100-Phones: A Large VI-SLAM Dataset for Augmented Reality Towards Mass Deployment on Mobile Phones

Guofeng Zhang*, Jin Yuan*, Haomin Liu, Zhen Peng, Chunlei Li, Zibin Wang, Hujun Bao†

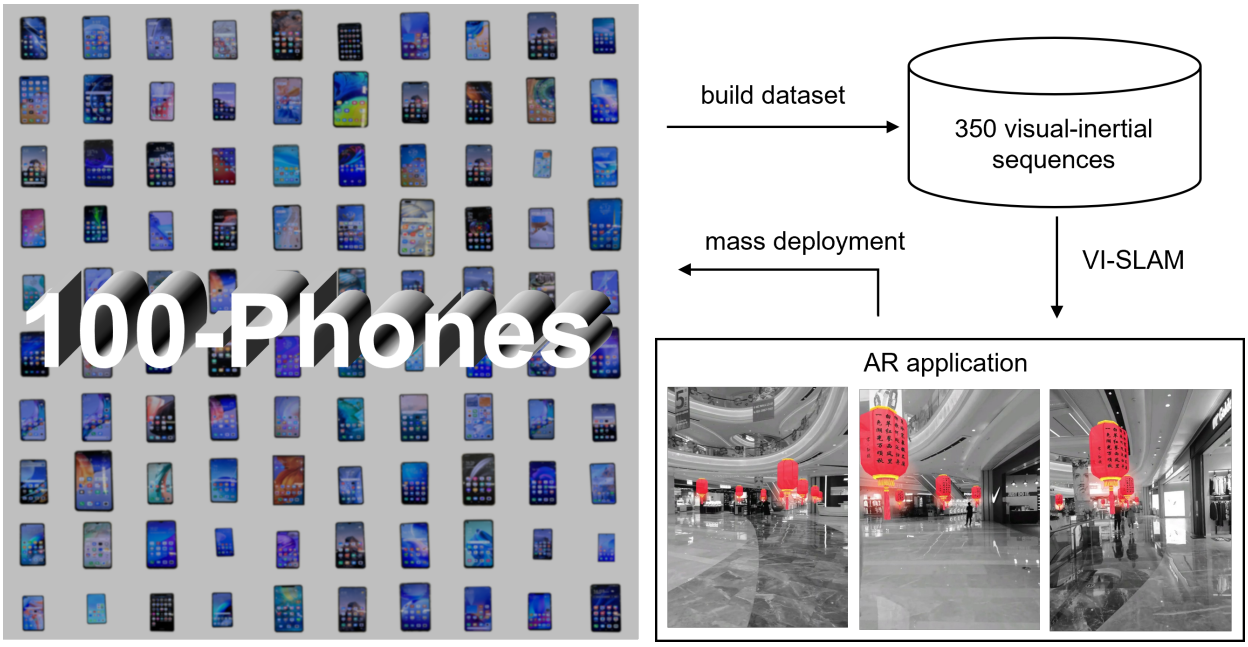


Fig. 1: We propose 100-Phones, a large visual-inertial dataset that consists of 350 sequences collected by 100 different models of phones, aiming to promote the mass deployment of VI-SLAM for AR on mobile phones.

Abstract—Visual-inertial SLAM (VI-SLAM) is a key technology for Augmented Reality (AR), which allows the AR device to recover its 6-DoF motion in real-time in order to render the virtual content with the corresponding pose. Nowadays, smartphones are still the mainstream devices for ordinary users to experience AR. However the current VI-SLAM methods, although performing well on high-end phones, still face robustness challenges when deployed on a larger stock of mid- and low-end phones. Existing VI-SLAM datasets use either very ideal sensors or only a limited number of devices for data collection, which cannot reflect the capability gaps that VI-SLAM methods need to solve when deployed on a large variety of phone models. This work proposes 100-Phones, the first VI-SLAM dataset covering a wide range of mainstream phones in the market. The dataset consists of 350 sequences collected by 100 different models of phones. Through analysis and experiments on the collected data, we conclude that the quality of visual-inertial data vary greatly among the mainstream phones, and the current open source VI-SLAM methods still have serious robustness issues when it comes to mass deployment on mobile phones. We release the dataset to facilitate the robustness improvement of VI-SLAM and to promote the mass popularization of AR. Project page: https://github.com/zju3dv/100-Phones.

Index Terms—Augmented Reality, Dataset, Benchmark, VI-SLAM

1 Introduction

Simultaneous Localization and Mapping (SLAM) is a technique that enables a mobile device to track its own 6 Degrees of Freedom (6-DoF) motion in an unknown environment, while simultaneously constructing the map of the environment. Visual-inertial SLAM (VI-SLAM) combines complementary visual and inertial measurements, achieving impressive results over the past decades [2, 20, 30, 38, 55]. Nowadays, VI-SLAM has become a standard technology for Augmented Reality (AR), enabling accurate recovery of the 6-DoF pose of AR device for rendering virtual content in alignment with the real-world environment.

In recent years, dedicated AR glasses have been developed offering excellent AR experiences, such as Microsoft HoloLens[1], Apple Vision Pro[2], and XReal Light[3]. However, these devices are still in an early stage and have not been widely adopted. The mainstream AR devices are still smartphones. With the built-in camera and IMU of the smartphone, ordinary users can also experience simple but practical AR applications, such as AR navigation, AR tourism, AR home decoration, etc. There are commercial products like Apple ARKit[4]

- *Guofeng Zhang and Hujun Bao are with the State Key Lab of CAD&CG, Zhejiang University. E-mail: {zhangguofeng, baohujun}@zju.edu.cn.*
- *Jin Yuan, Haomin Liu, Zhen Peng, Chunlei Li and Zibin Wang are with SenseTime. E-mail: {yuanjin1, liuhaomin, pengzhen1, lichunlei, wangzibin}@sensetime.com.*
- * *Equal contribution.*
- † *Corresponding author.*

[1] https://www.microsoft.com/en-us/hololens
[2] https://www.apple.com/apple-vision-pro/
[3] https://www.xreal.com/light/
[4] https://developer.apple.com/augmented-reality/arkit/

and Google ARCore[5] that provide VI-SLAM capabilities for Apple and Android phones respectively. While Apple phones have relatively consistent sensor characteristics, Android phones on the market differ widely, which poses great challenges for VI-SLAM methods. It remains a mystery how ARCore can achieve the mass deployment on the diverse Android phones. In the literature by contrast, the existing VI-SLAM methods only perform well on high-end phones, but poorly when deployed on a larger stock of mid- and low-end phones. There is no work to analyze the impact of different phone models, making it difficult for the emerging VI-SLAM methods to be widely applied. A suitable dataset is the key to fill this gap. However, the existing public VI-SLAM datasets use either very ideal camera and IMU [1, 45], or only a limited number of high-end phones [21, 26] for data collection, which cannot reflect the capability gaps that VI-SLAM methods need to solve for mass deployment on existing smartphones.

In this work, we propose 100-Phones, the first VI-SLAM dataset covering a wide range of mainstream phones in the market, as illustrated in Fig. 1. The specific contributions include:

- We propose a large VI-SLAM dataset that consists of 350 sequences collected by 100 different models of phones. The dataset is released to facilitate the improvement of VI-SLAM methods for mass deployment on mobile phones.
- We analyze the sensor characteristics of the 100 phone models, leading to the conclusion that the quality of visual-inertial data vary greatly among mainstream phones, which is challenging for VI-SLAM methods.
- We design the dataset into four subsets, each with a distinct motion pattern, to reflect different aspects of VI-SLAM methods. We also design two ground-truth solutions for small-scale and large-scale scenes respectively.
- We propose an evaluation metric that not only directly evaluates the AR effect in a goal-oriented way, but also unifies the traditional individual metrics into a single one to reflect the accumulating drift within a subsequence, enabling a concise analysis of VI-SLAM methods.
- We conduct experiments using four representative VI-SLAM systems, leading to the conclusion that the current VI-SLAM methods still have serious robustness issues when it comes to mass deployment on mobile phones.

2 Related Work

In this section, we first review the existing methods of visual-only SLAM and visual-inertial SLAM respectively, then review the existing visual-inertial datasets.

2.1 Visual-only SLAM

PTAM [18] is the pioneering real-time monocular SLAM system for AR. It utilizes front-end and back-end threads, where the front-end for camera tracking, and the back-end for environment mapping by Bundle Adjustment (BA) [52]. This framework is commonly adopted by subsequent visual SLAM systems [6, 9, 25, 32, 49]. Among them, ORB-SLAM [32, 33] utilizes ORB features [41] to improve accuracy and enable loop closure. SVO [9] and DSO [6] propose photometric-based direct methods to improve robustness of the feature-based indirect counterparts in low-texture environments. There are also deep learning-based methods [50, 51, 57] to further improve the robustness of the traditional geometry-based methods. Recently, with the rapid progress of Neural Radiance Fields (NeRF) [29, 31], the implicit representation has also been applied to visual SLAM for dense mapping [40, 58].

Despite decades of development that have significantly improved the robustness of visual SLAM, the visual-only methods still face robustness challenges in environments with poor texture, motion blur, defocus, etc. On the other hand, there are also robustness issues in scenarios with pure rotation or distant view, where there is lack of parallax for monocular camera of mobile phones to infer the 3D information [13]. Furthermore, monocular visual SLAM struggles to recover the true scale.

