A Low-cost and Scalable Framework to Build Large-Scale Localization Benchmark for Augmented Reality

Haomin Liu, Linsheng Zhao, Zhen Peng, Weijian Xie, Mingxuan Jiang, Hongbin Zha, Hujun Bao, Guofeng Zhang

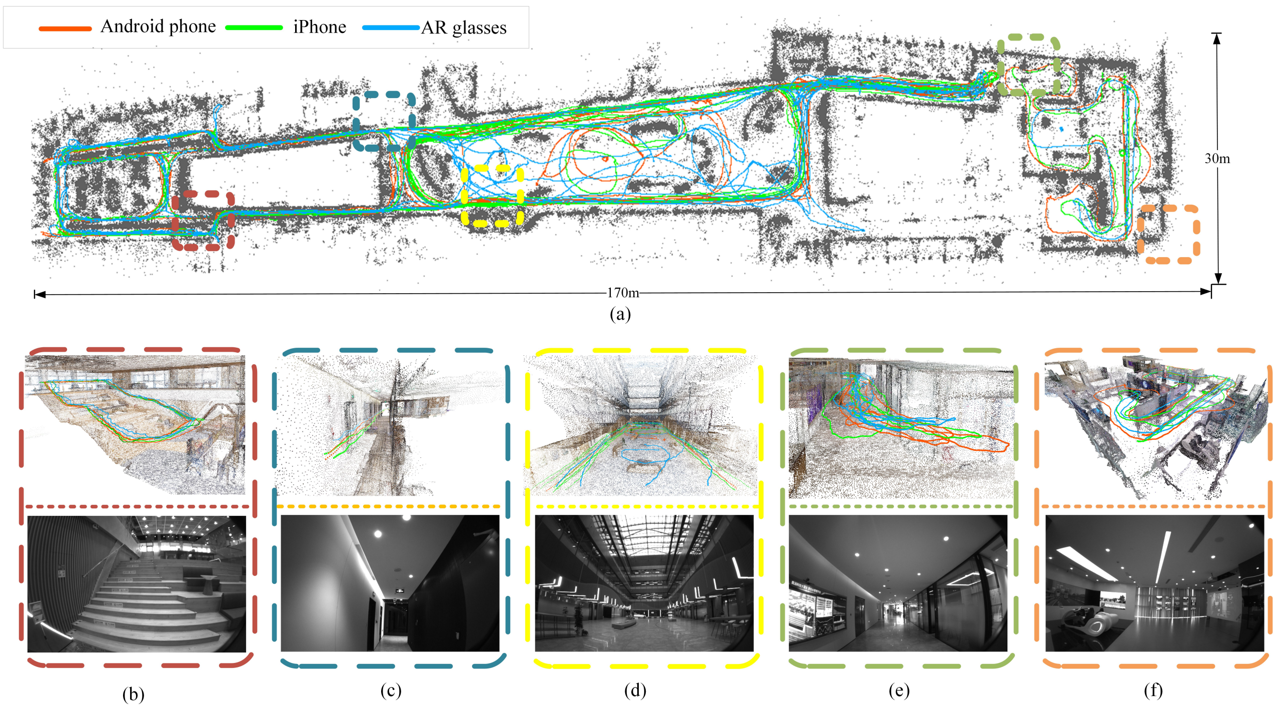


Fig. 1: The reconstructed HD map of a large-scale indoor office building, overlaid with generated groundtruth trajectories of multiple AR devices including an Android phone (red), an iPhone (green) and an AR glasses (blue). The global top view is shown in (a). We select five challenging scenarios as shown by the dashed areas boxes in (a) and the details in (b)∼(f), respectively (b) an area of stairs, (c) a narrow corridor with poor texture, (d) a very spacious atrium with most features ten meters away, (e) a wide corridor with fair texture, and (f) an exhibition hall with screens of dynamically changing content.

***Abstract*—Nowadays the application of AR is expanding from small or medium environments to large-scale environments, where the visual-based localization in the large-scale environments becomes a critical demand. Current visual-based localization techniques face robustness challenges in complex large-scale environments, requiring tremendous number of data with groundtruth localization for algorithm benchmarking or model training. The previous groundtruth solutions can only be used outdoors, or require high equipment/labor costs, so they cannot be scalable to large environments for both indoors and outdoors, nor can they produce large amounts of data at a feasible cost. In this work, we propose LSFB, a novel low-cost and scalable framework to build localization benchmark in large-scale indoor and outdoor environments. The key is to reconstruct an accurate HD map of the environment. For each visual-inertial sequence captured in the environment, the groundtruth poses are obtained by joint optimization taking both the HD map and visual-inertial constraints. The experiments demonstrate the obtained groundtruth poses have cm-level accuracy. We use the proposed method to collect a localization dataset by mobile phones and AR glasses in various environments with various motions, and release the dataset as the first large-scale localization benchmark for AR.**

***Index Terms*—Augmented Reality (AR), Benchmark, SLAM, Visual Localization, Indoor Localization.**

Haomin Liu is with Peking University, Beijing, China, and is also with SenseTime Research, Beijing, China (e-mail: liuhaomin@sensetime.com).

Linsheng Zhao, Zhen Peng, Weijian Xie, and Mingxuan Jiang are with SenseTime Research, Hangzhou, China (e-mail: {zhaolinsheng, pengzhen1, xieweijian, jiangmingxuan }@sensetime.com).

Hongbin Zha is with Peking University, Beijing, China (e-mail: zha@cis.pku.edu.cn).

Hujun Bao and Guofeng Zhang are with the state key lab of CAD&CG, Zhejiang University, Hangzhou, China (e-mail: bao@cad.zju.edu.cn; zhangguofeng@zju.edu.cn). Guofeng Zhang is the correspondence author.

I. Introduction

Localization is the core to augmented reality (AR). Seamless blending of the virtual content and the physical world requires localizing the 6 degree of freedom (6 DoF) pose

TABLE I: Comparison of existing groundtruth solutions

Benchmark	Groundtruth solution	Accuracy	Cost	Environment scale	Indoor / outdoor	Suitable for AR
KITTI [1]	GPS + IMU	∼ dm	medium	large	no / yes	no
Rawseeds [2]	GPS + Visual tags	∼ m	medium	medium	yes / yes	no
VI-Eye [3]	External LiDAR	∼ cm	medium	medium	yes / yes	no
NCLT [4]	GPS + IMU + LiDAR	∼ dm	medium	large	yes / yes	no
Complex Urban [5]	GPS + IMU + LiDAR	∼ dm	medium	large	yes / yes	no
NAVER LABS [6]	LiDAR + SfM	∼ cm	medium	large	yes / yes	no
EgoCart [7]	SfM	∼ cm	medium	large	yes / yes	no
PennCOSYVIO [8]	Visual tags + Manual measuring	∼ dm	medium	medium	yes / yes	yes
ADVIO [9]	IMU + Manual labelling	∼ dm – m	medium	large	yes / yes	yes
Aachen Day-Night & CMU Seasons [10]	SfM + Manual annotation	∼ cm	medium	large	yes / yes	no
UMA-VI [11]	SfM	∼ cm	low	large	yes / yes	no
EuRoC [12]	MoCap	∼ mm	high	small	yes / no	yes
ZJU-SenseTime [13]	MoCap	∼ mm	high	small	yes / no	yes
Hilti [14] for indoors	MoCap	∼ mm	high	small	yes / no	yes
Hilti [14] for outdoors	Manual measuring	∼ mm	medium	medium	no / yes	no
Proposed LSFB	HD map + VI-constraints	∼ cm	low	large	yes / yes	yes

of display device in the environment. The most widely used technique is visual-inertial SLAM (VI-SLAM), which uses the complementary visual and IMU measurements to localize the mobile device in unknown environments [15], [16]. The current VI-SLAM has achieved great accuracy and robustness in small and medium-scale environments [17], and has been integrated into AR products such as Apple ARKit, Google ARCore, and Microsoft Holenlens. The recent explosion of the concept of Metaverse, which refers to a digital world that is parallel to the physical world and accessible through AR/VR devices, has given rise to the demand for ubiquitous and persistent localization of AR devices in large-scale environments. However, current VI-SLAM techniques still face robustness challenges (localization failure or wrong localization) and the drift problem (error accumulation over time) in complex large-scale environments. Visual localization by matching current image to a pre-built high-definition (HD) map of the environment [18] is a promising way to correct errors. However, current visual localization also suffers from robustness issues in complex environments [6], [10]. Wireless signals such as WiFi, Bluetooth, and magnetic field are generally robust, but they can only achieve m-level accuracy [19] which is insufficient for AR.

To push these localization techniques forward, it is important to have an appropriate benchmark with groundtruth localization. First, benchmark is a fundamental tool to quantitatively evaluate different algorithms, to fine-tune the parameters, and to handle corner cases. Second, recently more and more deep learning-based approaches are proved to be superior to the traditional approaches for both visual SLAM [20], [21] and visual localization [22], [23], but require large amounts of data covering various types of environments and motions. Existing solutions for obtaining groundtruth either require GPS that is not available indoors [1], or require high equipment/labor costs [8], [12], so they cannot be scalable to large environments for both indoors and outdoors, nor can they produce large amounts of data at a feasible cost. Third, most wireless signal-based localization methods require collection of wireless signals associated with groundtruth locations throughout the entire environment. The groundtruth locations are traditionally labeled by human on the floor plan for each collected signal [19], hindering its large-scale promotion, and creating demand for an automatic way to obtain groundtruth locations.

In this work, we propose LSFB, a novel **L**ow-cost and **S**calable **F**ramework to build localization **B**enchmark in large-scale environments. As shown in Fig. 1, our framework is based on building an accurate HD map of the environment. For each visual-inertial sequence captured in the environment, the groundtruth poses are obtained by joint optimization taking both the HD map and visual-inertial constraints. Compared to the existing groundtruth solutions, our method is the only one with a combination of the following attributes:

- cm-level accuracy
- low cost
- scalable to large environment
- suitable for both indoor and outdoor
- suitable for AR

Detailed comparison is listed in Table I, and explained in Sec. II-A.

A preliminary conference version of LSFB appeared in [24]. In this paper, we elaborate on the details of the method, and make the following additional contributions:

- We quantitatively evaluate the accuracy of groundtruth in a medium-scale indoor scene by laser rangefinder, and in a large-scale outdoor scene by high-precision GPS, whereas the prior paper only evaluated the accuracy in a small room by VICON. The ATE for small/medium/large scene is less than 2/4/10 cm respectively. Details are in Sec. VI-A.
- We extend the dataset with various types of challenging scenarios (spacious, textureless, dynamic, stairs, etc) and motions (looking around, nearly rotation, running, etc), and also extend with wireless signals (WiFi, Bluetooth, and magnetic field) in the indoor environments. Details are in Sec. VI-B and shown in the supplementary video. This extension makes this dataset the only one to date containing visual-inertial measurements and wireless signals in a variety of large-scale environments with various motions.
- We evaluate three state-of-the-art SLAM systems by the dataset to verify the validity of the dataset, listed in Sec. VI-C. The results also show that the current SLAM algorithms still face challenges of robustness in the com-

plex large-scale environments, which in turn confirms the necessity of building such a benchmark.

