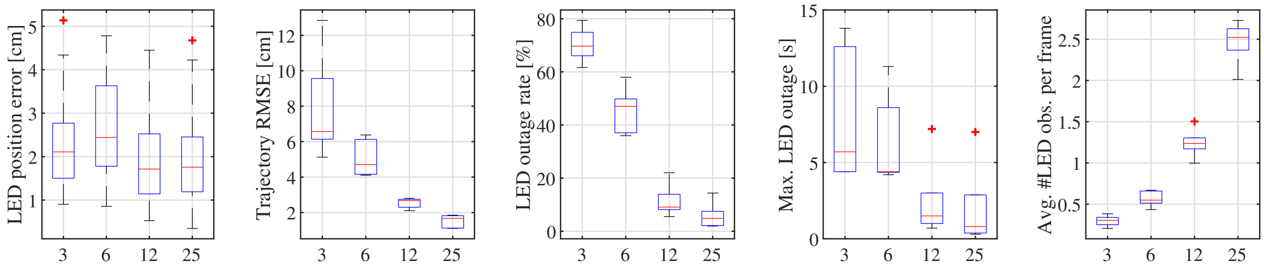


Fig. 10: Mapping results by M1 on five datasets and with 3, 6, 12, and 25 LEDs. From left to right: LED position errors, RMSE of optimized trajectories, LED outage rate, max LED outage, and the average number of LED observations per frame.

truth well, even when LEDs are sparse. The map accuracy is consistent among the four LED setups, as seen from the boxplot of LED position errors ($5\,\mathrm{cm}$@max) in Fig. 10. As a result, the LED mapper can run under very sparse LED distribution and build an accurate LED map. The trajectory errors, as shown by the boxplot in Fig. 10, grow evidently as LEDs become sparser. This degradation is due to insufficient loop-closure constraints from LED observations on IMU motions. To examine the availability of LED observations, we report in Fig. 10 the LED outage rate, the maximum outage, and the average number of observations per frame. The outage rate is computed as the accumulated time of LED outages divided by the total time (the higher this rate, the severer the outage). The maximum outage measures the longest period without LED observations. The results clearly show the lack of LED constraints when LEDs are sparsely placed.

D. Impact of the Inhomogeneity of LED Layout

So far, we consider only those LED setups of homogeneous layout (LEDs are evenly scattered over the area for mapping). Sometimes, LEDs may not be evenly deployed but are clustered on one side of the mapping area. It is helpful to study the impact of inhomogeneous LED layouts on mapping accuracy.

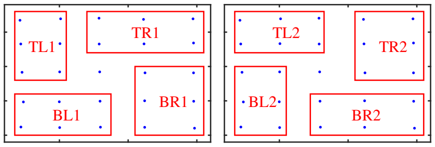


Fig. 11: Eight testbed setups of inhomogeneous LED layouts with 6 LEDs. TL: top left, TR: top right, BL: bottom left, BR: bottom right.

In the experiment, we consider eight setups of inhomogeneous LED layouts with 6 LEDs in the testbed, as shown in Fig. 11. In each setup, all the 6 LEDs for mapping are clustered on one corner of the testbed. For comparison, we take the homogeneous LED setup with 6 LEDs (see Fig. 9b) as a baseline. These LED layouts have different homogeneity but the same sparsity (i.e., the same number of LEDs in a given area). Like in Section VI-C, we test the M1 variant of the LedMapper on the five testbed datasets. Also, we follow the same experimental settings, e.g., fixing the VIO scale by pre-calibration and aligning the results by SE(3).

In Fig. 12, for each dataset, we report the mapping results by M1 under inhomogeneous LED setup (see boxplots) alongside the baseline result under homogeneous LED setup (see triangles). Fig. 13 shows the time-evolving number of LED observations per frame on dataset #1, corresponding to one inhomogeneous and one homogeneous LED setup.

