

MobileARLoc: On-device Robust Absolute Localisation for Pervasive Markerless Mobile AR

Changkun Liu^{1*}, Yukun Zhao^{1*}, Tristan Braud^{1,2}

¹Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong

²Division of Integrative Systems and Design, The Hong Kong University of Science and Technology, Hong Kong
{cliudg,yzhaoeg}@connect.ust.hk, braudt@ust.hk

Abstract—Recent years have seen significant improvement in absolute camera pose estimation, paving the way for pervasive markerless Augmented Reality (AR). However, accurate absolute pose estimation techniques are computation- and storage-heavy, requiring computation offloading. As such, AR systems rely on visual-inertial odometry (VIO) to track the device’s relative pose between requests to the server. However, VIO suffers from drift, requiring frequent absolute repositioning. This paper introduces MobileARLoc, a new framework for on-device large-scale markerless mobile AR that combines an absolute pose regressor (APR) with a local VIO tracking system. Absolute pose regressors (APRs) provide fast on-device pose estimation at the cost of reduced accuracy. To address APR accuracy and reduce VIO drift, MobileARLoc creates a feedback loop where VIO pose estimations refine the APR predictions. The VIO system identifies reliable predictions of APR, which are then used to compensate for the VIO drift. We comprehensively evaluate MobileARLoc through dataset simulations. MobileARLoc halves the error compared to the underlying APR and achieves fast (80 ms) on-device inference speed.

I. INTRODUCTION

Visual localization systems utilize the visual data captured by a device’s camera to determine its 6 degrees of freedom (6DOF) absolute pose (translation and rotation) within established world coordinates for a known scene. Accurate visual positioning is essential for augmented reality (AR) applications where content is anchored into the physical world without markers, paving the way to pervasive AR.

Highly accurate localisation systems typically rely on 3D structure-based methods [1]–[4] that detect and match visual features in images against a 3D model of the environment. However, these methods are often demanding in terms of computation and storage [5]. Scaling up such methods to larger environments thus requires offloading computations to distant servers, adding latency and jitter [6], and raising significant privacy concerns [7]. These constraints have led the industry to adopt on-device localisation with strict access control on the camera frames in home environments¹. However, the computational cost of structure-based methods prevents their application on-device at a larger scale. Absolute pose regressors (APRs) are end-to-end machine learning models that estimate the device pose using a single monocular image. They provide fast on-device inference with minimal storage, even in large

environments and over multiple scenes [8]. However, their low accuracy and robustness have prevented their application to large-scale mobile AR [9]. Although absolute localisation is a significant challenge, most markerless AR applications track the relative pose through visual-inertial odometry (VIO). VIO systems calculate the displacement between camera frames using visual and inertial data from sensors. As such, they tend to display high accuracy in the short term, but they drift over time [10]. APR and VIO present antagonistic features: APR poses are noisy yet drift-free, while VIO is very accurate with errors building up over time [11]. We believe that VIO’s high accuracy could thus improve APR’s imprecision, while the APR’s predictions could address VIO drift.

This paper introduces MobileARLoc, an on-device visual localisation framework for large-scale mobile AR that combines the complementary properties of VIO and APR. Accurate APR predictions should be consistent with the relative odometry estimates obtained from VIO. Otherwise, the APR prediction should be considered unreliable. MobileARLoc aligns the APR and VIO coordinate systems to identify reliable APR poses and refine unreliable predictions. When several consecutive APR poses are consistent with the VIO relative output, they are considered accurate. MobileARLoc calculates the average of reliable absolute predictions as the *reference pose* and the rigid transformation between this reference pose and the corresponding VIO poses to align the coordinate systems. Following the alignment stage, MobileARLoc enters the *pose optimization stage*. Each APR pose is compared to the corresponding VIO pose. The APR prediction is output directly if reliable. Otherwise, MobileARLoc outputs the VIO pose converted into absolute world coordinates using rigid transformation. We introduce a new similarity metric to detect VIO drift. When drift is detected, MobileARLoc reenters the alignment stage to select new reliable poses and calculate rigid transformation.