[5] https://developers.google.com/ar/

2.2 Visual-inertial SLAM

Visual-inertial SLAM (VI-SLAM) integrates the complementary IMU measurements to improve the robustness of visual SLAM and recover the true scale at the same time. In order to fuse visual and inertial measurements, the filter-based methods use inertial measurements for state prediction and visual measurements for state update. Representative works include MSCKF [30] and OpenVINS [12]. In contrast, the optimizer-based methods repeatedly re-linearize the visual and inertial factors to achieve superior accuracy, but at the expense of increased computational demands [4]. To improve the efficiency of optimization, OKVIS [20] introduces a sliding window of keyframes, where the oldest keyframe is marginalized upon the arrival of a new keyframe. VINS-Fusion [38, 39] implements the sliding window-based optimization in the front-end, and executes loop closure and 4-DoF pose graph optimization in the back-end to reduce error accumulation. It further supports temporal calibration that estimates the camera-IMU time offset that commonly exists in mobile phones. DM-VIO [53] also adopts the similar framework, but minimizes the photometric error instead of the re-projection error as the indirect methods. It further proposes a delayed marginalization to improves accuracy. ORB-SLAM3 [2] builds on the visual-only ORB-SLAM2 [33] which maintains a global map based on the visibility among keyframes. The global map-based ORB-SLAM3 achieves superior accuracy compared to the sliding window-based counterparts.

Incorporating IMU greatly improves the robustness of visual SLAM, however, the solution space is also increased. The gravity, scale, velocity and bias have to be additionally recovered, which requires sufficient motion excitation to ensure the observability [56]. The recovery usually needs an explicit initialization process, which has higher requirements for the visual reliability and the motion excitation [14, 34]. As a result, the integration of IMU introduces new robustness challenges. Furthermore, due to the substantial accumulation of IMU errors, if the visual information is unreliable for a long time, the robustness is still problematic. While there exist deep learning-based approaches to mitigate IMU errors [3, 27], the visual reliance cannot be eradicated. When it comes to mass deployment on mobile phones where different models have different error characteristics, these robustness issues will become more severe.

2.3 Visual-inertial Dataset

In the field of robotics, the most widely used visual-inertial datasets are KITTI [11] and EuRoC [1], which are collected by car and MAV respectively. They both adopts stereo cameras and high-quality IMU, with ground-truth obtained by high-precision GPS and motion capture (MoCap) system respectively. They are not suitable for AR, as AR devices are usually handheld or head-mounted which have different motion patterns from cars or MAVs. ZJU-SenseTime [21] and NEAR [54] are two AR-oriented visual-inertial datasets. The visual-inertial data are collected by handheld phones, and ground-truth poses are also obtained by MoCap system.

The MoCap system can achieve mm-level accuracy. However, it requires deploying expensive outside-in sensors in the environment, so it is difficult to be extended to large-scale scenes or outdoors. The TUM-VI dataset [45], although collects data in large-scale scenes, only provides ground-truth for parts the trajectories inside the room where the MoCap system was installed. PennCOSYVIO [37] overcomes this limitation by deploying visual tags in large-scale scenes, which are manually measured to provide visual constraints to recover ground-truth poses, achieving dm-level accuracy. NAVER LABS [19] and EgoCart [47] use natural features as visual constraints without any special deployments, and apply Structure from Motion (SfM) [43] to recover ground-truth poses with cm-level accuracy. However, the sensors are mounted on wheeled carts, which are not suitable for AR. LSFB [24, 26] proposes a visual-inertial dataset for AR in large-scale scenes. The visual map of the scene is pre-built by SfM, and each

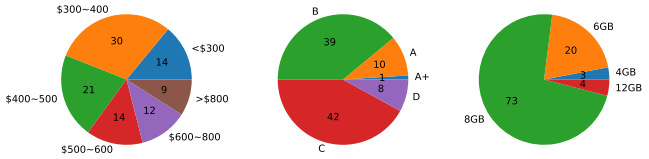
trajectory is recovered by combining visual localization in the pre-built map and the visual-inertial constraints, achieving cm-level accuracy. They use two mobile phones and an AR glasses to build the dataset for AR. Sarlin *et al.* [42] also employ a similar way to use heterogeneous AR devices in large scenes to construct a benchmark dataset called LaMAR.

While robots or AR glasses usually adopt global shutter fisheye cameras and high-quality IMUs, in this work, we focus on smartphones whose cameras are typically rolling shutter and pinhole with small Field of View (FoV), and IMUs are of lower quality. More importantly, different phone models may have large variations in their sensor characteristics. Existing visual-inertial datasets only cover one or a limited number of devices, making it impossible to reflect the robustness issues of VI-SLAM methods when deployed on a large variety of phone models. We propose a visual-inertial dataset covering 100 different phone models to fill this gap.

3 Proposed Dataset

Google Pixel 4a	Huawei P40	Realme GT	Vivo X60 Pro
Honor 20	Huawei P40 Pro	Samsung A60	Vivo Y51s
Honor 20 Pro	Huawei P40 Pro+	Samsung Galaxy Note10	Vivo Y52s
Honor 20s	Huawei P60	Samsung Galaxy S21 Ultra	Vivo Y73s
Honor 30	Meizu 17 Pro	Sharp AQUOS sense4 lite	Vivo Z5
Honor 30 Pro	Motorola Edge S	Sharp AQUOS sense4 plus	Vivo Z5i
Honor 30 Pro+	Motorola Moto G100	Vivo iQOO 7	Vivo Z6
Honor 30s	OnePlus 9	Vivo iQOO Neo	Xiaomi Mi 10
Honor 50	OnePlus 9 Pro	Vivo iQOO Neo5	Xiaomi Mi 10 Lite
Honor 80	Oppo A95	Vivo iQOO U3	Xiaomi Mi 10 Pro
Honor 8X	Oppo Find X2	Vivo iQOO Z1x	Xiaomi Mi 10s
Honor Magic5	Oppo Find X2 Pro	Vivo iQOO Z3	Xiaomi Mi 11
Honor V30 Pro	Oppo Find X3	Vivo iQOO5 Pro	Xiaomi Mi 11 Lite
Honor X10	Oppo K7	Vivo Nex	Xiaomi Mi 8
Huawei Mate 20	Oppo R17 Pro	Vivo S1 Pro	Xiaomi Mi 9
Huawei Mate 30 Pro	Oppo Reno	Vivo S10	Xiaomi Mi 9 Pro
Huawei Mate 50	Oppo Reno Ace	Vivo S6	Xiaomi Mi CC9
Huawei Mate X	Oppo Reno Ace 2	Vivo S7	Xiaomi Mi CC9 Pro
Huawei Mate X2	Oppo Reno Z	Vivo S7e	Xiaomi Mi Mix 3
Huawei Nova 10	Oppo Reno3	Vivo S7t	Xiaomi Redmi K30 Pro
Huawei Nova 2S	Oppo Reno4	Vivo X27	Xiaomi Redmi K40
Huawei Nova 3	Oppo Reno5	Vivo X30	Xiaomi Redmi K40 Pro
Huawei Nova 8	Oppo Reno5 A	Vivo X50 Pro	Xiaomi Redmi Note10 Pro
Huawei Nova 8 Pro	Oppo Reno5 K	Vivo X50 Pro+	ZTE Nubia Red Magic 6R
Huawei P30	Oppo Reno6	Vivo X60	ZTE Nubia Z30 Pro

Table 1: List of 100 phone models.



(a) Price (b) Processor level (c) Memory

Fig. 2: Distribution of (a) prices, (b) processor performance levels and (c) memory sizes among 100 phone models.The pie chart is annotated outside, while the numbers within the pie chart represent the number of phone models.

We select 100 mainstream Android phone models currently available on the market to build the dataset, and name it 100-Phones. Specifically, we first selected 80 mobile phones from the 560 most popular models since 2018, acquired from Tencent WeTest[6] who provides mobile compatibility testing services. Consider these 80 models are mainly domestic phones in China, we then added 10 popular international models based on their consumer ratings on Amazon. These popular models are generally moderately priced. So we further added 2 expensive (>$800) and 8 cheap (<$300) phones. During selection, we also considered the processor performance. The evaluation website NanoReview[7] conducts a comprehensive rating of mainstream smartphone processors and classifies them into five levels: A+, A, B, C, and D. We selected models ensuring the total number of A+, A, and B was equal to the total number of C and D. The 100 selected phone models are listed in Table 1, and the distribution of prices, processor performance levels, and memory sizes is shown in Fig. 2. Note that iPhones were not selected. This is due to the fact that all iPhones are equipped with high-quality sensors to support Apple ARKit. Existing VI-SLAM methods have already performed well on iPhones [22, 25]. Therefore, they fall beyond the scope of this work.

In this section, we describe the construction of the proposed dataset. We first calibrate the 100 phones, as detailed in Sect. 3.1. Then we analyze their sensor characteristics based on the calibration results, as reported in Sect. 3.2. We design the dataset into four subsets, each with a distinct motion pattern, to reflect different aspects of VI-SLAM methods. The first three subsets are collected in a small-scale scene using the 100 phones, and the fourth subset is collected in three large-scale scenes using five selected phones. The statistics of the four subsets are given in Table 2, and the details are presented in Sect. 3.3 and Sect. 3.4 for subsets in small-scale and large-scale scenes respectively. The images and IMUs are acquired with their timestamps using the Android API, specifically android.hardware.camera2 for images, and android.hardware.sensor.TYPE_GYROSCOPETYPE/TYPE_ACCELEROMETER for gyroscope/acceleration respectively. The image resolution is set to 640×480. During data acquisition, all other softwares were shut down to ensure that system resources would not be preempted.

sub-dataset	"circle"	"line"	"rotation"	"general"
#phones	100	100	100	5
#sequences	100	100	100	50
#frames	170K	122K	169K	322K
scale	2m radius	10m line	stand still	$\sim$ Km^2

Table 2: Statistics of four sub-datasets.