II. Related Work

In this section, we first compare the existing groundtruth solutions, then review the relevant localization techniques.

A. Groundtruth Solution

The existing groundtruth solutions are listed in Table I. The very popular KITTI [1] uses a high precision GPS/IMU to produce 6 DoF groundtruth poses in large-scale outdoor environments. To extend to indoor area where GPS is not available, Rawseeds [2], VI-Eye [3], NCLT [4], Complex Urban [5], and NAVER LABS [6] propose to use visual tags or LiDAR scan matching as additional pose constraints. In Rawseeds [2], the visual tags are attached to the moving robot, observed by an external network of pre-calibrated cameras and laser scanners. Similarly in VI-Eye [3], external LiDARs are required to be deployed in the environment. The deployment of external cameras and/or LiDARs limits the scalability. Opposite to this outside-in solution, NCLT [4], Complex Urban [5], and NAVER LABS [6] use the inside-out measurement of LiDAR scan matching, alleviating the scalability limitation. NAVER LABS [6] further uses structure from motion (SfM) with 6 industrial cameras and 4 smartphones mounted on the mapping platform to refine poses, achieving cm-level accuracy. EgoCart [7] also obtains groundtruth poses by SfM, but it only uses two stereo cameras mounted on a shopping cart (for example the Amazon Dash Cart[1]). All these methods have to mount the localization sensors on a wheeled ground vehicle/robot/cart to collect data. The motion style is quite different from the handheld/wearable AR devices, and the equipment and operation cost is not low.

By contrast, PennCOSYVIO [8] and ADVIO [9] use a handheld VI-sensor as the localized device, thus they are suitable for AR and the device cost is lower. But they respectively require manually measured visual tags placed in the environment [8], or manually labeled locations on the floor plan [9], therefore, the operation cost is higher. The requirement of deploying visual tags by PennCOSYVIO also limits the environment scale. Aachen Day-Night and CMU Seasons proposed in [10] are two commonly used datasets for long-term visual localization. Groundtruth poses are obtained by SfM [25]. Considering the challenges of day-night and across seasons, manually labeled correspondences are required, increasing operation costs. Also the unordered images are not suitable for AR. UMA-VI [11] also obtain groundtruth poses by SfM, but only for the start and end segments for each trajectory, since all challenges occur in the middle segments and the images cannot be registered by SfM. It is not suitable for AR that requires groundtruth for each frame, and offers limited help in improving robustness to the challenging cases.

In terms of accuracy, the localization error of groundtruth for most of the aforementioned solutions are dm$\sim$m according to the literature [9], which is insufficient for AR. The most accurate solution to date requires a high-precision motion capture (MoCap) system such as the VICON system, which is used by EuRoC [12] and ZJU-SenseTime [13], providing groundtruth of mm-level accuracy. However, they require many expensive cameras deployed in the environment ($\sim$30K dollar for a 20 m^2 room). For large-scale environments, the cost will be prohibitively high. Hilti [14] proposes different solutions for indoors and outdoors. For indoors, it also uses MoCap as EuRoC and ZJU-SenseTime. For outdoors, data is collected in a "stop and go" fashion, and a total station is used to measure the prism during the periods of "stop". The manual measuring increases the operation cost and limits the scalability, and missing groundtruth during "go" periods makes it not suitable for AR.

Compared to the existing methods, LSFB proposed in this work does not require manual labeling, or deploying equipment or markers in the environments, thus is low-cost and scalable to large environments both indoors and outdoors. The localized device does not need to be mounted on the platform, and can be moved freely to collect various motion data suitable for AR. The only requirement is that the HD map of the environment should be accurately reconstructed before data collection, which is enabled by the recently matured 3D reconstruction techniques [26]–[31]. With the help of HD map, LSFB can achieve cm-level accuracy.

B. Localization

Simultaneous Localization and Mapping (SLAM) is one of the most widely used localization techniques for AR. PTAM [32] is the first visual SLAM (V-SLAM) system that is able to track camera motion in real-time on mobile devices. Bundle Adjustment (BA) [33], [34] that jointly optimizes keyframe poses and 3D points is performed in a background thread to ensure accuracy without damaging real-time performance. This framework is used by most of the recent V-SLAM systems [35]–[37]. Visual-Inertial SLAM (VI-SLAM) combines complementary visual and inertial measurements to achieve better accuracy and robustness, and becomes a standard technique for current AR applications. The most successful VI-SLAM systems include MSCKF [38], OKVIS [39], VINS [15], ORB-SLAM3 [16], etc. Delmerico and Scaramuzza give an elaborated evaluation of the state-of-the-art VI-SLAM systems [17].

SLAM localizes camera poses in the local coordinate of the first frame. For many AR applications, localization should be performed in the global coordinate where the virtual contents are attached to. The global localization can be obtained by GPS outdoors [40], or by matching to a pre-built HD map for both indoors and outdoors [18]. The global localization can also be used to limit the accumulating error of local SLAM. For example, camera can be relocated by global localization such that the accumulated error is reset. There are also works that propose to tightly couple the constraints from global localization and from local SLAM in order to limit the accumulating error and ensure the smoothness at the same time [41], [42].

Visual localization is the key building block to obtain the global localization in the pre-built HD map, or to detect

[1] https://www.amazon.com/b?ie=UTF8&node=21289116011

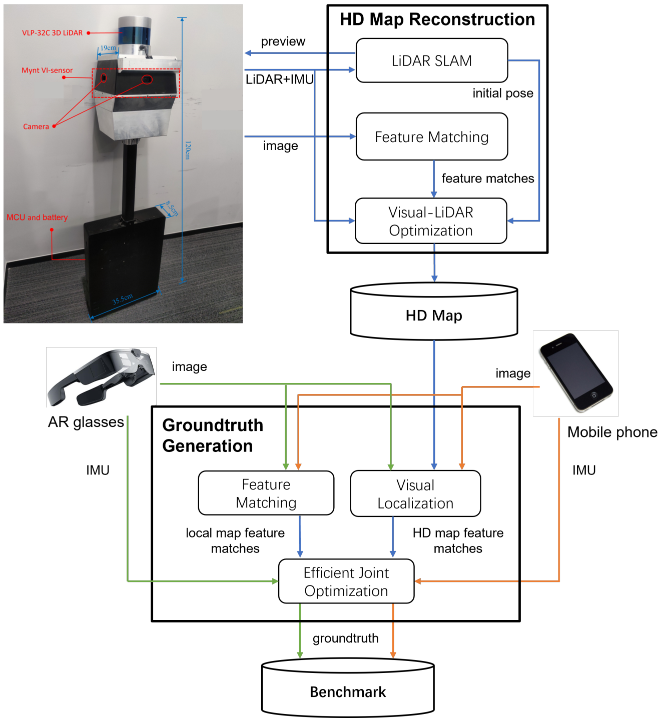


Fig. 2: The framework of the proposed method.

loops for SLAM [43]. Traditional methods [44]–[46] mainly aim to improving the efficiency and accuracy of the 2D-3D correspondence search. Recent approaches resort to learning-based features (SuperPoint [47], D2-Net [48], FSRA [49], DISAM [50] etc) to improve the robustness against viewpoint changes, motion blurs, and illumination variations. Localization can also be estimated in an end-to-end manner (PoseNet [51], MapNet [52], LPN [53], etc) due to the strong capacity of deep neural networks. The main limitation is the generalization problem. The poses would be terribly wrong in viewpoints that are not covered by the training set. A large amount of training data with groundtruth localization is required to make these learning-based methods work. Besides, visual-based methods still face great robustness challenges of repetitive structure, poor texture, occlusions, large viewpoint changes, low light, etc [6].

Wireless signal-based localization is more robust to the challenges of visual-based methods, but usually requires the deployment of signal transmitter in the environment, such as WiFi Access Point (AP), Ultra Wideband (UWB), and Bluetooth Beacon [54]–[56]. Spatial variation of the ambient magnetic field caused by magnetic material in the structures of buildings can also be a useful signal for indoor localization [57], [58], which eliminates the need for the deployment of signal transmitter. However, building the signal map of the environment usually requires manual labeling of the collected wireless/magnetic measurements on the floor plan [19]. In addition, these methods can only achieve m-level accuracy according to a recently held competition for wireless signal-based indoor localization [19], thus they are insufficient for AR. Nevertheless, wireless signal-based methods can be combined with visual-based methods owning to their complementary nature in terms of accuracy and robustness.

III. Method Overview

The framework of the proposed method is illustrated in Fig. 2. In this section, we give an overview of the proposed method, focusing on how the building blocks are integrated to build the benchmark. The details will be elaborated later in Sec. IV for HD map reconstruction and in Sec. V for groundtruth generation.

First we introduce the coordinate system and notation used in this paper. The vector coordinates have their reference system as the left superscript. ${}^{B}\mathbf{T}_{A}$ denotes the transformation from coordinate A to B by

$$ {}^{B}\mathbf{X} = {}^{B}\mathbf{T}_{A} \circ {}^{A}\mathbf{X} = \begin{bmatrix} {}^{B}\mathbf{R}_{A} & {}^{B}\mathbf{t}_{A} \end{bmatrix} \begin{bmatrix} {}^{A}\mathbf{X} \\ 1 \end{bmatrix} \tag{1} $$

where $\mathbf{R}$ and $\mathbf{t}$ are rotation and translation components respectively. $f({}^{W}\mathbf{T}_{C}, \mathbf{X})$ is defined as the function that projects a 3D point in world coordinate ${}^{W}\mathbf{X}$ to the image by camera pose ${}^{W}\mathbf{T}_{C}$

$$ f({}^{W}\mathbf{T}_{C}, {}^{W}\mathbf{X}) = \pi\left(\mathbf{K}\left({}^{W}\mathbf{T}_{C}^{-1} \circ {}^{W}\mathbf{X}\right)\right) \tag{2} $$

where $\mathbf{K}$ is the pre-calibrated camera intrinsic matrix, and $\pi([x;y;z]) = [x/z;y/z]$.

We develop a backpack 3D scanner for HD map reconstruction. The scanner is comprised of a Velodyne Ultra Puck VLP-32C 3D LiDAR to capture the 3D point cloud, and a customized Mynt VI-sensor with four cameras to capture the 360° images of the environment, as shown in the top-left picture of Fig. 2. The total weight is 12.5kg and the dimensions are illustrated in the picture. Other similar off-the-shelf 3D scanners (such as NavVis VLX[2], BMS3D[3], etc) can also be used. To guarantee the quality of data acquisition, we run a 3D LiDAR-SLAM that performs real-time dense mapping during data acquisition (Sec. IV-B). The reconstruction result is displayed in the handheld screen, guiding the operator to perform a complete and algorithm-friendly data acquisition. The HD map is comprised of a set of 3D feature points, associated with feature descriptors for the subsequent visual localization. 2D features are extracted and matched throughout the 360° images. The 3D points could be obtained by triangulation from 2D feature points and known camera poses [59]. However, we find that the pose from online LiDAR-SLAM is not accurate enough. Directly using this pose for triangulation would result in noisy HD map. We propose an offline optimization taking both the visual and LiDAR constraints to further refine the HD map (Sec. IV-C).