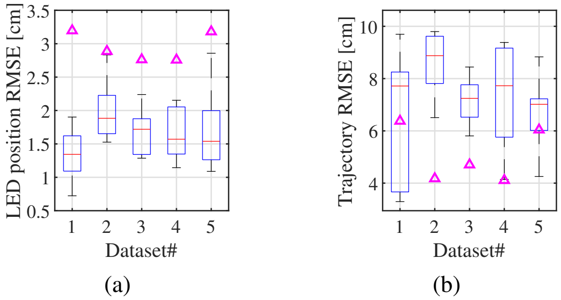


Fig. 12: Mapping results by M1 with 6 LEDs on five datasets: (a) RMSE of LED position estimates and (b) RMSE of trajectory estimates. The boxplots summarize the results under eight inhomogeneous LED layouts. The triangles show the baseline results under the homogeneous LED layout.

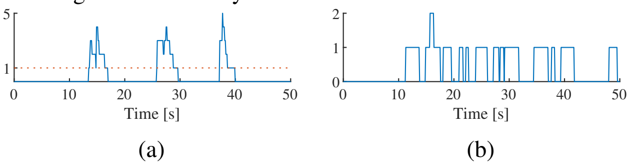


Fig. 13: Results of the time-evolving number of LED observations per frame on dataset #1, corresponding to (a) one inhomogeneous (TL1) and (b) one homogeneous LED setup with 6 LEDs.

As seen from Fig. 12a, the LED position RMSE on the five datasets is smaller than the baseline result. In the experiment, the system achieves improved LED mapping accuracy in the inhomogeneous case. This is likely due to more LED observations in the small clustered area. Under our inhomogeneous LED setups, the LEDs are all within a local region, in which the camera can observe multiple LEDs per frame (see Fig. 13a). This can help yield more consistent and accurate LED position estimates. By contrast, in the homogeneous case, the camera can often observe one LED per frame (see Fig. 13b) due to the scattered distribution.

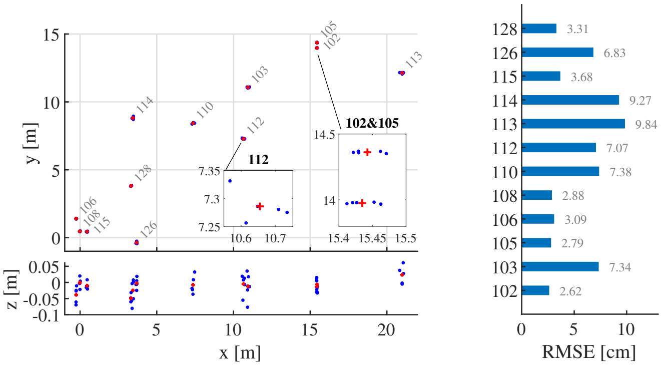
Meanwhile, the accuracy of trajectory estimates tends to degrade under inhomogeneous LED setups, as shown in Fig. 12b. We suspect this is because of the less frequent correction to the drifting VIO poses based on intermittent LED detections (see Fig. 13a). The batch optimized trajectory can serve as the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIM.2021.3123293, IEEE Transactions on Instrumentation and Measurement

best achievable positioning result by a real-time VLP system, which takes as input the VIO estimates and LED detections as used by LED mapping. In this sense, to allow for better VLP accuracy, we prefer homogeneous LED layouts in a given area.

E. Mapping Tests at a Lab Office

Now, we aim to test LedMapper in more realistic settings and at a larger scale. We carry out experiments in our lab at HKUST, a typical office scene that covers about $20\,\mathrm{m} \times 15\,\mathrm{m}$. We place 12 LEDs randomly on the ground along pathways and leave them facing the ceiling. In principle, our system can work from LEDs placed at will since they are no more than 3D landmarks. The choice of putting them on the ground is for evaluation convenience.