We summarize our main contributions as follows:

- 1) We **design an APR-agnostic framework**, MobileARLoc, for real-time on-device pose estimation. MobileARLoc leverages VIO data to select reliable APR predictions, refine unreliable predictions and compensate drift.
- 2) We **implement and evaluate MobileARLoc** over two popular APR models, PoseNet (PN) [12] and MS-Transformer (MS-T) [8]. MobileARLoc improves MS-

*Both authors contributed equally to this research.

¹<https://developer.oculus.com/blog/mixed-reality-with-passthrough/>

T's accuracy by up to 47% in translation and 66% in rotation over the average of the three outdoor scenes.

- 3) We integrate MobileARLoc into a **real-life mobile AR application** and evaluate its performance.

II. RELATED WORK

A. Absolute Pose Regression

Absolute Pose Regressors train deep neural networks to regress the 6-DOF camera pose of a query image. The first APR is PoseNet (PN) [12]. Since then, there have been several improvements to APR, mainly related to the backbone architecture and loss functions [11], [13], [14]. MS-Transformer (MS-T) [8] extends the single-scene paradigm of APR for learning multiple scenes. MapNet+ [11] and DFNet_{dm} [14] finetune pre-trained network on unlabeled test data to improve the accuracy. However, in-test-time finetuning neural networks is time-consuming and unlabeled test-set data is difficult to obtain in advance in real applications. Among these works, MapNet [11] aims to minimize the loss of the per-image absolute pose and the loss of the relative pose between image pairs. However, formulating relative pose constraints as loss terms during training shows limited accuracy improvement and has been surpassed by state-of-the-art (SOTA) APRs like MS-T [8]. Our framework improves accuracy at test time by using the relative pose independently from the training strategy. Compared to most modern approaches, MobileARLoc does not rely on additional unlabeled test data.

B. Uncertainty estimation and Pose Optimization

APRs suffer from limited generalizability [9]. Uncertainty-aware APRs aim to infer which images will likely result in accurate pose estimation and identify the outliers. Several prior works have explored uncertainty estimation during APR training. Bayesian PoseNet [15] and AD-PoseNet [16] model the uncertainty by measuring the variance of several inferences of the same input data. CoordiNet [17] models heteroscedastic uncertainty during training. Deng *et al.* [18] represent uncertainty by predicting a mixture of multiple unimodal distributions. Although these uncertainty-aware APRs provide both pose predictions and uncertainty estimates, the accuracy of predictions is much lower than other APRs [8], [13], [14]. Moreover, existing uncertainty estimation methods can be time-consuming [15] and lack extensibility due to the need for specific loss functions and training schemes [16]–[19].

Our framework enables greater flexibility compared to existing uncertainty-aware methods by being APR-agnostic, enabling the integration of most mainstream APRs. Our method performs a rigid transformation of the VIO's pose to optimize unreliable APR poses directly, without iterative optimization. It improves the accuracy of APRs with minimal overhead, enabling reliable APR usage in mobile AR applications.

III. PROPOSED APPROACH

A. Definition

Given a query image I_i in a known scene, APR \mathcal{R} outputs global translation $\hat{\mathbf{x}}_i$ and rotation $\hat{\mathbf{q}}_i$ in an established world

coordinate system for the scene, so that $\mathcal{R}(I_i) = \hat{p}_i = \langle \hat{\mathbf{x}}_i, \hat{\mathbf{q}}_i \rangle$ is the estimated camera pose for I_i . The Ground Truth (GT) of I_i in the world coordinate system is $p_i = \langle \mathbf{x}_i, \mathbf{q}_i \rangle$. The camera pose of I_i in VIO coordinate system is noted $p_i^{vio} = \langle \mathbf{x}_i^{vio}, \mathbf{q}_i^{vio} \rangle$. The relative translation between two consecutive images I_i and I_{i+1} is characterized by