Once the HD map is reconstructed, it can be used to generate grountruth localization for all the mobile devices moving in the same environment. For each frame in the localized sequence, visual localization is performed to find a set of feature correspondences between the 2D feature points in the localized images and the 3D feature points in the HD map (Sec. V-A). We call the obtained 2D-3D correspondences as *HD map feature matches*. The camera poses could be simply obtained from these correspondences by PnP [60]. However, visual localization has robustness difficulties in the complex

[2] https://www.navvis.com/vlx
[3] https://www.aniwaa.com/product/3d-scanners/viametris-bms3d/

large-scale environments [6], insufficient as groundtruth. In addition, there would be jitters due to lack of temporal constraint, thus it is not suitable for AR with strict requirement on smoothness. We propose to use the visual-inertial measurements in the localized sequence as temporal constraint to improve robustness and smoothness. For this end, we extract and match features throughout the localized image sequence. We call them *local map feature matches*. Groundtruth generation is formulated as an optimization problem taking the visual-inertial constraints (Sec. V-B). The optimization is solved by Bundle Adjustment (BA) [33]. Since BA is known to be sensitive to the initial value, it is the most ideal to perform BA after each frame is added. However, it would quickly become computationally infeasible as the number of frames grows. We propose an efficient optimization strategy to tackle this challenge (Sec. V-C).

IV. HD Map Reconstruction

This section elaborates the hardware and software solution for HD map reconstruction. The solution is similar to NAVER LABS [6], where a camera-LiDAR mounted 3D scanner is developed for high-quality data acquisition, and the visual and LiDAR measurements are fused for accurate 3D reconstruction. Compared to NAVER LABS, we focus on accurate reconstruction of 3D feature points which will be used for subsequent groundtruth generation, whereas NAVER LABS focuses on camera poses which are directly used as groundtruth.

A. Backpack 3D Scanner

As shown in the top-left picture of Fig. 2, a 3D LiDAR is placed on top of a fixed structural part containing an array of 4 cameras capturing the 360° images of the environment and an IMU. The camera specification is listed in Table III. All sensors are hardware synchronized and carefully calibrated.

For each of the 4 cameras i, we separately use Kalibr [61] to calibrate the intrinsic parameters and the camera-IMU extrinsic ${}^{I}\hat{\mathbf{T}}_{C_i}$, and use AutoWare [62] to calibrate the LiDAR-camera extrinsic ${}^{C_i}\hat{\mathbf{T}}_{L}$. Other calibration tools such as [63], [64] can also be used. To account for the error in each of the separated calibration, we construct the pose graph to further optimize the 4 camera-IMU extrinsics ${}^{I}\mathbf{T}_{C_i}$ and the LiDAR-IMU extrinsic ${}^{I}\mathbf{T}_{L}$ jointly

$$\mathop{\arg\min}_{{}^{I}\mathbf{T}_{C_i},{}^{I}\mathbf{T}_{L}} \sum_{i=1}^{4} \|\log({}^{I}\mathbf{T}_{C_i} \circ {}^{I}\hat{\mathbf{T}}_{C_i}^{-1})\|^2_{\Sigma_C} + \sum_{i=1}^{4} \|\log({}^{I}\mathbf{T}_{C_i}^{-1} \circ {}^{I}\mathbf{T}_{L} \circ {}^{C_i}\hat{\mathbf{T}}_{L}^{-1})\|^2_{\Sigma_L} \tag{3}$$

where function $\log(\cdot)$ converts the rigid transformation to the minimal 6D residual vector. The covariance matrix Σ_C/Σ_L accounts for the error in the separated calibration, which is set to be the diagonal matrix with all rotation components 0.01 rad^2 for both Σ_C and Σ_L, all translation components 0.03 m^2 for Σ_C and $\mathrm{diag}(0.03, 0.03, 0.15)$ for Σ_L. We set the z-translation component of Σ_L to a larger value, because the vertical resolution of LiDAR measurement is much lower than the horizontal resolution, leading to larger calibration error in the z-translation.

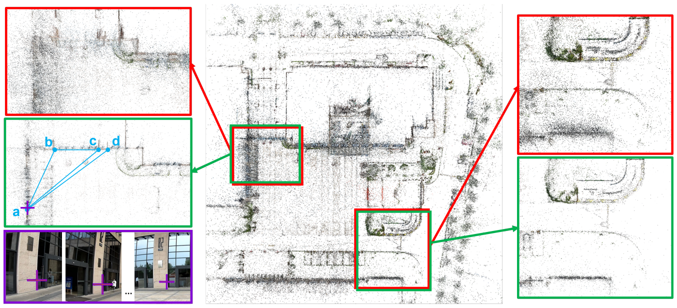


Fig. 3: The reconstructed HD map using poses of LiDAR SLAM to triangulate features (red box), and refined by visual-LiDAR optimization (green box). After optimization, the noise of point is alleviated and the 3D structure becomes clearer. Four 3D points are selected to evaluate the accuracy of four distances among the four points (blue dots and lines). The 3D points are triangulated from manually labeled 2D points on the images (purple crosses in the purple box).

B. LiDAR SLAM

We select Google Cartographer [28] for its robustness, convenience and the ability to close large loops in real-time. Other LiDAR SLAM such as [65], [66] can also be used as alternatives. During online scanning, IMU measurements are used to project LiDAR points to the latest LiDAR frame at time i (called undistortion). The optimal LiDAR pose ${}^{W}\mathbf{T}_{L_i}$ that best aligns the undistorted point cloud to the global map is recovered in real-time. The stream of 360° images, the undistorted LiDAR point clouds and the LiDAR poses are stored for offline visual-LiDAR optimization. The LiDAR point cloud on each LiDAR frame is downsampled to 20% for compression purpose.

C. Visual-LiDAR Optimization

Given the undistorted LiDAR point cloud for each frame, we first use LOAM [67] to extract plane and edge features and obtain a set of plane/edge correspondences between point clouds by ICP [68]. Denote the plane and edge correspondence set as $\mathbb{L}_p = \{({}^{n}\mathbf{p}, {}^{m}\mathbf{q}, {}^{m}\mathbf{n})\}$ and $\mathbb{L}_e = \{({}^{n}\mathbf{p}, {}^{m}\mathbf{q}, {}^{m}\mathbf{l})\}$ respectively, where ${}^{n}\mathbf{p}$ is the feature point in the source cloud n, ${}^{m}\mathbf{q}$ the matched point in the target cloud m with the normal vector ${}^{m}\mathbf{n}$ for plane feature and the direction vector ${}^{m}\mathbf{l}$ for edge feature. The LiDAR constraint $\mathbb{L}_p$ and $\mathbb{L}_e$ are combined with the visual constraint to refine the accuracy of HD map. The set of image feature correspondences is denoted as $\mathbb{V} = \{x_{ij}\}$, where x_{ij} is the 2D image observation of the 3D feature j in frame i. Combining LiDAR and visual constraints, we jointly optimize the LiDAR poses ${}^{W}\mathbf{T}_{L_i}$ and 3D feature points $\mathbf{X}_j$

$$\begin{aligned}\mathop{\arg\min}_{{}^{W}\mathbf{T}_{L_i},\mathbf{X}_j} & \sum_{({}^{n}\mathbf{p},{}^{m}\mathbf{q},{}^{m}\mathbf{n})\in\mathbb{L}_p} \|{}^{m}\mathbf{n}^T \cdot ({}^{W}\mathbf{T}_{L_m}^{-1} \circ {}^{W}\mathbf{T}_{L_n} \circ {}^{n}\mathbf{p} - {}^{m}\mathbf{q})\|^2_{\Sigma_p} \\ & + \sum_{({}^{n}\mathbf{p},{}^{m}\mathbf{q},{}^{m}\mathbf{l})\in\mathbb{L}_e} \|{}^{m}\mathbf{l} \times ({}^{W}\mathbf{T}_{L_m}^{-1} \circ {}^{L_m}\mathbf{T}_{L_n} \circ {}^{n}\mathbf{p} - {}^{m}\mathbf{q})\|^2_{\Sigma_e} \\ & + \sum_{\mathbf{x}_{ij}\in\mathbb{V}} \|f({}^{W}\mathbf{T}_{L_i} \circ {}^{L}\mathbf{T}_{C}, \mathbf{X}_j) - \mathbf{x}_{ij}\|^2_{\Sigma_v}\end{aligned} \tag{4}$$

where the three residuals are LiDAR point-to-plane, LiDAR point-to-edge distance error and image feature reprojection error. Σ_p, Σ_e and Σ_v are the covariance matrices for the three

TABLE II: The accuracy comparison between LiDAR-SLAM and visual-LiDAR optimization.

		ab	ac	ad	bc	avg
dist. (m)	laser rangefinder	11.07	15.72	16.96	7.62	
err. (cm)	LiDAR-SLAM	41.45	33.20	29.51	22.40	31.64
	vis.-LiDAR opt.	5.25	1.04	19.89	0.26	6.61

residuals, which are set to be the diagonal matrices with diagonal elements 0.2 m^2, 0.5 m^2 and 2 pixel^2 respectively. ${}^W\mathbf{T}_{L_i}$ is initialized by the result of LiDAR SLAM and X_j is initialized by triangulation [59]. Eq. 4 is solved iteratively using the LM algorithm [69].

The plane/edge correspondences depend on LiDAR poses. After LiDAR poses are refined, we update the plane/edge correspondences and solve Eq. 4 again. This process is repeated until convergence. The reconstructed HD map is visualized in Fig. 3. It can be observed that directly using the poses of LiDAR SLAM to triangulate features results in very noisy visual map. By the proposed visual-LiDAR optimization, the reprojection error corresponding to the third term in Eq. 4 is reduced from 11.62 to 1.10 pixels. To quantitatively evaluate the accuracy, we select four 3D points that are triangulated from manually labeled 2D points on the images. We use a laser rangefinder to measure four distances among the four points, and compare them with the distances from the 3D points triangulated by poses before and after the proposed optimization. The results are listed in Table II. The average distance error is reduced from 31.64 to 6.61 cm after optimization.

V. Groundtruth Generation

With an accurate HD map of the environment and a localized visual-inertial sequence captured in it, we integrate the global visual localization constraint from HD map (Sec. V-A) and the local motion constraint from visual-inertial measurements to formulate groundtruth generation as a joint optimization problem (Sec. V-B), and propose a strategy to solve the problem efficiently for large-scale dataset (Sec. V-C).

A. Visual Localization

We extract features for each localized image and match against HD map to find the HD map feature matches. Compared with our prior work [24], we replace the traditional handcrafted SIFT [70] feature with the deep learning-based feature SuperPoint [47] for its superior robustness. Each 2D image feature is associated with a 3D point in the HD map by an efficient method for Approximate Nearest Neighbor (ANN) search. Specifically, all features in the HD map are indexed using product quantization [71] in advance. For each feature in the query image, the k nearest neighbours from the HD map are retrieved, where $k = 25$ in our experiments. The voting strategy proposed in [72] is used to remove outliers whose votes are less than 6. Outliers are further removed by RANSAC [73] with P3P algorithm [60]. For AR glasses with stereo camera, inliers should satisfy the reprojection constraint for both left and right images.

B. Visual-Inertial Constraints

The visual-inertial constraints used to generate groundtruth are comprised of the HD map feature matches obtained by visual localization (Sec. V-A), the local map features matches in the localized sequence, and the IMU measurements between consecutive frames.