3.1 Calibration

For each of the 100 phones, we use Kalibr [10, 36] to calibrate the camera parameters and the camera-IMU parameters. For the sake of completeness and to facilitate subsequent discussions, we give a brief description of the calibration process proposed in [10]. The involved notations and equations will be used in the subsequent analysis of sensor characteristics.

Camera calibration is performed by capturing an AprilTag [35] at different viewpoints, and recover the camera intrinsic parameters K together with the camera pose T_i for each image i by

$$\arg\min_{K,T_i} \sum_{ij} ||f(K,T_i,X_j) - x_{ij}||^2, \tag{1}$$

where $f(K,T,X)$ projects a 3D point X to the image by camera intrinsic K and extrinsic T. X_j is the j-th 3D point of AprilTag and x_{ij} is its 2D observation detected on i-th image.

Camera-IMU calibration is performed by capturing another sequence of the AprilTag in which the phone is moved to excite all IMU axes. As proposed in [10], the camera-IMU extrinsic T_{CI} and time offset Δt, together with the IMU pose $\Phi_T(t)$ and IMU bias $b(t)$ at continuous time t, are recovered by

$$\begin{aligned} \arg\min_{T_{\text{CI}},\Delta t,\Phi_T(t),b(t)} & \sum_{ij} ||f(K, T_{\text{CI}} \circ \Phi_T(t_i + \Delta t), X_j) - x_{ij}||^2_{\sigma_x} \\ & + \sum_k ||\Phi_\omega(t_k) - \omega_k||^2_{\sigma_\omega} + \sum_k ||\Phi_a(t_k) - a_k||^2_{\sigma_a} \\ & + \int ||\dot{b}(t)||^2_{\sigma_w} dt, \end{aligned} \tag{2}$$

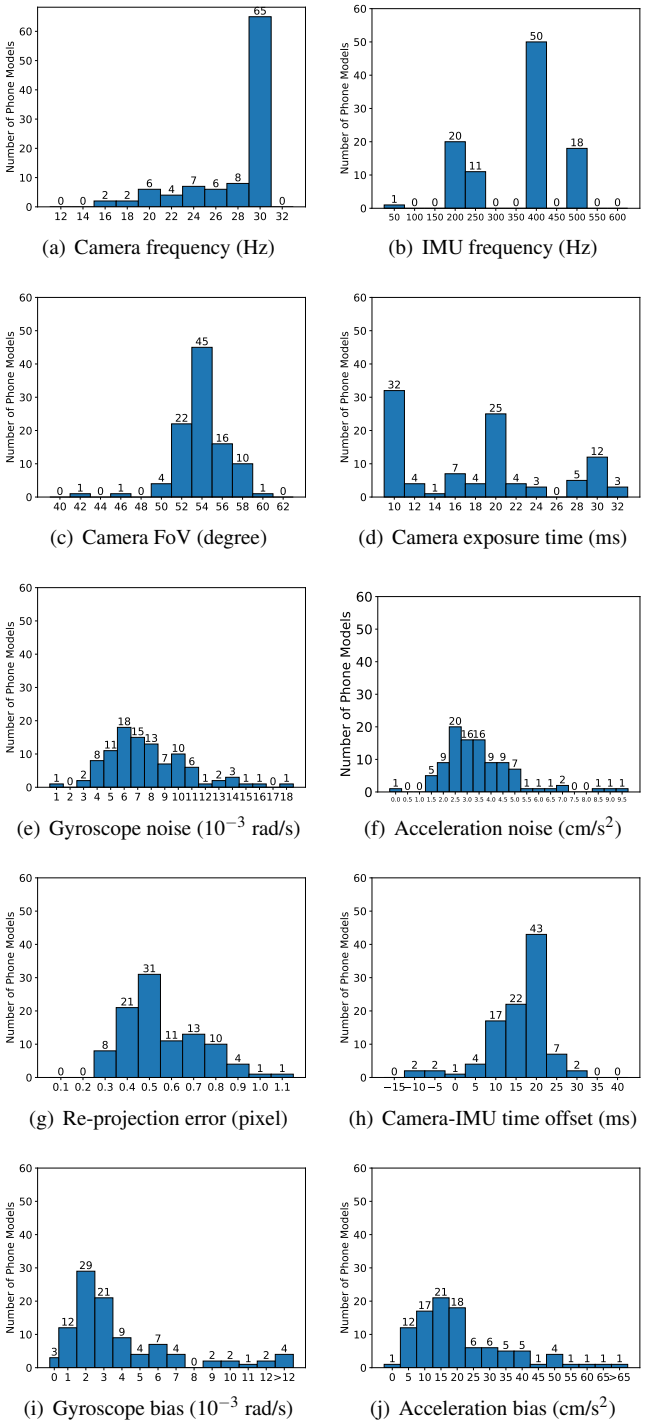
where ω_k and a_k are respectively the IMU gyroscope and acceleration measurement at timestamp t_k. The IMU pose trajectory is represented by B-spline, such that its derivatives can be expressed analytically.

[6] https://wetest.qq.com/products/cloud-phone
[7] https://nanoreview.net/en/soc-list/rating

$\Phi_\omega(t)$ and $\Phi_a(t)$ are the derived angular velocity and linear acceleration respectively. σ_x, σ_ω, σ_a and σ_w are variance of noise for image point, gyroscope and acceleration measurement, and bias random walk respectively. Details are referred to [10].

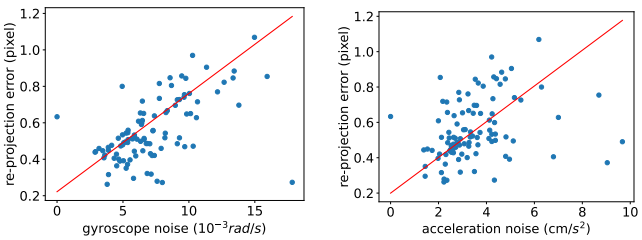
3.2 Sensor Characteristics

We select ten sensor metrics that are closely related to VI-SLAM to analyze the sensor characteristics of the 100 phone models. For each metric, we calculate the distribution of its value among the 100 phone models, as shown in Fig. 3.

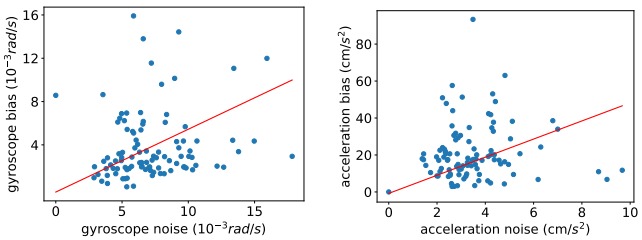


(a) Camera frequency (Hz)
(b) IMU frequency (Hz)
(c) Camera FoV (degree)
(d) Camera exposure time (ms)
(e) Gyroscope noise (10^{-3} rad/s)
(f) Acceleration noise (cm/s^2)
(g) Re-projection error (pixel)
(h) Camera-IMU time offset (ms)
(i) Gyroscope bias (10^{-3} rad/s)
(j) Acceleration bias (cm/s^2)

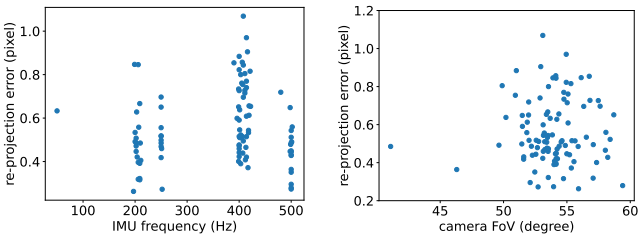
Fig. 3: Distribution among 100 phone models for the ten sensor metrics (a)∼(j).