$$\hat{\Delta}_{trans}(i+1, i) = \|\hat{\mathbf{x}}_{i+1} - \hat{\mathbf{x}}_i\|_2, \quad (1)$$

$$\Delta_{trans}^{vio}(i+1, i) = \|\mathbf{x}_{i+1}^{vio} - \mathbf{x}_i^{vio}\|_2, \quad (2)$$

$$\Delta_{trans}(i+1, i) = \|\mathbf{x}_{i+1} - \mathbf{x}_i\|_2. \quad (3)$$

Similarly, we get relative rotation between I_i and I_{i+1} in degree, \mathbf{q}^{-1} denotes the conjugate of \mathbf{q} , and we assume all quaternions are normalized: $\hat{\Delta}_{rot}(i+1, i) = 2 \arccos |\hat{\mathbf{q}}_{i+1}^{-1} \hat{\mathbf{q}}_i| \frac{180}{\pi}$, $\Delta_{rot}^{vio}(i+1, i) = 2 \arccos |\mathbf{q}_{i+1}^{vio-1} \mathbf{q}_i^{vio}| \frac{180}{\pi}$ and $\Delta_{rot}(i+1, i) = 2 \arccos |\mathbf{q}_{i+1}^{-1} \mathbf{q}_i| \frac{180}{\pi}$. $\hat{u}_{i,i+1} = \langle \hat{\Delta}_{trans}(i, i+1), \hat{\Delta}_{rot}(i, i+1) \rangle$ is the odometry of I_i and I_{i+1} from predicted poses of APR. $u_{i,i+1}^{vio} = \langle \Delta_{trans}^{vio}(i, i+1), \Delta_{rot}^{vio}(i, i+1) \rangle$ is the odometry of I_i and I_{i+1} from the VIO system. Similarly, $u_{i,i+1} = \langle \Delta_{trans}(i, i+1), \Delta_{rot}(i, i+1) \rangle$ is the GT odometry of I_i and I_{i+1} . We then define the Relative Position Error (RPE) and the Relative Orientation Error (ROE) for the VIO and the APR as follows:

$$\text{RPE}_{i,i+1}^{\langle vio, GT \rangle} = |\Delta_{trans}(i+1, i) - \Delta_{trans}^{vio}(i+1, i)| \quad (4)$$

$$\text{ROE}_{i,i+1}^{\langle vio, GT \rangle} = |\Delta_{rot}(i+1, i) - \Delta_{rot}^{vio}(i+1, i)| \quad (5)$$

$$\text{RPE}_{i,i+1}^{\langle apr, vio \rangle} = |\hat{\Delta}_{trans}(i+1, i) - \Delta_{trans}^{vio}(i+1, i)| \quad (6)$$

$$\text{ROE}_{i,i+1}^{\langle apr, vio \rangle} = |\hat{\Delta}_{rot}(i+1, i) - \Delta_{rot}^{vio}(i+1, i)| \quad (7)$$

B. Detecting reliable pose estimations using VIO

Modern VIO systems have low drift at a small temporal scale, and $u_{i,i+1}^{vio}$ tends to be very close to the GT odometry, $u_{i,i+1}$. We can assume $\text{RPE}_{i,i+1}^{\langle vio, GT \rangle}$ and $\text{ROE}_{i,i+1}^{\langle vio, GT \rangle}$ are almost 0. We model the uncertainty of the APR output, \hat{p}_{i+1} , with $u_{i,i+1}^{vio}$, taking advantage of this property. If the $\text{RPE}_{i,i+1}^{\langle apr, vio \rangle}$ and $\text{ROE}_{i,i+1}^{\langle apr, vio \rangle}$ of multiple consecutive images are very small, we consider these predictions are accurate.