To find the local map feature matches, we select both KLT [74] and SIFT [70] feature for their complementary property, i. e. better robustness of KLT, and better accuracy of SIFT and its ability to close loops. We use the KLT feature for consecutive frame tracking and estimate the camera motion combined with IMU measurements, similar to the VIO systems [15], [38]. Then, we use the local map-based method to find matches for SIFT features, similar to ORB-SLAM [36]. Specifically, the local map of 3D points generated by SIFT features are projected to the current frame, and the feature matches are searched around the projection region. The similar idea is also used to enhance feature matching against HD map used in our prior work [24], which proposed to match HD map features for each individual frame. In this work, we leverage the motion estimates between adjacent frames to promote feature matching. The HD map points that were successfully matched to the previous frames but failed to be matched to the current frame by visual localization (Sec. V-A) are given a second chance, by projecting them to the current frame and searching around the projection region. In this way, the number of commonly matched HD points between consecutive frames can be increased, by which the jumpiness of visual localization can be alleviated. For the SuperPoint features that are extracted but not matched to the HD map, we also add them to the local map, and match them to the subsequent frames.

In addition to the local map-based feature matching, we perform loop detection to match the local map features among non-consecutive frames. In this way, even if it sometimes fails to match to the HD map, it has a certain ability to eliminate the accumulating error of VIO. We use VLAD [75] to categorize feature descriptors into clusters according to the pre-trained VLAD codebook, and retrieve the most similar image k for each current image i. For each local map feature in image i, we search in the corresponding cluster for the feature with the closest descriptor in image k as the candidate feature match. Candidates are filtered by RANSAC with P3P algorithm.

Combining the visual constraint from both HD map and local map feature matches, and the inertial constraint from IMU measurements, the problem of groundtruth generation is formulated as the optimization that jointly optimizes the camera motion parameters and the 3D points of local features:

$$\begin{aligned}\arg\min_{\mathbf{C}_i,\mathbf{X}_j} \; & \sum_i \left(\sum_j \|f({}^W\mathbf{T}_i,\mathbf{X}_j^*) - \mathbf{x}_{ij}^*\|^2_{\Sigma^*} + \sum_j \|f({}^W\mathbf{T}_i,\mathbf{X}_j) - \mathbf{x}_{ij}\|^2_{\Sigma} \right) \\ & + \sum_i h(\mathbf{C}_i,\mathbf{C}_{i+1})\end{aligned} \tag{5}$$

where $\mathbf{C}_i = ({}^W\mathbf{T}_i, \mathbf{v}_i, \mathbf{b}_a, \mathbf{b}_g)$ are camera motion parameters for image i, comprised of the camera pose ${}^W\mathbf{T}_i$, the velocity $\mathbf{v}_i$ and the IMU bias $\mathbf{b}_a$ and $\mathbf{b}_g$ for acceleration and gyroscope measurement respectively. $\{(\mathbf{X}_j^*, \mathbf{x}_{ij}^*)\}$ and $\{(\mathbf{X}_j, \mathbf{x}_{ij})\}$ are respectively the set of HD map and local map feature matches

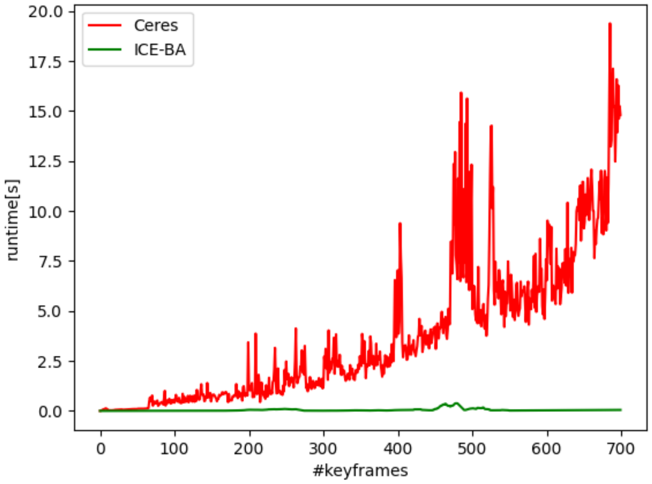


Fig. 4: The runtime comparison of BA

found in image i, with $\mathbf{X}_j^*$ the fixed 3D point in the HD map and $\mathbf{X}_j$ the 3D point of local feature to be optimized. $\mathbf{x}_{ij}^*$ and $\mathbf{x}_{ij}$ are the corresponding 2D feature measurement in the image. $h(\cdot)$ is the IMU cost function evaluating the difference between relative camera motion and the pre-integration result calculated from IMU pre-integration [76].

C. Efficient Joint Optimization

Eq. 5 is solved by BA [33]. BA is essentially a non-linear optimizer, which is sensitive to the initial value. It is the most ideal to perform BA after each frame is processed to give the optimal estimate for the next frame. However, it would quickly become computationally infeasible as the number of frames grows. We leverage the keyframe strategy that is widely used in current SLAM systems [32], [36], [37] to reduce the computational complexity. Specifically, for each incoming frame i, if it observes less than 35 previous local features, it will be selected as a new keyframe, and triggers the global BA performed on all keyframes. Otherwise we record the transformation ${}^k\mathbf{T}_i$ from i to its nearest keyframe k, which is the one shares the most common features with i. The 3D point of local feature j is also transformed to the nearest keyframe k of the frame from which the feature is first extracted ${}^k\mathbf{X}_j = {}^W\mathbf{T}_k^{-1} \circ {}^W\mathbf{X}_j$. After all frames are processed, we recover the camera pose for each non-keyframe i by ${}^W\mathbf{T}_i = {}^W\mathbf{T}_k \circ {}^k\mathbf{T}_i$ and the 3D point for each local feature j by ${}^W\mathbf{X}_j = {}^W\mathbf{T}_k \circ {}^k\mathbf{X}_j$ as the initial value for the final global BA performed on all keyframes and non-keyframes.

TABLE III: Specification of cameras

	Backpack 3D scanner	AR device		
		An. phone	iPhone	AR glasses
#Cam.	4	1	1	2
Res.	1280 × 800	640 × 480	640 × 480	640 × 400
FoV	106° × 65°	67° × 53°	68° × 53°	126° × 80°
Freq.	20 Hz	30 Hz	30 Hz	30 Hz
GS / RS*	GS	RS	RS	GS

* Global shutter / rolling shutter

Even with the keyframe strategy, the bottleneck still exists since the complexity of BA grows quadratically with the number of keyframes. For large-scale dataset, it will quickly become computationally infeasible to perform global BA for each incoming keyframe. Most SLAM systems reduce the computational complexity by local BA or the pose graph approximation [15], [36], which is not accurate enough to be used as groundtruth. We use the incremental BA solver ICE-BA [77] to break this limit. ICE-BA is specialized for the VI-SLAM problem in which the incoming visual-inertial measurements arrive sequentially. It explicitly leverages this fact and re-uses the intermediate results of previous optimization to avoid redundant new computation, to achieve the exact solution with significantly higher efficiency. Fig. 4 compares the scalability to a traditional BA solver Ceres [69] whose runtime grows quadratically as expected, whereas ICE-BA remains nearly constant thus being scalable to long-time exploration in large-scale environment.

VI. Results

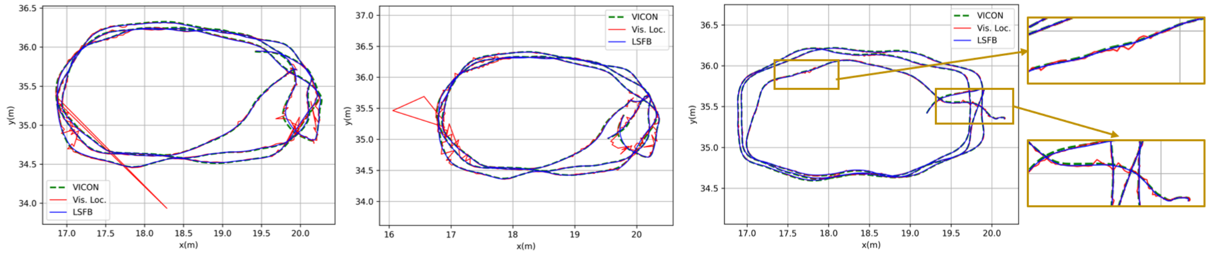
In this section, we first conduct experiments to analyze the accuracy of the proposed LSFB quantitatively (Sec. VI-A), then give a detailed introduction to the collected groundtruth dataset which will be released as a new localization benchmark for AR (Sec. VI-B). Finally, we use the dataset to evaluate and analyze three state-of-the-art SLAM systems (Sec. VI-C).

Throughout the experiments, we use the customized backpack 3D scanner to reconstruct the HD map of the environments, and use three off-the-shelf AR devices to collect the visual-inertial data in the environment. The three AR devices are: an Android phone Huawei Mate 30 Pro, an iPhone XS Max and an AR glasses of Shadow Creator Action One Pro. The camera specifications for the backpack 3D scanner and the AR devices are listed in Table III.

A. Accuracy

We conduct three experiments to quantitatively evaluate the accuracy of the proposed LSFB, in a small, medium and large scale environment respectively.

For the first experiment, we use the high-precision MoCap system VICON as proposed in [12], [13] to evaluate the accuracy of LSFB in a small room (∼20 m^2). For each of



(a) (b) (c)

Fig. 5: Trajectory comparison in a small room for (a) Android phone, (b) iPhone and (c) AR glasses. For each AR device, we compare the trajectories obtained by VICON, visual localization and LSFB respectively.

TABLE IV: RMS-ATE (cm) in the small room

		An. Phone	iPhone	AR glasses
LSFB		1.46	1.14	0.54
Visual localization	SIFT (scc. rate)	11.23 (97%)	6.81 (65%)	8.83 (98%)
	SuperPoint (scc. rate)	8.23 (100%)	5.34 (99%)	0.81 (100%)
LSFB with portion of visual localization	75%	1.45	1.20	0.64
	50%	1.55	1.46	0.69
	25%	2.34	2.02	0.83
	0%	2.50	1.82	1.36

TABLE V: RMS-ATE (cm) for AR glasses in three environments

Env. scale	small	medium	large
LSFB	0.54	3.48	9.00
LSFB w/o loop	0.56	4.19	9.18

the three AR devices, we walked around the room three times at the speed of ∼1 m/s. The camera poses estimated by both LSFB and visual localization are compared to that measured by VICON as groundtruth. The results of RMS-ATE (Root Mean Squared Absolute Trajectory Error) [78] are listed in Table IV. For visual localization, we compare ATE together with the success rate using SIFT and SuperPoint respectively. The success rate is defined as the percentage of frames that have matched more than 30 HD map features. It can be seen that both accuracy and success rate of SuperPoint are higher than that of SIFT. However, there is still frequent jitter in the trajectory of SuperPoint, as shown in Fig. 5. By contrast, LSFB generates smoother trajectory with ATE below 2cm for all devices. In this simple experiment, most frames can be matched to HD map by visual localization. This may not be the case in practice for large-scale environments. To investigate the effect of HD map on the accuracy of LSFB, we further conduct another experiment to remove a portion of HD map feature matches and evaluate the result using remaining constraints. Specifically, we divide the HD map into four quadrants equally according to the (x,y) coordinates of 3D points. Each time we remove HD map points belonging to one more quadrant, remaining namely 75%, 50%, 25% and 0% portion of the original HD map feature matches. The results are listed in Table IV. It can be seen that as long as there are more than half of the HD map constraints, the accuracy remains similar, which is obviously better than that without any HD map constraints.