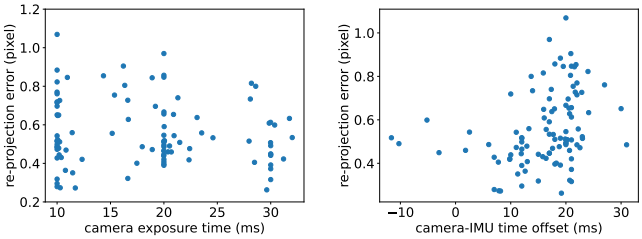
(a) Re-proj. error w.r.t gyr. noise
(b) Re-proj. error w.r.t acc. noise



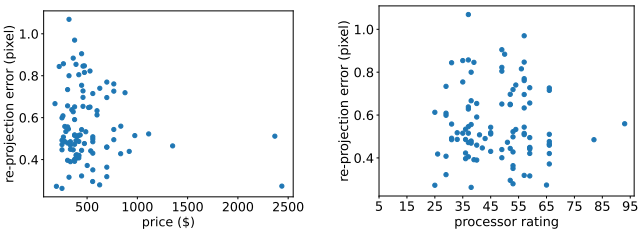
(c) Gyr. bias w.r.t gyr. noise
(d) Acc. bias w.r.t acc. noise



(e) Re-proj. error w.r.t IMU freq.
(f) Re-proj. error w.r.t FoV



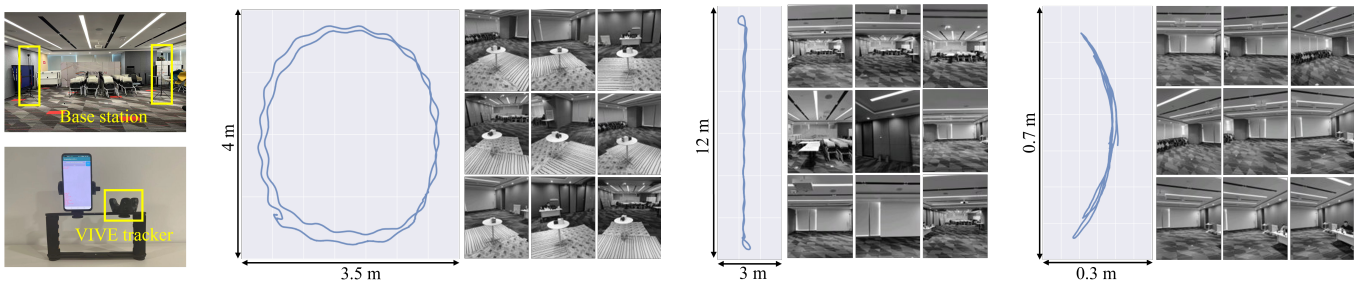
(g) Re-proj. error w.r.t expo.
(h) Re-proj. error w.r.t time offset



(i) Re-proj. error w.r.t price
(j) Re-proj. error w.r.t proc. rating

Fig. 4: Correlation between the selected pairs of characteristics (a)∼(j). One blue dot corresponds to one phone model. A red line is fitted by RANSAC algorithm [7] for (a)∼(d).

The metric of camera frequency and IMU frequency are respectively the number of visual and inertial measurements VI-SLAM can acquire per second. Usually the more measurements per second, the more beneficial for the state estimation of VI-SLAM. As shown in Fig. 3 (a) and (b), camera frequency is relatively consistent while IMU frequency varies among the 100 phones. In terms of camera FoV and exposure time, they both affect the robustness of VI-SLAM. With a larger FoV,



(a) SteamVR (b) "Circle" (c) "Line" (d) "Rotation"

Fig. 5: Dataset in small-scale scene. (a) shows the devices of SteamVR tracking 2.0 to obtain ground-truth. (b)(c)(d) are the collected data for three sub-datasets, each with ground-truth trajectory on the left and representative images on the right.

more features can be observed in a single frame, and more common features can be matched between different frames, which are both beneficial for VI-SLAM. On the other hand, exposure time affects VI-SLAM by the image quality. Too short exposure time will make the image too dark, while too long exposure time will easily cause motion blur, which are both challenging to VI-SLAM. As shown in Fig. 3 (c) and (d), FoV is relatively consistent while exposure time varies among the 100 phones. It can also be seen that compared with the global shutter fisheye camera of the EuRoC dataset [1], the camera of mobile phone is more challenging to VI-SLAM, as the FoV is smaller, and it is more prone to motion blur. The remaining metrics are obtained from camera-IMU calibration described in Sect. 3.1. The metrics of re-projection error, gyroscope noise and acceleration noise are directly obtained from the first three error terms in Eq. (2). These error terms can reflect the noise level of visual and inertial measurements to some extent. Camera-IMU time offset is the resulting Δt, and bias are the resulting $\frac{1}{n}\sum_{i=1}^{n} b(t_i)$. Since VI-SLAM needs to estimate them online, the larger time offset or bias exist in the raw data, the more challenge for online estimation. As shown in Fig. 3 (e)∼(j), image noise is relatively consistent, while IMU noise, bias and camera-IMU time offset vary greatly among the 100 phones.

Intuitively, high-end phones have better image and IMU quality than low-end phones in all aspects. To verify, we further investigate whether there is any correlation between different sensor metrics. As shown in Fig. 4, we select pairs of sensor metrics and observe how one metric varies with the other, using the metric values of the 100 phone models, one phone corresponding to one dot in the figure. We observe a rough linear correlation among re-projection error, gyroscope noise and acceleration noise, as shown in Fig. 4 (a) and (b). This is partially due to the fact that they come from the three error terms of Eq. (2) in the first place, which are naturally balanced during optimization. Nevertheless, it reflects to some extent that the quality is a coarsely consistent among visual and inertial measurements. For phones with smaller values of these three metrics, the visual and inertial measurements simultaneously fit the calibration model better, than the phones with larger values. We also observe a coarse linear correlation between the IMU noise and IMU bias, as shown in Fig. 4 (c) and (d), which indicates that the phones with better data quality usually have IMU biases adjusted to a lower level. For other metrics, we do not find obvious correlations as shown in Fig. 4 (e)∼(h). For example, the phones with higher sensor quality, may have lower IMU frequency or larger camera-IMU time offset. We also analyze the price and the processor performance rating in Fig. 4 (i) and (j) respectively, and do not find obvious correlations either. This makes it more complicated to deploy a SLAM method to different models of phones with sensor characteristics that may vary randomly.

3.3 Dataset in Small-scale Scene

We use each phone to collect three simple yet typical motions in a small scene, creating three sub-datasets, each containing 100 sequences corresponding to the 100 phones:

- "Circle": The operator is walking around a 2m radius circle twice, as shown in Fig. 5(b). This is the simplest sub-dataset, in which most feature points have sufficient visual parallax, and IMU experiences sufficient motion excitation along all axes.
- "Line": The operator is walking forth and back along a 10m line, as shown in Fig. 5(c). This is a typical motion during AR navigation. It presents greater challenges compared to "circle", because IMU experiences less motion excitation along the straight line, and feature points quickly leave the field of view when turning around.
- "Rotation": The operator is rotating in place for one minute, as shown in Fig. 5(d). This is a typical motion pattern during AR tourism, where the user stands still and rotates the phone to observe the surrounding AR effects. Although it is not a pure rotation, the parallax of the feature points is much smaller than the previous two sub-datasets, making it the most challenging one for visual-based methods.

We choose SteamVR Tracking 2.0 to obtain ground-truth poses for its portability and cm-level accuracy in small scenes [15]. Specifically, we place four SteamVR base stations[8] in the scene, and rigidly mount the phone and VIVE tracker[9] together, as shown in Fig. 5(a). We follow the method of [21] to temporally and spatially align the ground-truth trajectory of VIVE tracker to that of IMU. For temporal alignment, we estimate the VIVE-IMU time offset Δt by

$$\arg\max_{\Delta t} \frac{\sum_t ||\theta_{\text{VIVE}}(t)||||\theta_{\text{IMU}}(t+\Delta t)||}{\sqrt{\sum_t ||\theta_{\text{VIVE}}(t)||^2}\sqrt{\sum_t ||\theta_{\text{IMU}}(t+\Delta t)||^2}}, \tag{3}$$

where $\theta_{\text{VIVE}}(t)$ and $\theta_{\text{IMU}}(t)$ are relative rotation between timestamps $(t, t+1s)$ in the trajectory of VIVE tracker and IMU respectively. For spatial alignment, we capture images of an AprilTag [35] at different viewpoints, and recover the VIVE-camera extrinsic T_{CV} by

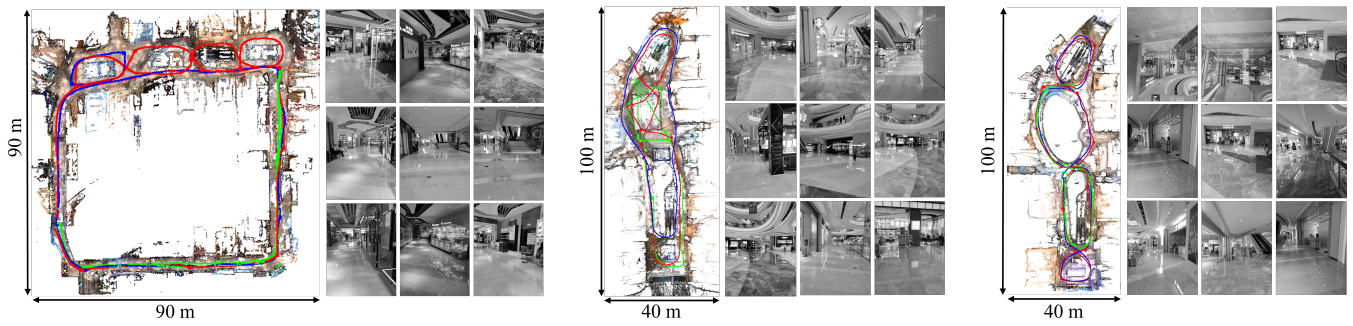
$$\arg\min_{T_{\text{CV}}} \sum_{ij} ||f(T_{\text{CV}} \circ T_{V_i}, X_j) - x_{ij}||^2, \tag{4}$$

where $f(T,X)$ projects a 3D point X to the image by camera pose T. T_{V_i} is the ground-truth pose of VIVE tracker at the time of i-th image. X_j is the j-th 3D point of AprilTag and x_{ij} is its 2D observation detected on i-th image.