(a) (b)

Fig. 14: Results in the lab-scale test on five datasets using 12 LEDs. (a) shows the positions and IDs of mapped LEDs. For each LED, the blue dots are estimates, and the red plus is the mean. (b) shows the statistics of the LED position RMSE among five datasets.

In this experiment, we do not have ground truth for the IMU trajectory and LED positions. To evaluate the mapping performance, we examine the consistency of mapped locations among multiple runs. Five datasets are collected in the lab using the mobile mapping device. During collection, we point the cameras forward and face them to ground LEDs when walking by. We revisit these LEDs to form closed loops before returning to the start point. We test the mapper by the baseline method M1 due to the lack of ground truth measurements of control points or pairwise distances (needed by M2–M4).

In Fig. 1 of Section I, we show the results on a typical lab dataset with the locations and IDs of mapped LEDs and the optimized trajectory, as well as the VIO input. All 12 LEDs are mapped successfully. As expected, the optimized trajectory has less drift (see z-axis) and better consistency. We take this map as a reference and align the results from the other datasets to it by Sim3. This is because M1 is subject to a non-unit VIO scale that can vary among datasets. For each LED, we compute the errors of estimated positions to their mean and take the RMSE as a measure of consistency. The results on five datasets are summarized in Fig. 14. As can be seen from Fig. 14a, the mapped LED locations are consistent among five runs (see the clusters of blue dots); yet, discrepancies are evident in zoomed views. The position RMSE is within $10\,\mathrm{cm}$ for all 12 LEDs, as shown in Fig. 14b. Mapping errors at such a level of degree are acceptable, considering that the lab area is 15x larger than our previous testbed. According to the study in Section VI-B, the mapping performance can be further improved using M2–M4, given any existing or surveyed prior knowledge.

VII. Discussion of Limitations

Currently, we use homemade circular LEDs for evaluation. The blob detector and tracker of the VLC front-end are designed for circular LEDs. Still, the system can run with other shaped LEDs (e.g., linear tubes, square panels) by adapting the front-end. Moreover, pose estimation using a single such LED is achievable, while the appearance (often symmetrical) may need extra modification (e.g., a colored marker on corners) for distinctiveness [42]. For a modified square LED of known size, the camera observation model for square fiducials [38] can be readily applied to our system. Also, due to limited LEDs for the study, we have assessed the system in a room-sized testbed and a $300\,\mathrm{m}^2$ office area. There is a practical difficulty in preparing enough homemade LEDs for experiments at a larger scale. With more LEDs available in future work, evaluation in wider environments will be desired.

The data payload and maximum decoding distance d_m by our VLC implementation could be insufficient in reality. A larger-sized data payload is essential to large-scale deployment with thousands of LEDs. Also, to enable operation in scenarios with high ceilings (e.g., shopping malls rather than our office buildings), a greater d_m is desired. Otherwise, the LED-camera distance can easily exceed d_m, causing decoding failure. As mentioned previously, these limitations are mainly due to the small LED surface (i.e., $15.5\,\mathrm{cm}$) in use. In real applications, one can effectively increase the data payload and (or) the maximum decoding distance by simply using LEDs of a larger surface. As for standard LEDs for daily lighting, a square panel can be $50\,\mathrm{cm}$ wide, while a linear tube can be $120\,\mathrm{cm}$ long. In future work, advanced VLC modulation/coding schemes can be explored for improved performance.

To ease hardware setup, we resort to a low-quality VI sensor without hardware triggering. Under the same environments and motion profiles, it yields inferior VIO performance with more drifts and scale errors. This can affect the best achievable accuracy of the LedMapper. Also, VI measurements are now loosely fused by our system since these are preprocessed by a third-party VIO estimator before use, leading to suboptimal results. For higher mapping accuracy, a high-quality VI sensor alongside a tightly coupled implementation is more advocated.