The MoCap-based experiment can only be conducted in the small room due to the cost of equipment and deployment. To quantitatively evaluate the accuracy in larger environment, we design another experiment in a medium-scale indoor office (∼500 m^2). As shown in Fig. 6(a)(b), we mark 15 points $p_{1\ldots10}$ and $q_{1\ldots5}$ at the intersections of the floor tiles, so that each pair of points (p_i,p_j) and (q_i,q_j) is on the horizontal line, and each (p_i,q_i) is on the vertical line. Denote the set of horizontal pairs as $\mathbb{H}=\{(p_i,p_j)\}\cup\{(q_i,q_j)\}$, and the set of vertical pairs as $\mathbb{V}=\{(p_i,q_i)\}$. We use a laser rangefinder[4] to measure the distance for these point pairs as shown in Fig. 6(c). Note that other pairs such as (p_1,q_2) cannot be measured due to occlusion in between. We calculate the coordinate of the 15 marked points by solving the least squares problem

$$\underset{p_{1\ldots10},q_{1\ldots5}}{\arg\min}\ \frac{1}{\sigma_m^2}\left(\sum_{(p,q)\in\mathbb{H}\cup\mathbb{V}}(|p-q|-m(p,q))^2\right)+ \frac{1}{\sigma_d^2}\left(\sum_{(p_i,p_j)\in\mathbb{H}}(y(p_i)-y(p_j))^2+\sum_{(p,q)\in\mathbb{V}}(x(p)-x(q))^2\right) \tag{6}$$

where $m(p,q)$ is the distance measurement between point pair (p,q). $x(p)$ and $y(p)$ are the x and y coordinate of point p respectively. The first term accounts for the measurement errors with variance σ_m, and the second/third term accounts for the deviations of points from horizontal/vertical lines with variance σ_d. We set σ_m and σ_d to 1 mm. The resulting error of solving Eq. 6 is 2.76/0.27/9.45 mm for the three terms respectively, indicating the accuracy of the marked points, which is sufficient to evaluate the accuracy of the proposed LSFB. We fix the AR glasses on a square box and place the box on each of the marked points for a while, as shown in Fig. 6(d), along the blue trajectory shown in Fig. 6(a). The estimated localization for the 15 points are manually selected by the z-coordinates as shown in Fig. 6(e). The ATE of the 15 points is shown in Fig. 6(f), and the RMS-ATE is 3.48 cm.

To evaluate the accuracy in even larger environment, we use the high-precision GPS & GNSS equipment NovAtel

[4] We use the Deli 100m laser rangefinder with ±1 mm error.

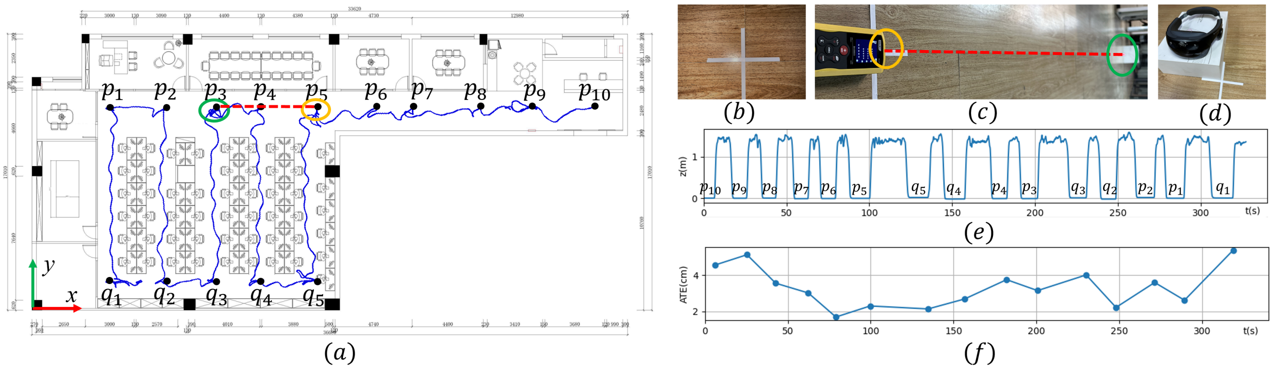


Fig. 6: Accuracy evaluation in a medium-scale indoor office. (a) The AR glasses are moved along the trajectory passing through 15 selected points. (b) The points are selected at the intersections of the floor tiles and marked as crosses. (c) The distances between point pairs are measured by a laser rangefinder. (d) The AR glasses are fixed on a square box and the box is placed on each of the marked points for a while. (e) Eleven moments when the box was resting on the ground are manually selected in the trajectory according to the z-coordinate. (f) The ATE of the 15 points on the trajectory recovered by the proposed LSFB.

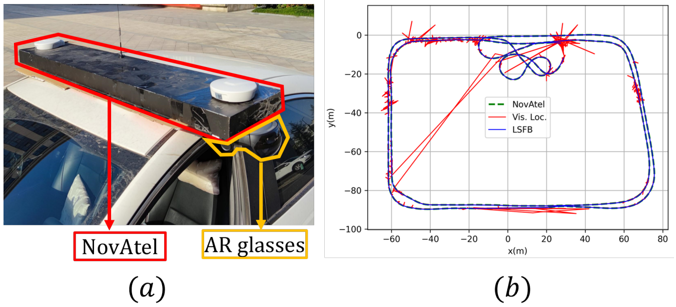


Fig. 7: Accuracy evaluation in a large-scale outdoor office park. (a) NovAtel and AR glasses are mounted on a car. (b) Trajectories obtained by NovAtel, visual localization and LSFB respectively.

PwrPak7 [5] which has mm-level accuracy. As illustrated in Fig. 7, we mount the NovAtel and AR glasses on a car, and drive a few laps in a large-scale outdoor office park (∼10000 m^2). Since both the body and world reference systems are different between NovAtel and LSFB, as well as the time reference, we should perform hand-eye calibration and time synchronization to align the two trajectories [79]. We denote the pose obtained by NovAtel at each timestamp t as ${}^G\mathbf{T}_N(t)$, and the pose obtained by LSFB as ${}^H\mathbf{T}_A(t)$, where $G/N/H$ and A are reference system of GPS/NovAtel/HD map and AR glasses respectively. We align the two trajectories as

$$\underset{{}^G\mathbf{T}_H,\, {}^N\mathbf{T}_A,\, t_0}{\arg\min} \sum_t \|\log({}^G\mathbf{T}_H \circ {}^H\mathbf{T}_A(t) \circ {}^N\mathbf{T}_A^{-1} \circ {}^G\mathbf{T}_N^{-1}(t_0+t))\|^2_{\Sigma_N} \tag{7}$$

The covariance matrix Σ_N is set to be the diagonal matrix with rotation components 0.01 rad^2 and translation components 0.003 m^2. The aligned trajectories are shown in Fig. 7(b). The RMS-ATE for LSFB and visual localization are 9.00 and 151.51 cm respectively. Compared to the first experiment in small room, the advantages of LSFB over visual localization is more prominent in large-scale environment, since most HD map points are far away and do not provide enough constraints for accurate visual localization. Incorporating temporal constraints significantly improves the accuracy and smoothness.

Note that the three data collection methods described above are experiments we conduct to evaluate the accuracy of LSFB, not the method we use to collect the dataset. The RMS-ATE for the three environments are listed in Table V. We also compare ATE without loop closure. We find that HD map is the major reason for the high accuracy. With the constraint of HD map, the accumulating error has been effectively bounded. In this case, loop closure has only a small improvement in accuracy.

B. Dataset

We use the proposed LSFB to collect localization data in various types of large-scale environments with various types of motions. The environments and motions are listed in Table VI and Table VII respectively, and details are shown in the supplementary video.

In our prior work [24], we collected data in three indoor and/or outdoor environments, listed in the first three rows of Table VI. To enrich the diversity of environments, this work further selects a large indoor office which contains a variety of typical challenging scenarios (spacious, textureless, dynamic, stairs, etc) to expand the dataset. As listed in Table VI, we first collect localization sequences covering the entire office (4-th row), then collect in each challenging scenario individually to focus on the performance in the specific scenario (the last five rows). We also extend various types of motions as listed in Table VII. The first six sequences (O0∼O2 and M0∼M2) are from the prior work [24], where the motion is mostly slowly walking forward. In this work, we extend other motions that are often present in AR applications such as looking around and sideways walking, as well as the challenging motions for visual SLAM such as nearly rotation, running, and walking through the crowd. Each motion is performed three times using Android phone, iPhone and AR glasses respectively, obtained 75 sequences in total for the 25 motions. The HD maps overlaid with the 75 groundtruth trajectories for the nine environments, and the corresponding images for the three AR devices are shown in Fig. 1 in the supplementary document.

Another extension is we collect the wireless signals in addition to the visual-inertial measurements for the indoor sequences, specifically sequences captured in "medium indoor office" and "large indoor office". For indoor localization, the visual-based methods and the wireless signal-based methods are complementary in nature, where the former is more accurate and the latter is more robust. However, the existing datasets contain only one of them and cannot be used to fuse the two information. To fill this gap, we deploy the WiFi APs, and Bluetooth Beacons in the indoor environments, and collect the WiFi and Bluetooth signals as well as the magnetic field received on the mobile phone. The timestamps of wireless signals are synchronized with visual-inertial measurements, such that the localization for each signal can be extracted from the groundtruth trajectory generated from visual-inertial

TABLE VI: Various types of environments in the dataset

Environment	Size (m×m)	Seq.[1]	Description
Office park (outdoor)	100×100	O0	Office park including large office buildings and a medium-scale plaza
Office park (indoor & outdoor)	100×100	O1, O2	Office park including houses with small rooms and passages and a spacious plaza
Medium indoor office	15×35	M0 – M2	Medium-scale indoor office
Large indoor office	135 ×30	L0 – L3	Large-scale indoor office with various challenging scenarios listed in following rows
Exhibition hall[2]	20×30	E0 – E2	Medium-scale exhibition hall with screens of dynamically changing content
Atrium[2]	15×60	A0 – A4	Spacious atrium space with 15m high ceiling
Corridor (wide)[2]	4×40	C0 – C2	Wide corridor with fair texture
Corridor (narrow)[2]	2×30	C3	Narrow corridor with poor texture
Stairs[2]	10×20	S0 – S2	Wide stairs with rich texture

[1] The sequences in Table VII.
[2] A part of "large indoor office".