3.4 Dataset in Large-scale Scenes

The first three sub-datasets mainly focus on the robustness to different phone models. We design the fourth sub-dataset for the robustness to large-scale scenes and general motions. We select five phones and collect data in three large-scale scenes. Each phone collects ten sequences. Each sequence has 2K∼15K frames, and moves 70∼500m, resulting

[8] https://www.vive.com/us/accessory/base-station2/
[9] https://www.vive.com/us/accessory/tracker3/



(a) (b) (c)

Fig. 6: Dataset collected in the three large-scale scenes (a)(b)(c). For each scene, the reconstructed visual map is shown, overlaid with three recovered trajectories in different colors (red, green and blue). The representative images are shown on the right.

	Vivo iQOO Z3	Vivo S7t	Xiaomi Mi 9	Vivo iQOO 7	Huawei Mate 30 Pro
price ($)	<300	300∼400	400∼500	500∼600	>800
processor perf. level	C	C	C	A	B
cam. freq. (Hz)	30	27	21	30	24
IMU freq. (Hz)	202	202	411	417	480
cam. FoV (degree)	53.41	54.09	49.89	55.35	51.98
cam. expo. (ms)	30	20	16	30	10
gyr. noise (10^{-3} rad/s)	5.38	6.51	8.68	5.34	6.47
acc. noise (cm/s^2)	2.91	2.50	3.86	9.04	2.18
gyr. bias (10^{-3} rad/s)	5.43	2.02	1.62	6.94	1.96
acc. bias (cm/s^2)	12.14	11.97	17.11	6.78	12.71
re-proj. error (pixel)	0.51	0.44	0.80	0.37	0.72
time offset (ms)	36	27	29	36	15

Table 3: The price and sensor characteristics of the 5 selected phones to collect dataset in large-scale scenes.

in 50 sequences of this sub-dataset. The selection of the five phones is based on their prices and processor performance, as listed in Table 3. Note that except for Vivo iQOO 7 with a relatively large acceleration noise, other sensor metrics are distributed in the common range.

We use the method proposed in [26] to obtain ground-truth in the large-scale scenes. The method is based on reconstructing an accurate visual map of the large-scale scene. For each visual-inertial sequence captured in it, the ground-truth poses are obtained by joint optimization taking both the visual localization in the pre-built map and visual-inertial constraints. The visual map is reconstructed by SfM [43] using a panoramic camera Insta360 ONE RS[10]. For each image acquired by mobile phone, the features of SuperPoint [5] are extracted and matched against the pre-built visual map by the method of [17,44]. KLT [46] and SIFT [28] features are also extracted and matched among the acquired images. The obtained feature correspondence is denoted as (X_j, x_{ij}), where X_j is the j-th 3D point and x_{ij} is the corresponding 2D point in i-th image. The joint optimization is formulated as

$$\underset{M_i, X_j \in \mathscr{X}}{\arg\min} \sum_{ij} ||f(T_i, X_j) - x_{ij}||^2 + \sum_i h(M_i, M_{i+1}), \tag{5}$$

where $M_i = (T_i, v_i, b_i)$ is the motion state for image i, and T_i, v_i and v_i are camera pose, IMU velocity and IMU bias respectively. $\mathscr{X}$ is the set of 3D points of the KLT and SIFT features. $f(T,X)$ projects a 3D point X to the image by camera pose T. $h(M_i, M_j)$ is the IMU cost function evaluating the deviation of motion states (M_i, M_j) and IMU measurements between image (i, j) by IMU pre-integration [8]. Eq. (5) is efficiently solved by ICE-BA [23].

Fig. 6 shows the reconstructed visual map for the three scenes, overlaid with three representative trajectories on each scene. Thanks to the accurate visual map obtained by SfM, the recovered trajectory has cm-level accuracy as reported in [26]. This accuracy is enough to evaluate current VI-SLAM methods, which usually achieve only m-level accuracy in large-scale scenes.

[10] https://www.insta360.com/cn/product/insta360-oners

4 Evaluation Metrics

We use the Absolute Positional / Rotational Error (APE/ARE) and the Relative Positional / Rotational Error (RPE/RRE) that are commonly used in the literature [4, 21, 48] to evaluate the performance of VI-SLAM. These metrics are defined as

$$\begin{aligned}
\varepsilon_{\text{APE}} &= \sqrt{\frac{1}{n}\sum_{i=1}^{n} ||p_i - p_i^*||^2}, \\
\varepsilon_{\text{ARE}} &= \sqrt{\frac{1}{n}\sum_{i=1}^{n} ||\log(R_i^{-1} R_i^*)||^2}, \\
\varepsilon_{\text{RPE}} &= \sqrt{\frac{1}{n-1}\sum_{i=1}^{n-1} ||(p_{i+1} - p_i) - (p_{i+1}^* - p_i^*)||^2}, \\
\varepsilon_{\text{RRE}} &= \sqrt{\frac{1}{n-1}\sum_{i=1}^{n-1} ||\log((R_{i+1}^{-1} R_i)^{-1}((R_{i+1}^*)^{-1} R_i^*))||^2},
\end{aligned} \tag{6}$$

where $T_i = (p_i, R_i)$ is the i-th pose of SLAM comprised of the position p_i and the rotation matrix R_i. The superscript * denotes the corresponding ground-truth pose. The trajectory of SLAM is aligned to ground-truth by SE(3) before calculating APE/ARE. Following [2, 26], we also calculate the Scale Error (SE) as

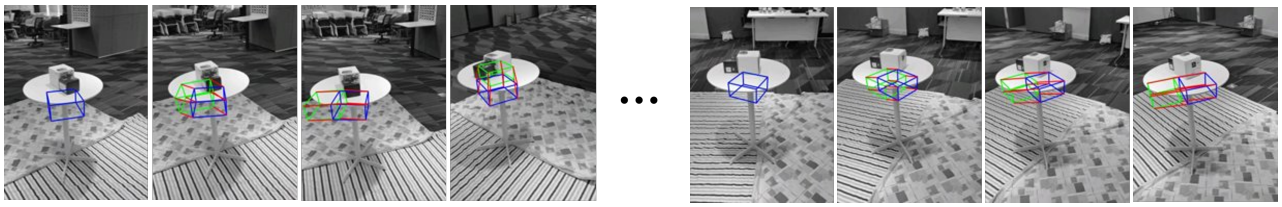
$$\varepsilon_{\text{SE}} = |s - 1|, \tag{7}$$

where s is the scale estimate from Sim(3) alignment. We distinguish APE after SE(3) and Sim(3) alignment as two different metrics.

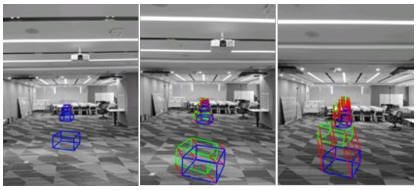
These traditional metrics cannot directly reflect how stable the AR effect is seen by the user. For example, the user is more likely to perceive the positional error of the virtual objects near them, and the rotational error of the objects far away. Moreover, using multiple individual metrics is not conducive to analyzing the overall performance of VI-SLAM methods. For instance, one VI-SLAM method may have a smaller positional error than another method, but a larger rotational error at the same time, making it difficult to judge which method is better. To address these issues, we propose a metric called AR Deviation (ARD) to measure the gap between rendered virtual cubes using the pose from SLAM and from ground-truth. ARD is calculated as

$$\varepsilon_{\text{ARD}} = \sqrt{\frac{1}{mn}\sum_{i=1}^{n}\sum_{j=1}^{m} ||f(T_i, X_j) - f(T_i^*, X_j)||^2}, \tag{8}$$

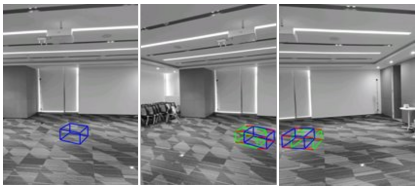
where $f(T,X)$ projects a 3D vertex X of the virtual cube to the image by camera pose T. Similar idea has been proposed in [16]. The difference



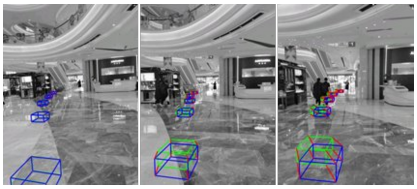
(a) "Circle"



(b) "Line"



(c) "Rotation"



(d) "General"

Fig. 7: Illustration of AR deviation for the four sub-datasets (a)∼(d). The green cube is the ground-truth, and the blue cube is the evaluated one. The AR deviation is represented by the red lines that connect the corresponding vertices of the two cubes.

is that our aim is to reflect the accumulating error within a subsequence, rather than the absolute pose error of a single frame. Specifically, instead of aligning the whole trajectory, we align the coordinate system every 10 seconds. We denote the pose at the alignment moment as T_i. In the next 10 seconds, we transform each pose T_k of k-th frame and the object vertices X_j to the coordinate system of i-th frame by

$$\begin{aligned} T_k' &= T_k \circ T_i^{-1} \\ T_k^{*\prime} &= T_k^* \circ (T_i^*)^{-1} \\ X_j' &= T_i^* \circ X_j \end{aligned} \tag{9}$$

Fig. 7 illustrates the process. Green and blue cubes represent the AR effect rendered using the pose of SLAM and ground-truth respectively. Red lines between the vertices of green and blue cubes represent the AR deviation. The errors are reset every 10 seconds. See the fifth image of Fig. 7(a) for example, where the green and blue cubes coincide. We place different numbers and positions of virtual cubes for different sub-datasets. Specifically, for "circle", we place one virtual cube at the center of the circular trajectory. For "line", we place one virtual cube every 2m on the straight-line trajectory. For "rotation", we place one virtual cube at 2m in front of the first frame. For "general", we place a virtual object every 2m along the 10m trajectory in front of each current frame. See Fig. 7 and the supplementary video for details.