LedMapper is not fully automated as it still requires manual input, e.g., surveying a few LEDs as control points. Yet, these are often necessary to align the LED map with the workspace for VLP. Even so, the human effort has been much reduced than manual surveys, as control points take only a minor portion of LEDs (e.g., $3/25$ in our case). To align the map, the mapper now needs at least two close control points. We safely expect improved map accuracy if using more. However, this will cause increased human effort and hence less efficiency. In practice, a trade-off should be sought between accuracy and efficiency. When the architectural floor plan has updated LED locations, it can assist as informative priors. Yet, one cannot directly turn it into a usable map for VLP due to the lack of LED identifiers on a standard floor plan.

Compared to photodiode-based VLP systems, camera-based systems are less affected by multipath effects [2], [3] from diffuse reflections off rough surfaces (e.g., walls, floors). In daily scenarios, specular reflective materials (e.g., glass, mirrors) are relatively few [43] but challenging to camera-based systems, alongside our LedMapper. When LEDs are close to a mirror surface, the LED and its mirroring can be observed and decoded by the camera at the same time. While it is not easy to disambiguate between them, we can circumvent this issue by discarding the affected LED detections. In a worse situation, only the LED mirroring is detected. Currently, our LedMapper can not handle this corner case. In practice, specular reflections can be in part reduced by adding a polarizer on the camera lens. To solve this issue, however, much research effort is required in future work.

VIII. Conclusion

This paper introduced a novel system designed for efficient and accurate offline mapping of modulated LEDs for VLP, named LedMapper. Compared to manual surveys, it required much less human effort in building a usable LED map, thereby reducing the deployment costs of VLP systems in reality. A handheld mapping device with low-cost visual-inertial sensors was utilized. The mapping process entailed a surveyor wandering around the workspace with the device for data collection. Given collected sensor data and some existing or surveyed priors, it can build an accurate and workspace-aligned LED map by formulating a full-SLAM problem within a factor graph. Compared to its heuristic counterparts, LedMapper exploited input information in a sounder way, credited to the principled design following probabilistic state estimation. Finally, the system was extensively evaluated with real-world experiments in a room-scale controlled testbed and a 15x larger lab office, showing its efficacy and performance gains.

In future work, we will adapt the system to different-shaped LEDs and evaluate it in larger-scale settings. It is rewarding to do tightly coupled integration for higher mapping accuracy or to explore advanced VLC modulation/coding methods for better performance. Lastly, improving the system robustness to specular reflections is challenging and yet to be solved.

References

[1] A. Rahman, T. Li, and Y. Wang, "Recent advances in indoor localization via visible lights: A survey," *Sensors*, vol. 20, no. 5, p. 1382, 2020.

[2] M. Maheepala, A. Z. Kouzani, and M. A. Joordens, "Light-based indoor positioning systems: A review," *IEEE Sensors J.*, vol. 20, no. 8, pp. 3971–3995, 2020.

[3] M. Afzalan and F. Jazizadeh, "Indoor positioning based on visible light communication: A performance-based survey of real-world prototypes," *ACM Comput. Surveys (CSUR)*, vol. 52, no. 2, pp. 1–36, 2019.

[4] 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.

[5] 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.

[6] A. R. J. Ruiz and F. S. Granja, "Comparing Ubisense, BeSpoon, and DecaWave UWB location systems: Indoor performance analysis," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 8, pp. 2106–2117, 2017.

[7] T. Liu, X. Niu, J. Kuang, S. Cao, L. Zhang, and X. Chen, "Doppler shift mitigation in acoustic positioning based on pedestrian dead reckoning for smartphone," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–11, 2020.

[8] 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.

[9] 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.

[10] Q. Liang, J. Lin, and M. Liu, "Towards robust visible light positioning under LED shortage by visual-inertial fusion," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*. IEEE, pp. 1–8.

[11] Q. Liang and M. Liu, "A tightly coupled VLC-inertial localization system by EKF," *IEEE Rob. Autom. Lett.*, vol. 5, no. 2, pp. 3129–3136, 2020.