[5] https://novatel.com/products/receivers/enclosures/pwrpak7

TABLE VII: Various types of motions in the dataset

Seq.[1]	#Frames[2]	Length (m)[2]	Description
O0	10528 / 9641 / 4909	490 / 464 / 240	Walk around an office building and around the medium-scale plaza
O1	5116 / 3205 / 3831	225 / 1175 / 147	Walk around a house outside
O2	8363 / 8447 / 8119	382 / 341 / 330	Walk around a house, go in and out of some rooms, and walk around the spacious plaza outside
M0	4024 / 3760 / 3342	126 / 124 / 121	Walk along a large loop, looking ahead
M1	5230 / 4741 / 3628	143 / 138 / 125	Walk along a large loop, looking around with slow motion
M2	4749 / 4315 / 2752	141 / 129 / 89	Walk along a large loop, looking around with fast motion, and occasional occlusion or textureless
L0	12369 / 11829 / 7314	454 / 478 / 317	Walk along loops, looking ahead
L1	15179 / 13967 / 9357	558 / 579 / 419	Walk along loops, looking ahead
L2	16889 / 17210 / 14632	592 / 635 / 517	Walk along loops, looking around
L3	16517 / 17012 / 14483	604 / 638 / 540	Walk along loops, looking around
E0	2011 / 1841 / 1873	50 / 51 / 44	Walk along a line and return, looking ahead
E1	5137 / 4208 / 2847	140 / 127 / 86	Walk along a loop, looking around
E2	2412 / 2518 / 2052	13 / 14 / 8	Stand still, looking around
A0	1260 / 1245 / 1486	31 / 26 / 41	Walk along a loop, looking at the center
A1	3526 / 4044 / 3955	116 / 138 / 130	Walk along a loop, looking at the outer wall
A2	2076 / 2230 / 3601	60 / 67 / 105	Walk along a line and return, looking at the distance
A3	2053 / 4816 / 2640	62 / 166 / 86	Walk through the crowd, looking around
A4	1930 / 2417 / 2049	115 / 142 / 132	Run along a loop
C0	1942 / 2040 / 2197	64 / 63 / 62	Walk along a wide corridor and return, looking ahead
C1	2443 / 1960 / 2859	66 / 59 / 76	Walk along a wide corridor and return, looking around
C2	2278 / 1871 / 2154	60 / 63 / 63	Walk sideways along a wide corridor looking at the wall, then return looking at the other side
C3	2001 / 1827 / 2441	64 / 57 / 87	Walk along a narrow and textureless corridor and return, looking ahead
S0	1620 / 1912 / 1971	39 / 41 / 40	Walk along a line downstairs then upstairs, looking ahead
S1	2346 / 2976 / 2520	52 / 59 / 58	Walk along a loop downstairs then upstairs, looking around
S2	2533 / 2448 / 2186	54 / 53 / 53	Walk sideways along a loop downstairs then upstairs, looking at the other side of the stairs

[1] The first letter of sequence name corresponds to the environment listed in Table VI.

[2] The three numbers correspond to Android phone / iPhone / AR glasses respectively.

measurements. All these extensions make the dataset the only one to date containing both visual-inertial measurements and wireless signals in various large-scale environments with various motions.

C. SLAM Evaluation

We use the proposed LSFB dataset to evaluate three state-of-the-art SLAM systems, including two VI-SLAM systems ORB-SLAM3 [16] and VINS-Fusion [15], and a learning-based V-SLAM system Droid-SLAM [21]. The results are listed in Table VIII for the dataset of mobile phones and Table IX for the dataset of AR glasses.

For the VI-SLAM systems ORB-SLAM3 and VINS-Fusion, we align the resulting trajectory to groundtruth using both SE(3) and Sim(3) [78] in order to analyze the performance under situations with and without requirement of true scale. For the V-SLAM system Droid-SLAM, we use only Sim(3) for mobile phones with monocular camera, and both SE(3) and Sim(3) for AR glasses with stereo camera. After alignment, accuracy is measured by computing RMS-ATE. Scale error is also computed using the scale estimate s from Sim(3) alignment, calculated as $|1-s|$. For each sequence, we select the median of ATE after 3 executions. There are cases that all executions fail to process the whole sequence, due to initialization failure or other robustness issues of ORB-SLAM3, or the exceeded requirement of GPU memory by Droid-SLAM for long sequence. These cases are marked as $\times$ and grayed out. We also gray out unreliable results whose ATE is larger than 10% of the length of the longer side of environment (The size of environment for each sequence is listed in Table VI). We further gray out unreliable scale estimates if the scale error is larger than 10%. The corresponding $\mathrm{ATE}_{\mathrm{SE}(3)}$ is also marked gray since the SE(3) alignment is unreliable in this case. The remaining results are regarded as successful. We count the number of successful results in the second last row for $\mathrm{ATE}_{\mathrm{SE}(3)}$ and $\mathrm{ATE}_{\mathrm{Sim}(3)}$ respectively, as an indicator of robustness for the three competitors. To compare the accuracy, we mark the minimal $\mathrm{ATE}_{\mathrm{SE}(3)}$ and $\mathrm{ATE}_{\mathrm{Sim}(3)}$ as bold for each sequence, and count the number of best results in the last row. The method with the best robustness and the best accuracy is also marked as bold in the last two rows respectively.

From the benchmark result listed in Table VIII and Table IX, VINS-Fusion achieves the best robustness, and the two VI-SLAM systems are more robust than the V-SLAM system Droid-SLAM. For mobile phones, VINS-Fusion also achieves the best accuracy in terms of both $\mathrm{ATE}_{\mathrm{SE}(3)}$ and $\mathrm{ATE}_{\mathrm{Sim}(3)}$. For each method, using AR glasses is more robust and more accurate than using mobile phones, due to the stereo camera with larger FoV and global shutter of AR glasses. In this situation, ORB-SLAM3 achieves the best accuracy in terms of $\mathrm{ATE}_{\mathrm{SE}(3)}$ and Droid-SLAM achieves the best accuracy in terms of $\mathrm{ATE}_{\mathrm{Sim}(3)}$.

To explain the reason behind the comparative results, we analyze ORB-SLAM3 and found a robustness issue during fast rotation for monocular camera with small FoV. In this case, feature track easily gets lost if the corresponding 3D point has not been successfully triangulated. That is the reason why ORB-SLAM3 is inferior to VINS-Fusion for all of the sequences with frequent motion of looking around. For stereo camera with larger FoV, by which features can be easily triangulated as soon as they first appear in the keyframes, ORB-SLAM3 achieves the best accuracy for most of the sequences in terms of $\mathrm{ATE}_{\mathrm{SE}(3)}$. We also analyze Droid-SLAM and found that the inferiority is mainly due to the rolling shutter effect of mobile phones. It is more sensitive for the dense direct methods than the sparse indirect counterparts. For AR glasses with global shutter camera, Droid-SLAM achieves the best accuracy for most of the sequences in terms of $\mathrm{ATE}_{\mathrm{Sim}(3)}$. However in terms of $\mathrm{ATE}_{\mathrm{SE}(3)}$, we find that Droid-SLAM has large scale error in most cases, because the stereo

TABLE VIII: SLAM evaluation on LSFB dataset of Android phone and iPhone (RMS-ATE in m, scale error in %).

	Android phone							iPhone						
	ORB-SLAM3			VINS-Fusion			Droid-SLAM	ORB-SLAM3			VINS-Fusion			Droid-SLAM
	$\text{ATE}_{\text{SE}(3)}$	scale error	$\text{ATE}_{\text{Sim}(3)}$	$\text{ATE}_{\text{SE}(3)}$	scale error	$\text{ATE}_{\text{Sim}(3)}$	$\text{ATE}_{\text{Sim}(3)}$	$\text{ATE}_{\text{SE}(3)}$	scale error	$\text{ATE}_{\text{Sim}(3)}$	$\text{ATE}_{\text{SE}(3)}$	scale error	$\text{ATE}_{\text{Sim}(3)}$	$\text{ATE}_{\text{Sim}(3)}$
O0	**3.379**	5.887	**2.703**	3.816	2.199	3.729	19.320	471.819	98.462	32.174	20.278	34.775	**7.306**	8.305
O1	38.482	62.760	25.475	**5.348**	7.365	4.782	**0.769**	6.165	21.710	**2.063**	273.741	850.001	189.455	3.388
O2	4307.723	99.895	13.704	5249.965	99.768	19.775	23.727	7.525	21.472	3.850	4.915	16.485	**1.458**	12.094
M0	**0.338**	2.268	0.260	0.385	3.282	**0.218**	1.802	0.461	0.410	0.459	**0.305**	0.635	**0.298**	1.161
M1	0.447	3.841	0.281	**0.243**	0.645	**0.234**	5.195	0.426	3.781	**0.246**	**0.379**	1.584	0.345	2.209
M2	0.506	4.252	**0.318**	**0.399**	1.726	0.357	3.180	0.250	1.601	**0.206**	**0.233**	0.146	0.233	2.207
L0	2.789	7.149	**0.323**	**1.727**	4.139	0.381	23.759	1.919	3.820	1.168	**1.804**	4.177	**0.726**	×
L1	×	×	×	**2.666**	5.590	**1.128**	×	**1.810**	2.351	**1.549**	2.178	3.796	1.552	×
L2	2.769	6.377	1.336	**1.609**	2.688	**1.219**	×	×	×	×	**2.620**	4.455	**1.867**	×
L3	×	×	×	**1.148**	0.082	**1.147**	×	**1.259**	0.123	1.258	1.747	3.127	**1.252**	×
E0	0.746	8.659	**0.359**	**0.666**	2.818	0.623	3.150	**0.898**	8.787	**0.387**	1.930	17.794	0.429	3.265
E1	0.936	4.690	0.824	**0.526**	4.344	**0.329**	7.448	0.870	7.108	**0.474**	**0.838**	6.250	0.563	5.544
E2	×	×	×	209.469	99.924	0.282	**0.237**	×	×	×	0.114	35.636	**0.055**	0.232
A0	**0.082**	0.157	**0.081**	0.267	5.248	0.089	0.152	**0.087**	1.426	**0.070**	0.289	7.080	0.113	0.073
A1	6.923	0.698	6.923	932.448	99.082	14.244	7.500	**1.899**	4.770	**1.649**	2.273	6.878	1.813	8.143
A2	128.776	93.907	5.656	**0.832**	8.111	0.252	**0.029**	**1.774**	1.051	1.771	544.375	98.915	9.230	**0.316**
A3	3.258	0.988	3.257	**0.375**	2.310	**0.320**	0.361	1.233	8.013	**0.418**	**0.782**	3.456	0.626	10.406
A4	2.633	18.185	**0.476**	**0.872**	3.080	0.718	6.141	5.221	15.750	4.472	**4.203**	6.288	4.002	**2.201**
C0	127.864	94.300	6.298	435.002	99.459	9.665	8.714	**0.351**	0.977	0.337	0.524	4.145	0.262	**0.144**
C1	0.509	3.490	0.365	**0.407**	3.642	**0.137**	1.550	0.756	4.350	0.620	**0.491**	2.784	0.416	**0.097**
C2	**0.326**	2.831	**0.249**	0.370	0.392	0.368	4.097	1.037	3.139	1.005	**0.834**	6.200	**0.564**	2.050
C3	227.019	97.277	7.109	267.608	98.702	8.832	7.238	×	×	×	**0.536**	1.868	**0.506**	6.889
S0	0.269	0.771	0.265	**0.104**	1.063	**0.084**	0.780	0.648	4.723	0.568	**0.265**	4.117	**0.075**	4.868
S1	0.275	2.177	0.226	**0.126**	0.695	**0.115**	0.787	**0.287**	1.194	0.271	0.460	5.336	**0.197**	4.904
S2	0.303	4.126	**0.086**	**0.131**	0.963	0.110	5.306	**0.272**	2.935	0.158	163.701	97.579	6.032	**0.144**
#scc.	15		17	**20**		**21**	10	**18**		21	**18**		**22**	13
#best	4		8	**16**		**10**	3	9		9	**12**		**11**	5

TABLE IX: SLAM evaluation on LSFB dataset of AR glasses (RMS-ATE in m, scale error in %).