Finally, the metrics in Eq. (6) and (8) are calculated by Root Mean Squared Error (RMSE). If the error is drifting abnormally for certain frames, which frequently happens on low-end phones, the error will overwhelm the rest of the sequence. Therefore, we treat such frames as localization failures, remove them from the RMSE calculation, and count the failure rate as another evaluation metric. We consider a frame as failure if its RPE/RRE is larger than 0.2m/6°, or its ARD is large than 400 pixels. Frames before completion of visual-inertial initialization are also considered as failures.

5 Experiments

	VINS-Fusion	DM-VIO	ORB-SLAM3	OpenVINS
direct/indi.	indirect	direct	indirect	indirect
solver	optimizer	optimizer	optimizer	filter
map	sliding win.	sliding win.	global map	sliding win.
temp. calib.	yes	no	no	yes
loop closure	yes	no	yes	no

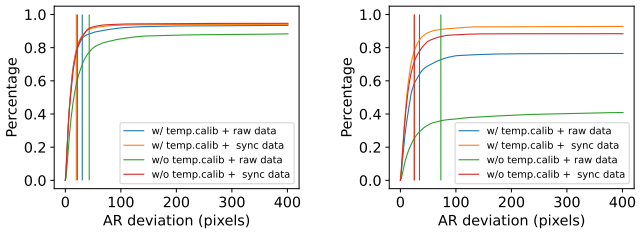
Table 4: Four representative VI-SLAM systems.

We select four representative monocular VI-SLAM systems to conduct experiments: VINS-Fusion [38, 39], DM-VIO [53], ORB-SLAM3 [2], and OpenVINS [12]. They have been evaluated on previous datasets EuRoC [1] and TUM-VI [45] in their papers, demonstrating that they represent the current state-of-the-arts. In Table 4, we compare five important aspects that can characterize different VI-SLAM systems. DM-VIO is the only photometric-based direct method while others are feature-based indirect methods. OpenVINS is the only filter-based method while others are optimizer-based methods. ORB-SLAM3 is the only one that maintains a globally consistent map while others only maintain a sliding window-based local map. VINS-Fusion is the only one that supports both temporal calibration of camera-IMU time offset and loop closure. In this section, we use the proposed 100-Phones dataset to evaluate these systems. We evaluate the front-end poses rather than the optimized poses, because AR needs to obtain poses in real-time.

Since no previous work has evaluated VI-SLAM systems using such a large number of devices, we first need to determine the algorithm configuration for the diverse sensor characteristics, specifically, how to handle the camera-IMU time offset, and how to set the IMU noise parameters. We conduct corresponding experiments to analyze the effect of time offset and IMU noise parameters on the performance of VI-SLAM in Sect. 5.1 and Sect. 5.2 respectively. For the sake of conciseness, both experiments do not use the full dataset or use all evaluation metrics. The full evaluation is given in Sect. 5.3.

5.1 Effect of Time Offset

Although we have calibrated the camera-IMU time offset for the 100 phones in our dataset, it is not feasible to pre-calibrate every phone in practice when it comes to mass deployment. Moreover, it is unclear whether the pre-calibrated time offset will change over time. To analyze the impact of time offset, we employ the two systems VINS-Fusion and OpenVINS which support online temporal calibration. We select the simplest "circle" dataset, which has sufficient parallax and motion excitation such that the observability for time offset estimation is guaranteed. We compare each system with and without temporal calibration. We further compare results using raw data and using the synchronized data by the pre-calibrated time offset. The combination of the two algorithmic configurations and the two data configurations results in four configurations. We choose AR deviation defined in Eq. (8) as the evaluation metric. In contrast to previous works that only use RMSE to compare accuracy, we propose to use the Cumulative Distribution Function (CDF) to compare robustness simultaneously. A point (x, y) on the CDF curve denotes the probability of having an error less than x is y. The curve above has better accuracy and robustness than the curve below. At the convergence point on the rightmost curve, $1 - y$ is the failure rate. The results are shown in Fig. 8, where the CDF curves and the traditional RMSE as the vertical lines are both shown.



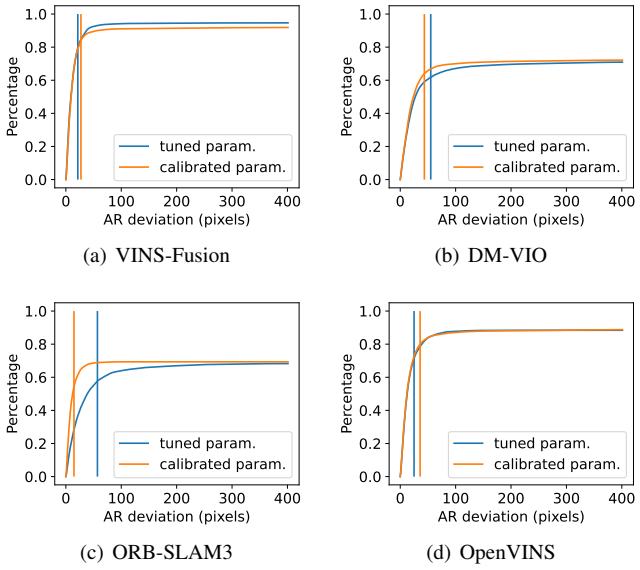
(a) VINS-Fusion (b) OpenVINS

Fig. 8: Effect of time offset on (a) VINS-Fusion and (b) OpenVINS. Curve represents CDF and vertical line represents RMSE of AR deviation.

Comparing VINS-Fusion and OpenVINS, the gaps among the four configurations of VINS-Fusion is much smaller than that of OpenVINS, indicating that VINS-Fusion is more robust to time offset. But the conclusions are consistent between the two systems. If time offset is not handled (green curve), the result is the worst. Either online calibration (blue curve) or offline synchronization (red curve) can significantly improve the result. There is still a gap between blue and red curves, indicating that there is still room for improvement of online calibration. If time offset has been synchronized offline, the results of VINS-Fusion with and without temporal calibration (yellow and red curve respectively) are very close, which reflects to some extent that the offline calibrated time offset does not change much over time. By contrast, the yellow curve is higher than the red curve by OpenVINS. We hypothesize that temporal calibration, although not necessary, may still be beneficial by involving redundant degrees of freedom for systems that are not robust enough.

This experiment suggests that the time offset must be properly handled. Considering the fact that the other two systems DM-VIO and ORB-SLAM3 do not support temporal calibration, we use the synchronized data for the following experiments.

5.2 Effect of IMU Parameters



(a) VINS-Fusion (b) DM-VIO

(c) ORB-SLAM3 (d) OpenVINS

Fig. 9: Effect of IMU parameters on the four systems (a)∼(d). Curve represents CDF and vertical line represents RMSE of AR deviation.

To run VI-SLAM, it is necessary to provide the IMU measurement noise and bias random walk noise as algorithm parameters. Existing datasets are collected with only one or very few devices, allowing for parameter tuning tailored to each dataset. However, this approach may not be suitable for our dataset, where the IMU noise among the 100 different models of phones varies greatly, as presented in Sect. 3.2. We conduct an experiment to analyze whether a unified set of tuned parameters can be generalized to all phone models in our dataset. We run all systems on the "circle" dataset using AR deviation defined in Eq. (8) as the metric, to compare the two configurations:

- Tuned parameters: We use the default parameters of VINS-Fusion tuned on EuRoC as the unified parameters for all phones. We find that this set of parameters performs better than other default parameters on all systems overall.
- Calibrated parameters: We use the IMU measurement noise and bias random walk noise obtained from the results of camera-IMU calibration as the IMU parameters set for each phone separately.