[12] Q. Liang, Y. Sun, L. Wang, and M. Liu, "A novel inertial-aided visible light positioning system using modulated LEDs and unmodulated lights as landmarks," *IEEE Trans. Autom. Sci. Eng.*, pp. 1–19, 2021.

[13] R. Amsters, E. Demeester, P. Slaets, D. Holm, J. Joly, and N. Stevens, "Towards automated calibration of visible light positioning systems," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*. IEEE, 2019, pp. 1–8.

[14] R. Amsters, E. Demeester, N. Stevens, and P. Slaets, "Calibration of visible light positioning systems with a mobile robot," *Sensors*, vol. 21, no. 7, p. 2394, 2021.

[15] 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.

[16] A. H. A. Bakar, T. Glass, H. Y. Tee, F. Alam, and M. Legg, "Accurate visible light positioning using multiple-photodiode receiver and machine learning," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–12, 2020.

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

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

[19] B. Fu, F. Han, Y. Wang, Y. Jiao, X. Ding, Q. Tan, L. Chen, M. Wang, and R. Xiong, "High-precision multicamera-assisted camera-IMU calibration: Theory and method," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–17, 2021.

[20] F. Dellaert, M. Kaess *et al.*, "Factor graphs for robot perception," *Foundations and Trends® in Robotics*, vol. 6, no. 1-2, pp. 1–139, 2017.

[21] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D LIDAR SLAM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2016, pp. 1271–1278.

[22] C. Zhang and X. Zhang, "Pulsar: Towards ubiquitous visible light localization," in *Proc. 23rd Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*. ACM, 2017, pp. 208–221.

[23] T. D. Barfoot, *State estimation for robotics*. Cambridge University Press, 2017.

[24] T. Glass, F. Alam, M. Legg, and F. Noble, "Autonomous fingerprinting and large experimental data set for visible light positioning," *Sensors*, vol. 21, no. 9, p. 3256, 2021.

[25] Y. Yue, X. Zhao, and Z. Li, "Enhanced and facilitated indoor positioning by visible-light GraphSLAM technique," *IEEE Internet Things J.*, vol. 8, no. 2, pp. 1183–1196, 2020.

[26] Q. Liang and M. Liu, "An automatic site survey approach for indoor localization using a smartphone," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 1, pp. 191–206, 2019.

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

[28] L. Oth, P. Furgale, L. Kneip, and R. Siegwart, "Rolling shutter camera calibration," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2013, pp. 1360–1367.

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

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

[31] J. Sola, "Quaternion kinematics for the error-state kalman filter," *arXiv preprint arXiv:1711.02508*, 2017.

[32] J. Munkres, "Algorithms for the assignment and transportation problems," *J. Soc. Ind. Appl. Math.*, vol. 5, no. 1, pp. 32–38, 1957.

[33] K. J. Wu, C. X. Guo, G. Georgiou, and S. I. Roumeliotis, "Vins on wheels," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2017, pp. 5155–5162.

[34] M. Li, H. Yu, X. Zheng, and A. I. Mourikis, "High-fidelity sensor modeling and self-calibration in vision-aided inertial navigation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2014, pp. 409–416.

[35] R. Mascaro, L. Teixeira, T. Hinzmann, R. Siegwart, and M. Chli, "GOMSF: Graph-optimization based multi-sensor fusion for robust UAV pose estimation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2018, pp. 1421–1428.

[36] J. Hedborg, P.-E. Forssén, M. Felsberg, and E. Ringaby, "Rolling shutter bundle adjustment," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recog. (CVPR)*. IEEE, 2012, pp. 1434–1441.

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

[38] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2011, pp. 3400–3407.

[39] S. Agarwal, K. Mierle, and Others, "Ceres solver," http://ceres-solver.org.