	ORB-SLAM3			VINS-Fusion			Droid-SLAM		
	$\text{ATE}_{\text{SE}(3)}$	scale error	$\text{ATE}_{\text{Sim}(3)}$	$\text{ATE}_{\text{SE}(3)}$	scale error	$\text{ATE}_{\text{Sim}(3)}$	$\text{ATE}_{\text{SE}(3)}$	scale error	$\text{ATE}_{\text{Sim}(3)}$
O0	×	×	×	3.276	6.960	2.239	**2.493**	5.157	**1.690**
O1	**1.630**	6.890	0.900	2.864	15.906	0.684	1.947	10.178	**0.281**
O2	×	×	×	2.836	13.011	**1.223**	×	×	×
M0	**0.108**	0.080	0.107	0.259	1.200	0.231	0.823	7.835	**0.050**
M1	0.221	2.140	0.068	**0.190**	0.400	0.186	1.016	9.634	**0.058**
M2	0.474	2.930	0.349	**0.401**	2.700	0.265	1.164	9.538	**0.071**
L0	**0.406**	0.110	**0.404**	1.629	2.259	1.344	×	×	×
L1	**0.892**	0.210	**0.888**	2.378	1.635	2.289	×	×	×
L2	**0.789**	1.560	**0.489**	2.405	2.918	1.994	×	×	×
L3	**0.993**	2.180	**0.502**	4.205	8.227	1.712	×	×	×
E0	0.179	2.380	0.045	**0.133**	1.740	**0.031**	0.894	10.212	0.068
E1	**0.122**	1.250	0.059	0.153	0.136	0.153	0.534	6.011	**0.033**
E2	**0.018**	4.380	**0.016**	0.029	0.922	0.029	0.047	6.088	0.074
A0	**0.072**	0.940	0.044	0.146	2.107	0.070	0.818	11.875	**0.038**
A1	**0.289**	1.270	0.164	0.718	0.161	0.717	1.482	7.193	**0.066**
A2	×	×	×	3.393	20.077	2.854	2.887	14.941	**2.576**
A3	**0.131**	1.660	0.067	0.263	0.290	0.263	0.777	10.526	**0.043**
A4	**0.399**	0.150	0.398	0.851	4.517	**0.110**	2.220	11.247	0.162
C0	**0.160**	1.570	0.047	0.193	2.048	0.040	1.150	10.800	**0.028**
C1	**0.073**	0.590	0.047	0.298	3.039	0.066	1.101	10.387	**0.038**
C2	**0.199**	1.710	0.104	0.326	3.245	0.056	1.104	10.100	**0.052**
C3	0.200	1.360	0.100	**0.137**	0.615	0.114	0.643	4.670	**0.054**
S0	**0.066**	0.270	0.064	0.150	1.847	0.107	0.538	8.384	**0.041**
S1	0.298	4.070	0.043	**0.147**	1.759	0.081	0.740	9.628	**0.038**
S2	**0.182**	2.460	**0.028**	0.442	5.795	0.084	1.105	13.602	0.038
#scc.	**22**		22	**22**		**25**	10		20
#best	**17**		6	5		3	1		**16**

camera fails to provide sufficient parallax for far points in large-scale environment. Actually the lack of parallax is also a common robustness challenge for the VI-SLAM competitors in large-scale environments. For example in E2, the person is standing still while looking around. For mobile phones, ORB-SLAM3 never succeeds in the initialization phase, and VIN-Fusion fails to recover the true scale. By using stereo camera, this problem is mitigated but not eradicated. For example in O2 and A2 when the person is walking in the spacious environment and looking at the distance ten meters away, the stereo camera could not provide enough parallax and scale error is large for VI-SLAM systems. Another common robustness challenge is textureless. For example in C3, the person is walking along a narrow and textureless corridor. In this case, there are few features for mobile phones with small FoV, which is a great challenge for SLAM. These robustness issues confirm the necessity of building such a benchmark and the merit of our collected dataset. To further verify the necessity of the proposed dataset, we evaluate the three SLAM systems using two previous datasets EuRoC [12] and ZJU-

SenseTime [13]. The robustness issues are only found by ORB-SLAM3 on the challenging half of the data in ZJU-SenseTime. By contrast, more robustness issues for each of the three SLAM systems can be found by the proposed dataset. Details are referred to Sec. I-C in the supplementary document.

To further analyze the inferior SLAM results of mobile phone, we evaluate another SLAM system VINS-Mono [80], which is specified for mobile phone with monocular rolling shutter camera. The comparison shows that the rolling shutter effect is a major source of localization error for mobile phones. Explicitly rectifying the rolling shutter can effectively improves the accuracy. Details are referred to Sec. I-D in the supplementary document.

VII. Conclusion

In this work, we present LSFB, a novel groundtruth collection method for localization in large-scale environment. To the best of our knowledge, it is the first groundtruth framework with low-cost and scalability to collect large amounts of data in the large-scale environment. The experiments demonstrate the obtained groundtruth poses have cm-level accuracy. We use the proposed method to collect a multimodal localization dataset containing both visual-inertial measurements and wireless signals captured by mobile phones and AR glasses in various environments with various motions, and release the dataset as the first large-scale localization benchmark for AR. We also use the benchmark to evaluate and analyze three state-of-the-art SLAM systems to verify the validity of the dataset, and confirm the necessity of building such a benchmark.

The current solution relies on the precision of visual localization. In case that the visual localization produces too many false positives, such as when there are too many repetitive structures in the environment, or when a large part of the environment has changed, the accuracy will be reduced. Another limitation is the requirement of people to carry a heavy backpack to collect HD map. The size of such HD map is as far as up to several hundreds of thousands of square meters. It is infeasible to collect data in the city-scale environments. For the future work, we will further improve the robustness, and set up the 3D scanner to a vehicle to achieve city-scale HD map reconstruction, in order to collect the city-scale localization data to further expand our benchmark. We hope that it would genuinely promote the localization techniques and the realization of AR applications in the city-scale environments.

VIII. acknowledgements

This work was partially supported by NSF of China (No. 61932003) and ZJU-SenseTime Joint Lab of 3D Vision. Thanks Jin Yuan for the evaluation of VINS-Fusion and VINS-Mono, and Longrui Dong for the evaluation of Droid-SLAM. Thanks Zhuang Zhang, Xiaopeng Huang, Meng Hang and Youji Feng for their contributions in the prior conference paper of this work. Particularly thanks Zhuang Zhang for providing additional quantitative evaluation of visual-LiDAR optimization in this paper.

References

[1] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the KITTI vision benchmark suite,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3354–3361.

[2] S. Ceriani, G. Fontana, A. Giusti, D. Marzorati, M. Matteucci, D. Migliore, D. Rizzi, D. G. Sorrenti, and P. Taddei, “Rawseeds ground truth collection systems for indoor self-localization and mapping,” *Autonomous Robots*, vol. 27, no. 4, p. 353, 2009.

[3] Y. He, L. Ma, Z. Jiang, Y. Tang, and G. Xing, “VI-Eye: semantic-based 3d point cloud registration for infrastructure-assisted autonomous driving,” in *The 27th Annual International Conference on Mobile Computing and Networking*, 2021, pp. 573–586.

[4] N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice, “University of michigan north campus long-term vision and Lidar dataset,” *The International Journal of Robotics Research*, vol. 35, no. 9, pp. 1023–1035, 2016.

[5] J. Jeong, Y. Cho, Y.-S. Shin, H. Roh, and A. Kim, “Complex urban dataset with multi-level sensors from highly diverse urban environments,” *The International Journal of Robotics Research*, vol. 38, no. 6, pp. 642–657, 2019.

[6] D. Lee, S. Ryu, S. Yeon, Y. Lee, D. Kim, C. Han, Y. Cabon, P. Weinzaepfel, N. Guérin, G. Csurka *et al.*, “Large-scale localization datasets in crowded indoor spaces,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3227–3236.

[7] E. Spera, A. Furnari, S. Battiato, and G. M. Farinella, “EgoCart: a benchmark dataset for large-scale indoor image-based localization in retail stores,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 4, pp. 1253–1267, 2019.

[8] B. Pfrommer, N. Sanket, K. Daniilidis, and J. Cleveland, “PennCOSYVIO: A challenging visual inertial odometry benchmark,” in *IEEE International Conference on Robotics and Automation*, 2017, pp. 3847–3854.

[9] S. Cortés, A. Solin, E. Rahtu, and J. Kannala, “ADVIO: An authentic dataset for visual-inertial odometry,” in *European Conference on Computer Vision*, 2018, pp. 419–434.

[10] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic *et al.*, “Benchmarking 6DoF outdoor visual localization in changing conditions,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8601–8610.

[11] D. Zuñiga-Noël, A. Jaenal, R. Gomez-Ojeda, and J. Gonzalez-Jimenez, “The UMA-VI dataset: Visual-inertial odometry in low-textured and dynamic illumination environments,” *The International Journal of Robotics Research*, vol. 39, no. 9, pp. 1052–1060, 2020.

[12] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, “The EuRoC micro aerial vehicle datasets,” *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.

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

[14] M. Helmberger, K. Morin, B. Berner, N. Kumar, G. Cioffi, and D. Scaramuzza, “The Hilti SLAM challenge dataset,” *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7518–7525, 2022.

[15] T. Qin, S. Cao, J. Pan, P. Li, and S. Shen, “VINS-Fusion: An optimization-based multi-sensor state estimator,” 2019. [Online]. Available: https://github.com/HKUST-Aerial-Robotics/VINS-Fusion

[16] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, “ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM,” *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[17] J. Delmerico and D. Scaramuzza, “A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots,” in *IEEE International Conference on Robotics and Automation*, 2018, pp. 2502–2509.

[18] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, “Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age,” *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[19] Y. Shu, Q. Xu, J. Liu, R. R. Choudhury, N. Trigoni, and V. Bahl, “Indoor location competition 2.0 dataset,” January 2021. [Online]. Available: https://www.microsoft.com/en-us/research/publication/indoor-location-competition-2-0-dataset/

[20] W. Wang, Y. Hu, and S. Scherer, “TartanVO: A generalizable learning-based VO,” *arXiv preprint arXiv:2011.00359*, 2020.

[21] Z. Teed and J. Deng, "Droid-SLAM: Deep visual SLAM for monocular, stereo, and RGB-D cameras," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[22] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother, "DSAC-Differentiable RANSAC for camera localization," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6684–6692.

[23] H. Taira, M. Okutomi, T. Sattler, M. Cimpoi, M. Pollefeys, J. Sivic, T. Pajdla, and A. Torii, "InLoc: Indoor visual localization with dense matching and view synthesis," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7199–7209.