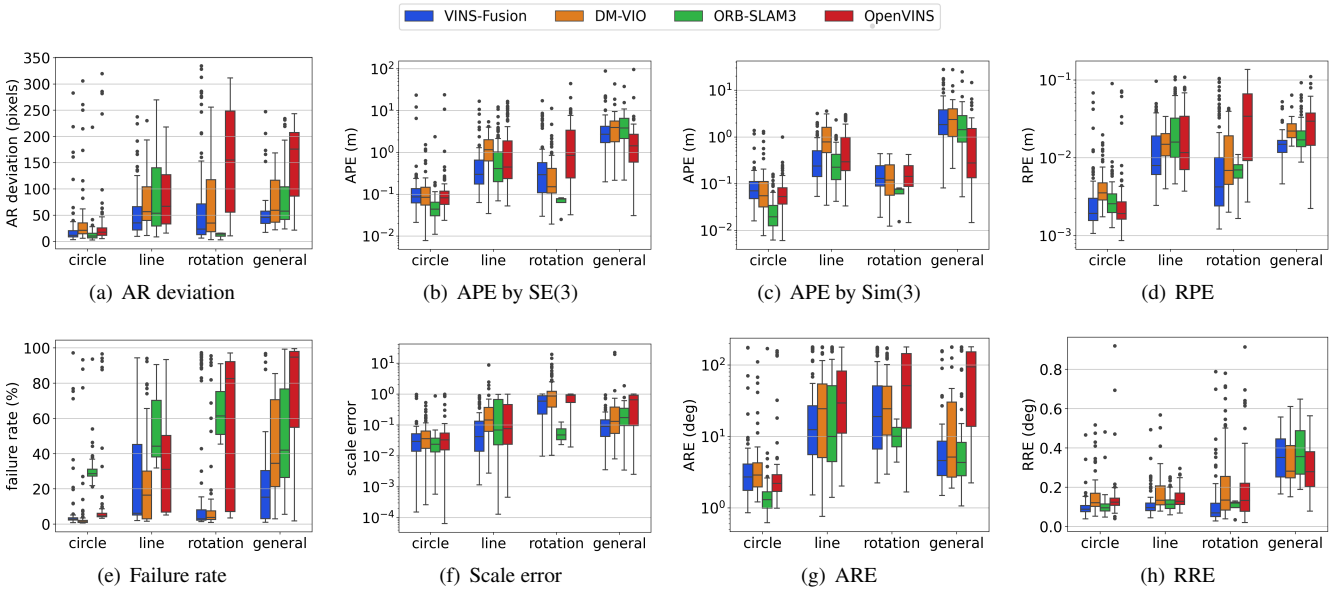
The results are shown in Fig. 9. For VINS-Fusion, the tuned parameters work slightly better in both accuracy and robustness. This is partly because they are originally tuned for VINS-Fusion. On the contrary, for DM-VIO and ORB-SLAM3, the calibrated parameters have better accuracy and robustness than the tuned parameters, and gap for ORB-SLAM3 is significant. For OpenVINS, the tuned parameters have slightly higher accuracy, and similar robustness compared to the calibrated parameters. Overall, the calibrated parameters perform better. So we use the calibrated parameters for the following experiments. More sophisticated strategies that consider both the calibration results and the algorithm characteristics can be further explored in the future.

5.3 Full Evaluation

We run all systems on the entire dataset using all the eight evaluation metrics introduced in Sect. 4. The results are shown in Fig. 10. To better illustrate the error distribution, we select boxplot to plot the results, where the median, the quartiles and the outliers are shown for each metric on each sub-dataset.

For the first three sub-datasets, all the 100 sequences in the same sub-dataset follow the same motion in the same scene. The performance variation comes from different models of phones. On the simplest "circle" dataset, the results are relatively consistent, but there are many outliers. On the other two datasets "line" and "rotation", the boxes are much taller, reflecting the performance variations among different models of phones. We found two main robustness issues for these two datasets. On the one hand, it is difficult to successfully accomplish visual-inertial initialization on the low-end phones, resulting in high failure rates, or large errors if the algorithm is overly optimistic about the success of initialization. This is as expected on "rotation" which lacks sufficient parallax. But it is also evident on "line" with sufficient parallax (even worse than "rotation" by VINS-Fusion and ORB-SLAM3), indicating that the low-quality IMU is the primary source of the robustness issue. On the other hand, the low-quality IMU causes serious error accumulation, and loops fail to be closed to correct the error. This is as expected on "line", in which there is a lack of shared views between the two trajectories forth and back along the line. But it is also evident on "rotation" in which there are sufficient shared views while rotating in place. It is possibly due to the error accumulation being too severe for most of the frames with shared views to be successfully matched. Consequently, after being aligned to the ground-truth trajectory, only a portion of the translations in the trajectory can be successfully aligned, and most of the rotations are not aligned, resulting in large APEs and unreasonable AREs. In such cases, the proposed metric of AR deviation proves to be a more effective indicator of performance. For the fourth dataset "general", there are also substantial variations in error distributions among different sequences, indicating the robustness issues related to large-scale scenes and general motions.

Comparing the overall performance among the four systems, VINS-Fusion performs the best, followed by DM-VIO. It is noteworthy that the direct method DM-VIO is expected to be sensitive to rolling shutter cameras, but the result is close to the best indirect VINS-Fusion, and bettern than the indirect ORB-SLAM3. This suggests that the low-quality IMU is the primary factor affecting robustness when applied to a large dataset of diverse mobile phones. The robustness of OpenVINS



(a) AR deviation (b) APE by SE(3) (c) APE by Sim(3) (d) RPE

(e) Failure rate (f) Scale error (g) ARE (h) RRE

Fig. 10: Boxplot of the eight metrics (a)∼(h). The performance of four systems on four sub-datasets are compared using these metrics. For each metric, we calculate the metric value for each sequence, and plot the median, the quartiles and the outliers over the 100 sequences for the first three sub-datasets, and over the 50 sequences for the fourth sub-dataset. (b)(c)(d)(f)(g) use logarithmic scale for better visibility of their wide range of values.

is the worst, with a large number of failures and large errors in the successful frames. Overall, all of the four systems face serious robustness issues on the proposed dataset. These issues are not revealed on the existing datasets such as EuRoC [1], where these systems achieve consistently cm-level accuracy and nearly 100% success rates. This also underscores the value of the proposed dataset.

6 Conclusion

In this work, we propose 100-Phones, a large VI-SLAM dataset consisting of 350 sequences collected by 100 different models of phones. The sensor characteristics among these 100 phone models vary greatly, posing unprecedented challenges for VI-SLAM methods. We design four sub-datasets with distinct motion patterns to reflect different aspects of VI-SLAM methods. We conduct evaluations on the collected dataset using four representative VI-SLAM systems, revealing that the current VI-SLAM methods still have serious robustness issues when it comes to mass deployment on mobile phones.

In the experiments, we discover that dealing with different camera-IMU time offsets and IMU noises among different phone models effectively improves the robustness of VI-SLAM. However, we only implements simple adjustments based on the calibration results. More sophisticated strategies that are tailored to specific algorithm characteristics deserve to be further explored. Moreover, there are more factors that merit further analysis, such as the rolling shutter effect of phone cameras, and the degenerated cases due to insufficient parallax or motion excitation, among others. We believe that the proposed dataset will facilitate future analyses aimed at improving the robustness towards mass deployment of VI-SLAM on mobile phones, ultimately promoting the widespread adoption of AR.

Acknowledgments

This work was partially supported by NSF of China (No. 61932003). We would also like to express our gratitude to Ruohao Zhan and Xinyu Chen for conducting the evaluations of DM-VIO and OpenVINS.

References

[1] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart. The EuRoC micro aerial vehicle datasets. *Int. J. Robotics Res.*, 35(10):1157–1163, 2016. 2, 5, 7, 9

[2] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós. ORB-SLAM3: an accurate open-source library for visual, visual-inertial, and multimap SLAM. *IEEE Trans. Robotics*, 37(6):1874–1890, 2021. 1, 2, 6, 7

[3] D. Chen, N. Wang, R. Xu, W. Xie, H. Bao, and G. Zhang. RNIN-VIO: robust neural inertial navigation aided visual-inertial odometry in challenging scenes. In *IEEE International Symposium on Mixed and Augmented Reality, Bari, Italy*, pp. 275–283. IEEE, 2021. 2

[4] J. A. Delmerico and D. Scaramuzza. A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots. In *IEEE International Conference on Robotics and Automation, Brisbane, Australia*, pp. 2502–2509. IEEE, 2018. 2, 6

[5] 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, Salt Lake City, UT, USA*, pp. 224–236. Computer Vision Foundation / IEEE Computer Society, 2018. 6

[6] J. Engel, V. Koltun, and D. Cremers. Direct sparse odometry. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(3):611–625, 2018. 2

[7] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981. 4

[8] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza. IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation. In L. E. Kavraki, D. Hsu, and J. Buchli, eds., *Robotics: Science and Systems XI, Sapienza University of Rome, Rome, Italy*, 2015. 6

[9] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza. SVO: semidirect visual odometry for monocular and multicamera systems. *IEEE Trans. Robotics*, 33(2):249–265, 2017. 2

[10] P. T. 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, Tokyo, Japan*, pp. 1280–1286. IEEE, 2013. 3, 4

[11] 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, Providence, RI, USA*, pp. 3354–3361. IEEE Computer Society, 2012. 2

[12] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang. OpenVINS: A research platform for visual-inertial estimation. In *IEEE International Conference on Robotics and Automation, Paris, France*, pp. 4666–4672. IEEE, 2020. 2, 7

[13] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*.