[40] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. IEEE, 2018, pp. 7244–7251.

[41] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. IEEE, 2012, pp. 573–580.

[42] S. Cincotta, A. Neild, and J. Armstrong, "Luminaire reference points (LRP) in visible light positioning using hybrid imaging-photodiode (HIP) receivers," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*. IEEE, 2019, pp. 1–8.

[43] B. Xie, K. Chen, G. Tan, M. Lu, Y. Liu, J. Wu, and T. He, "LIPS: A light intensity–based positioning system for indoor environments," *ACM Trans. Sens. Netw. (TOSN)*, vol. 12, no. 4, pp. 1–27, 2016.

Qing Liang (Member, IEEE) received the B.A. degree in automation from Xi'an Jiaotong University, Xi'an, China, in 2013, and the master's degree in instrument science and technology from Beihang University, Beijing, China, in 2016. He is currently pursuing the Ph.D. degree with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong.

His current research interests include sensor fusion, low-cost localization, and mobile robots.

He was a recipient of the Best Paper Award at the International Conference on Indoor Positioning and Indoor Navigation (IPIN) 2019 and the ABB Best Student Paper Finalist Award at the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2018.

Yuxiang Sun (Member, IEEE) received the bachelor's degree from the Hefei University of Technology, Hefei, China, in 2009, the master's degree from the University of Science and Technology of China, Hefei, China, in 2012, and the Ph.D. degree from The Chinese University of Hong Kong, Hong Kong, in 2017.

He is currently a Research Assistant Professor at the Department of Mechanical Engineering, The Hong Kong Polytechnic University, Hong Kong. Prior to that, he was a Research Associate at the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong. His research interests include autonomous driving, artificial intelligence, deep learning, and mobile robots.

Dr. Sun serves as an Associate Editor of IEEE Robotics and Automation Letters.

Chengju Liu (Member, IEEE) received the Ph.D. degree in control theory and control engineering from Tongji University, Shanghai, China, in 2011.

From October 2011 to July 2012, she was a Research Associate with the BEACON Center, Michigan State University, East Lansing, USA. From March 2011 to June 2013, she was a Postdoctoral Researcher with Tongji University where she is currently a Professor with the College of Electrical and Information Engineering. She is also the Team Leader of the TJArk Robot Team, Tongji University. Her research interests include intelligent control, motion control of legged robots, and evolutionary computation.

Ming Liu (Senior Member, IEEE) received the B.A. degree in automation from Tongji University, Shanghai, China, in 2005, and the Ph.D. degree from the Department of Mechanical and Process Engineering, ETH Zürich, Zürich, Switzerland, in 2013.

During his master's study at Tongji University, he stayed one year in Erlangen-Nünberg University and Fraunhofer Institute IISB, Germany, as a master visiting scholar. He is currently with the Department of Electronic and Computer Engineering, the Department of Computer Science and Engineering, and the Robotics Institute, The Hong Kong University of Science and Technology, Hong Kong. His research interests include dynamic environment modeling, deep-learning for robotics, 3-D mapping, machine learning, and visual control.

Dr. Liu was a recipient of the Best Student Paper Award at the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems 2012, the Best Paper in Information Award at the IEEE International Conference on Information and Automation 2013, the Best RoboCup Paper at the IEEE/RSJ International Conference on Intelligent Robots and Systems 2013, and twice the Winning Prize of the Chunhui-Cup Innovation Contest.

Lujia Wang (Member, IEEE) received the Ph.D. degree from the Department of Electronic Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong, in 2015.

From 2015 to 2016, she was a Research Fellow with the School of Electrical Electronic Engineering, Nanyang Technological University, Singapore. From 2016 to 2021, she was an Associate Professor with Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, Guangdong, China. She is currently a Research Assistant Professor at the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong. Her current research interests include cloud robotics, lifelong federated robotic learning, resource/task allocation for robotic systems, and applications on autonomous driving.