[24] H. Liu, M. Jiang, Z. Zhang, X. Huang, L. Zhao, M. Hang, Y. Feng, H. Bao, and G. Zhang, "LSFB: A low-cost and scalable framework for building large-scale localization benchmark," in *IEEE International Symposium on Mixed and Augmented Reality Adjunct*, 2020, pp. 219–224.

[25] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2016.

[26] E. B. Olson, "Real-time correlative scan matching," in *IEEE International Conference on Robotics and Automation*, 2009, pp. 4387–4393.

[27] T. Whelan, M. Kaess, J. J. Leonard, and J. McDonald, "Deformation-based loop closure for large scale dense RGB-D SLAM," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 548–555.

[28] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D LiDAR SLAM," in *IEEE International Conference on Robotics and Automation*, 2016, pp. 1271–1278.

[29] X. Gao, S. Shen, L. Zhu, T. Shi, Z. Wang, and Z. Hu, "Complete scene reconstruction by merging images and laser scans," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 10, pp. 3688–3701, 2019.

[30] W. Huang, G. Zhang, and X. Han, "Dense mapping from an accurate tracking SLAM," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 6, pp. 1565–1574, 2020.

[31] L. Zhou, D. Koppel, and M. Kaess, "LiDAR SLAM with plane adjustment for indoor environment," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7073–7080, 2021.

[32] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *The 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2007, pp. 225–234.

[33] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—a modern synthesis," in *International Workshop on Vision Algorithms*. Springer, 1999, pp. 298–372.

[34] M. I. Lourakis and A. A. Argyros, "SBA: A software package for generic sparse bundle adjustment," *ACM Transactions on Mathematical Software (TOMS)*, vol. 36, no. 1, pp. 1–30, 2009.

[35] W. Tan, H. Liu, Z. Dong, G. Zhang, and H. Bao, "Robust monocular SLAM in dynamic environments," in *IEEE International Symposium on Mixed and Augmented Reality*, 2013, pp. 209–218.

[36] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[37] H. Liu, G. Zhang, and H. Bao, "Robust keyframe-based monocular SLAM for augmented reality," in *IEEE International Symposium on Mixed and Augmented Reality*, 2016, pp. 1–10.

[38] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *IEEE International Conference on Robotics and Automation*, 2007, pp. 3565–3572.

[39] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[40] R. Jin, J. Liu, H. Zhang, and X. Niu, "Fast and accurate initialization for monocular vision/INS/GNSS integrated system on land vehicle," *IEEE Sensors Journal*, vol. 21, no. 22, pp. 26 074–26 085, 2021.

[41] S. Lynen, T. Sattler, M. Bosse, J. A. Hesch, M. Pollefeys, and R. Siegwart, "Get out of my lab: Large-scale, real-time visual-inertial localization." in *Robotics: Science and Systems*, vol. 1, 2015.

[42] J. Liu, W. Gao, and Z. Hu, "Optimization-based visual-inertial SLAM tightly coupled with raw GNSS measurements," in *IEEE International Conference on Robotics and Automation*, 2021, pp. 11 612–11 618.

[43] J. Ma, K. Zhang, and J. Jiang, "Loop closure detection via locality preserving matching with global consensus," *IEEE/CAA Journal of Automatica Sinica*, 2022.

[44] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof, "From structure-from-motion point clouds to fast location recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 2599–2606.

[45] T. Sattler, B. Leibe, and L. Kobbelt, "Improving image-based localization by active correspondence search," in *European Conference on Computer Vision*. Springer, 2012, pp. 752–765.

[46] T. Sattler, T. Weyand, B. Leibe, and L. Kobbelt, "Image retrieval for image-based localization revisited," in *British Machine Vision Conference*, R. Bowden, J. P. Collomosse, and K. Mikolajczyk, Eds. BMVA Press, 2012, pp. 1–12. [Online]. Available: https://doi.org/10.5244/C.26.76

[47] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-supervised interest point detection and description," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 224–236.

[48] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler, "D2-Net: A trainable CNN for joint description and detection of local features," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8092–8101.

[49] M. Dai, J. Hu, J. Zhuang, and E. Zheng, "A transformer-based feature segmentation and region alignment method for UAV-view geo-localization," *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.

[50] H. Hu, H. Wang, Z. Liu, and W. Chen, "Domain-invariant similarity activation map contrastive learning for retrieval-based long-term visual localization," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 2, pp. 313–328, 2021.

[51] A. Kendall, M. Grimes, and R. Cipolla, "PoseNet: A convolutional network for real-time 6-DoF camera relocalization," in *IEEE International Conference on Computer Vision*, 2015, pp. 2938–2946.

[52] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz, "Geometry-aware learning of maps for camera localization," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2616–2625.

[53] T. Wang, Z. Zheng, C. Yan, J. Zhang, Y. Sun, B. Zheng, and Y. Yang, "Each part matters: Local patterns facilitate cross-view geo-localization," *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.

[54] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *IEEE International Conference on Computer Communications*, vol. 2, 2000, pp. 775–784.

[55] M. Youssef and A. Agrawala, "The horus WLAN location determination system," in *The 3rd International Conference on Mobile Systems, Applications, and Services*, 2005, pp. 205–218.

[56] C. van der Beets, R. Nieminen, and T. Schneider, "FAPRIL: Towards faster privacy-preserving fingerprint-based localization," *Cryptology ePrint Archive*, 2022.

[57] V. Pasku, A. De Angelis, G. De Angelis, D. D. Arumugam, M. Dionigi, P. Carbone, A. Moschitta, and D. S. Ricketts, "Magnetic field-based positioning systems," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 2003–2017, 2017.

[58] A. Solin, M. Kok, N. Wahlström, T. B. Schön, and S. Särkkä, "Modeling and interpolation of the ambient magnetic field by gaussian processes," *IEEE Transactions on robotics*, vol. 34, no. 4, pp. 1112–1127, 2018.

[59] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[60] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An accurate O(n) solution to the PnP problem," *International Journal of Computer Vision*, vol. 81, no. 2, p. 155, 2009.

[61] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 1280–1286.

[62] S. Kato, E. Takeuchi, Y. Ishiguro, Y. Ninomiya, K. Takeda, and T. Hamada, "An open approach to autonomous vehicles," *IEEE Micro*, vol. 35, no. 6, pp. 60–68, 2015.

[63] V. Usenko, N. Demmel, D. Schubert, J. Stückler, and D. Cremers, "Visual-inertial mapping with non-linear factor recovery," https://vision.in.tum.de/research/vslam/basalt, pp. 422–429, 2019.

[64] D. Tsai, S. Worrall, M. Shan, A. Lohr, and E. Nebot, "Optimising the selection of samples for robust lidar camera calibration," https://github.com/acfr/cam_lidar_calibration, pp. 2631–2638, 2021.

[65] G. Kim and A. Kim, "Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018, pp. 4802–4809.

[66] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018, pp. 4758–4765.

[67] J. Zhang and S. Singh, “LOAM: Lidar odometry and mapping in real-time.” in *Robotics: Science and Systems*, vol. 2, no. 9, 2014.

[68] S. Rusinkiewicz and M. Levoy, “Efficient variants of the ICP algorithm,” in *The 3rd International Conference on 3D Digital Imaging and Modeling*. IEEE, 2001, pp. 145–152.

[69] S. Agarwal, K. Mierle, and Others, “Ceres solver,” http://ceres-solver.org.

[70] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[71] H. Jégou, M. Douze, and C. Schmid, “Product quantization for nearest neighbor search,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp. 117–128, 2011.

[72] J. L. Schönberger, T. Price, T. Sattler, J.-M. Frahm, and M. Pollefeys, “A vote-and-verify strategy for fast spatial verification in image retrieval,” in *Asian Conference on Computer Vision*. Springer, 2016, pp. 321–337.

[73] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[74] J. Shi *et al.*, “Good features to track,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 1994, pp. 593–600.

[75] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid, “Aggregating local image descriptors into compact codes,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 9, pp. 1704–1716, 2011.

[76] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, “IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation.” Georgia Institute of Technology, 2015.

[77] H. Liu, M. Chen, G. Zhang, H. Bao, and Y. Bao, “ICE-BA: Incremental, consistent and efficient bundle adjustment for visual-inertial SLAM,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1974–1982.

[78] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A benchmark for the evaluation of RGB-D SLAM systems,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 573–580.

[79] F. Furrer, M. Fehr, T. Novkovic, H. Sommer, I. Gilitschenski, and R. Siegwart, “Evaluation of combined time-offset estimation and hand-eye calibration on robotic datasets,” in *Field and Service Robotics*. Springer, 2018, pp. 145–159.

[80] T. Qin, P. Li, and S. Shen, “VINS-Mono: A robust and versatile monocular visual-inertial state estimator,” *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

Haomin Liu received the master and Ph.D. degrees in computer science from Zhejiang University in 2009 and 2017. He is currently research director in SenseTime Research. His research interests include structure from motion, SLAM, and augmented reality.

Linsheng Zhao is currently a researcher at SenseTime, Hangzhou. He received the master degrees from NanKai University in 2018. His research interests include wireless positioning, sensor fusion, and augmented reality.

Zhen Peng is currently a senior researcher at SenseTime, Hangzhou. He received the master degrees in Control Science and Engineering from Zhejiang University in 2017. His research interests include SLAM, sensor fusion and augmented reality.

Weijian Xie received the master degree in computer science from Zhejiang University in 2017. He is currently a PhD student at Zhejaing University. He is also affiliated with SenseTime Research as senior researcher. His research interests include SLAM, and augmented reality.

Mingxuan Jiang is currently a researcher at SenseTime, Hangzhou. He received his master’s degree in Robotics and Computation from University College London, London, UK, in 2018. His main research interests include simultaneous localization and mapping and structure from motion.

Hongbin Zha received the B.E. degree from the Hefei University of Technology, China, in 1983 and the M.S. and Ph.D. degrees from Kyushu University, Japan, in 1987 and 1990, respectively. After working as a Research Associate in the Kyushu Institute of Technology, Japan, he joined Kyushu University, Japan, in 1991 as an Associate Professor. Since 2000, he has been a Professor at the Key Laboratory of Machine Perception (Ministry of Education), Peking University, China. His research interests include computer vision, digital geometry processing, and robotics. He has published more than 350 technical publications in journals, books, and international conference proceedings.

Hujun Bao is currently a professor in the Computer Science Department of Zhejiang University, and the former director of the state key laboratory of Computer Aided Design and Computer Graphics. His research interests include computer graphics, computer vision and mixed reality. He leads the mixed reality group in the lab to make a wide range of research on 3D reconstruction and modeling, real-time rendering and virtual reality, realtime 3D fusion and augmented reality. Many of these algorithms have been successfully integrated into the mixed reality system SenseMARS.

Guofeng Zhang now is a Professor at State Key Lab of CAD&CG, Zhejiang University. He received his BS and Ph.D. degrees in Computer Science from Zhejiang University, in 2003 and 2009, respectively. He received the National Excellent Doctoral Dissertation Award, the Excellent Doctoral Dissertation Award of China Computer Federation and the best paper award of ISMAR 2020. His research interests include structure from motion, SLAM, 3D reconstruction, augmented reality, video segmentation, and editing.