Cambridge University Press, 2004. 2

[14] Y. He, B. Xu, Z. Ouyang, and H. Li. A rotation-translation-decoupled solution for robust and efficient visual-inertial initialization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada*, pp. 739–748. IEEE, 2023. 2

[15] V. Holzwarth, J. Gisler, C. Hirt, and A. Kunz. Comparing the accuracy and precision of SteamVR tracking 2.0 and Oculus Quest 2 in a room scale setup. In *International Conference on Virtual and Augmented Reality Simulations*, pp. 42–46, 2021. 5

[16] R. Ichikari, T. Kurata, K. Makita, T. Taketomi, H. Uchiyama, T. Kondo, S. Mori, and F. Shibata. Reference Framework on vSRT-method Benchmarking for MAR. In R. W. Lindeman, G. Bruder, and D. Iwai, eds., *International Conference on Artificial Reality and Telexistence and Eurographics Symposium on Virtual Environments, Adelaide, Australia*, pp. 229–236. Eurographics Association, 2017. 6

[17] H. Jégou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(1):117–128, 2011. 6

[18] G. Klein and D. W. Murray. Parallel tracking and mapping for small AR workspaces. In *IEEE/ACM International Symposium on Mixed and Augmented Reality, Nara, Japan*, pp. 225–234. IEEE Computer Society, 2007. 2

[19] D. Lee, S. Ryu, S. Yeon, Y. Lee, D. Kim, C. Han, Y. Cabon, P. Weinzaepfel, N. Guérin, G. Csurka, and M. Humenberger. Large-scale localization datasets in crowded indoor spaces. In *IEEE Conference on Computer Vision and Pattern Recognition, virtual*, pp. 3227–3236. Computer Vision Foundation / IEEE, 2021. 2

[20] S. Leutenegger, P. T. Furgale, V. Rabaud, M. Chli, K. Konolige, and R. Siegwart. Keyframe-based visual-inertial SLAM using nonlinear optimization. In P. Newman, D. Fox, and D. Hsu, eds., *Robotics: Science and Systems IX, Technische Universität Berlin, Berlin, Germany*, 2013. 1, 2

[21] J. Li, B. Yang, D. Chen, N. Wang, G. Zhang, and H. Bao. Survey and evaluation of monocular visual-inertial SLAM algorithms for augmented reality. *Virtual Real. Intell. Hardw.*, 1(4):386–410, 2019. 2, 5, 6

[22] P. Li, T. Qin, B. Hu, F. Zhu, and S. Shen. Monocular Visual-Inertial State Estimation for Mobile Augmented Reality. In W. Broll, H. Regenbrecht, and J. E. S. II, eds., *IEEE International Symposium on Mixed and Augmented Reality, Nantes, France*, pp. 11–21. IEEE Computer Society, 2017. 3

[23] 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, Salt Lake City, UT, USA*, pp. 1974–1982. Computer Vision Foundation / IEEE Computer Society, 2018. 6

[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, Recife, Brazil*, pp. 219–224. IEEE, 2020. 2

[25] H. Liu, G. Zhang, and H. Bao. Robust keyframe-based monocular SLAM for augmented reality. In W. Broll, H. Saito, and J. E. S. II, eds., *IEEE International Symposium on Mixed and Augmented Reality, Merida, Yucatan, Mexico*, pp. 1–10. IEEE Computer Society, 2016. 2, 3

[26] H. Liu, L. Zhao, Z. Peng, W. Xie, M. Jiang, H. Zha, H. Bao, and G. Zhang. A low-cost and scalable framework to build large-scale localization benchmark for augmented reality. *IEEE Transactions on Circuits and Systems for Video Technology*, 2023. doi: 10.1109/TCSVT.2023.3306160 2, 6

[27] W. Liu, D. Caruso, E. Ilg, J. Dong, A. I. Mourikis, K. Daniilidis, V. Kumar, and J. Engel. TLIO: tight learned inertial odometry. *IEEE Robotics Autom. Lett.*, 5(4):5653–5660, 2020. 2

[28] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.*, 60(2):91–110, 2004. 6

[29] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2022. 2

[30] 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, Roma, Italy*, pp. 3565–3572. IEEE, 2007. 1, 2

[31] T. Müller, A. Evans, C. Schied, and A. Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, 2022. 2

[32] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Trans. Robotics*, 31(5):1147–1163, 2015. 2

[33] R. Mur-Artal and J. D. Tardós. ORB-SLAM2: an open-source SLAM system for monocular, stereo, and RGB-D cameras. *IEEE Trans. Robotics*, 33(5):1255–1262, 2017. 2

[34] R. Mur-Artal and J. D. Tardós. Visual-inertial monocular SLAM with map reuse. *IEEE Robotics Autom. Lett.*, 2(2):796–803, 2017. 2

[35] E. Olson. Apriltag: A robust and flexible visual fiducial system. In *IEEE International Conference on Robotics and Automation, Shanghai, China*, pp. 3400–3407. IEEE, 2011. 3, 5

[36] L. Oth, P. T. Furgale, L. Kneip, and R. Siegwart. Rolling shutter camera calibration. In *IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA*, pp. 1360–1367. IEEE Computer Society, 2013. 3

[37] B. Pfrommer, N. Sanket, K. Daniilidis, and J. Cleveland. PennCOSYVIO: A challenging visual inertial odometry benchmark. In *IEEE International Conference on Robotics and Automation, Singapore, Singapore*, pp. 3847–3854. IEEE, 2017. 2

[38] T. Qin, P. Li, and S. Shen. VINS-Mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Trans. Robotics*, 34(4):1004–1020, 2018. 1, 2, 7

[39] T. Qin and S. Shen. Online temporal calibration for monocular visual-inertial systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, Madrid, Spain*, pp. 3662–3669. IEEE, 2018. 2, 7

[40] A. Rosinol, J. J. Leonard, and L. Carlone. NeRF-SLAM: Real-time dense monocular SLAM with neural radiance fields. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3437–3444, 2023. 2

[41] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski. ORB: an efficient alternative to SIFT or SURF. In D. N. Metaxas, L. Quan, A. Sanfeliu, and L. V. Gool, eds., *IEEE International Conference on Computer Vision, Barcelona, Spain*, pp. 2564–2571. IEEE Computer Society, 2011. 2

[42] P. Sarlin, M. Dusmanu, J. L. Schönberger, P. Speciale, L. Gruber, V. Larsson, O. Miksik, and M. Pollefeys. LaMAR: Benchmarking Localization and Mapping for Augmented Reality. In S. Avidan, G. J. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, eds., *European Conference on Computer Vision*, vol. 13667 of *Lecture Notes in Computer Science*, pp. 686–704. Springer, 2022. 3

[43] J. L. Schönberger and J. Frahm. Structure-from-motion revisited. In *IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA*, pp. 4104–4113. IEEE Computer Society, 2016. 2, 6

[44] J. L. Schönberger, T. Price, T. Sattler, J. Frahm, and M. Pollefeys. A vote-and-verify strategy for fast spatial verification in image retrieval. In S. Lai, V. Lepetit, K. Nishino, and Y. Sato, eds., *Asian Conference on Computer Vision, Taipei, Taiwan*, vol. 10111 of *Lecture Notes in Computer Science*, pp. 321–337. Springer, 2016. 6

[45] D. Schubert, T. Goll, N. Demmel, V. Usenko, J. Stückler, and D. Cremers. The TUM VI benchmark for evaluating visual-inertial odometry. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, Madrid, Spain*, pp. 1680–1687. IEEE, 2018. 2, 7

[46] J. Shi and C. Tomasi. Good features to track. In *Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA*, pp. 593–600. IEEE, 1994. 6

[47] 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 Trans. Circuits Syst. Video Technol.*, 31(4):1253–1267, 2021. 2

[48] 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, Vilamoura, Algarve, Portugal*, pp. 573–580. IEEE, 2012. 6

[49] 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, Adelaide, Australia*, pp. 209–218. IEEE Computer Society, 2013. 2

[50] K. Tateno, F. Tombari, I. Laina, and N. Navab. CNN-SLAM: real-time dense monocular SLAM with learned depth prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6243–6252, 2017. 2

[51] Z. Teed and J. Deng. DROID-SLAM: deep visual SLAM for monocular, stereo, and RGB-D cameras. In M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, eds., *Advances in Neural Information Processing Systems*, pp. 16558–16569, 2021. 2

[52] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment - A modern synthesis. In B. Triggs, A. Zisserman, and R. Szeliski, eds., *International Workshop on Vision Algorithms, held dur-*

ing ICCV '99, Corfu, Greece, Proceedings, vol. 1883 of *Lecture Notes in Computer Science*, pp. 298–372. Springer, 1999. 2

[53] L. von Stumberg and D. Cremers. DM-VIO: delayed marginalization visual-inertial odometry. *IEEE Robotics Autom. Lett.*, 7(2):1408–1415, 2022. 2, 7

[54] C. Wang, Y. Zhao, J. Guo, L. Pei, Y. Wang, and H. Liu. NEAR: the netease AR oriented visual inertial dataset. In *IEEE International Symposium on Mixed and Augmented Reality Adjunct, Beijing, China*, pp. 366–371. IEEE, 2019. 2

[55] K. Wu, A. M. Ahmed, G. A. Georgiou, and S. I. Roumeliotis. A square root inverse filter for efficient vision-aided inertial navigation on mobile devices. In L. E. Kavraki, D. Hsu, and J. Buchli, eds., *Robotics: Science and Systems XI, Sapienza University of Rome, Rome, Italy*, 2015. 1

[56] K. Wu, C. X. Guo, G. A. Georgiou, and S. I. Roumeliotis. VINS on wheels. In *IEEE International Conference on Robotics and Automation, Singapore, Singapore*, pp. 5155–5162. IEEE, 2017. 2

[57] F. Xue, X. Wang, S. Li, Q. Wang, J. Wang, and H. Zha. Beyond tracking: Selecting memory and refining poses for deep visual odometry. In *IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA*, pp. 8575–8583. Computer Vision Foundation / IEEE, 2019. 2

[58] Z. Zhu, S. Peng, V. Larsson, Z. Cui, M. R. Oswald, A. Geiger, and M. Pollefeys. NICER-SLAM: neural implicit scene encoding for RGB SLAM. *arxiv preprint arXiv:2302.03594*, 2023. 2