We define a distance threshold d_{th} for $\text{RPE}^{\langle apr, vio \rangle}$ and an orientation threshold o_{th} for $\text{ROE}^{\langle apr, vio \rangle}$. An estimated APR pose is considered accurate if the error close to GT within $\frac{d_{th}}{2}$ and $\frac{o_{th}}{2}$. Given two consecutive query images I_i, I_{i+1} ,

- 1) Estimated poses of I_i and I_{i+1} are accurate, then $\text{RPE}_{i,i+1}^{\langle apr, vio \rangle}$ and $\text{ROE}_{i,i+1}^{\langle apr, vio \rangle}$ are lower than d_{th} and o_{th} , respectively.
- 2) One of the estimated pose for I_i and I_{i+1} is not accurate, then either $\text{RPE}_{i,i+1}^{\langle apr, vio \rangle}$ should be larger than d_{th} or $\text{ROE}_{i,i+1}^{\langle apr, vio \rangle}$ should be larger than o_{th} .

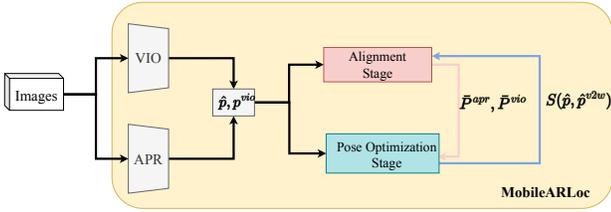


Fig. 1. MobileARLoc framework in a mobile AR system.

- 3) Both estimated poses of I_i and I_{i+1} are inaccurate. However, $\text{RPE}_{i,i+1}^{<apr,vio>}$ remains lower than d_{th} and $\text{ROE}_{i,i+1}^{<apr,vio>}$ lower than o_{th} .
- 4) Both estimated poses of I_i and I_{i+1} are inaccurate, and either $\text{RPE}_{i,i+1}^{<apr,vio>}$ is larger than d_{th} or $\text{ROE}_{i,i+1}^{<apr,vio>}$ is larger than o_{th} .

When either $\text{RPE}_{i,i+1}^{<apr,vio>}$ or $\text{ROE}_{i,i+1}^{<apr,vio>}$ is larger than its respective threshold (case (2) and (4)), the pose is flagged as inaccurate and can thus be filtered out. Similarly, in case (1), the two poses are identified as accurate. In case (3), two inaccurate poses are identified as accurate. Our method uses a probabilistic approach to reducing such false positives. APR error tends to be random with a large variance. As such, two consecutive images presenting a large APR error while being close to each other in the same direction as the VIO is a rare occurrence. By comparing more pairs of images, we further reduce the probability of false positive, filtering out the most unreliable predictions. We then obtain the rigid transformation between the VIO coordinate system and the world coordinate system by using the reliable predicted poses and VIO poses. To ensure the rigid transform relationship's reliability, we calculate the average pose of selected predicted poses as *reference pose*. The rotation and translation of the coordinate system of VIO and the world coordinate system change over time due to the VIO drift. Therefore, we only need to update the reliable poses occasionally and optimize the predicted pose by calculating the new rotation and translation.

C. MobileARLoc framework

Based on the above subsections, we present MobileARLoc, a new APR-agnostic framework that combines the outputs of APRs and information from smartphones' VIO systems to improve pose prediction accuracy. The framework keeps the most reliable prediction poses with the help of VIO, identifies unreliable poses, and optimizes them based on reliable poses. MobileARLoc combines the distinctive features of APR predictions, which are locally noisy but drift-free, with mobile VIO systems, which are locally smooth but tend to drift, as shown by [11]. This framework consists of two alternating looping stages: *Alignment* and *Pose optimization*, as shown in Figure 1. The *Alignment* stage identifies multiple reliable poses to calculate the *reference pose*. The *Pose Optimization* stage optimizes unreliable poses based on this *reference pose* and the VIO poses. MobileARLoc adaptively goes back to the *Alignment* phase to recalculate the *reference pose* and thus negate the effect of VIO drift.

Alignment stage checks the odometry of consecutive $N+1$ images. We consider I_j to be the first image to enter the alignment stage. When all N consecutive pairs of images satisfy the requirement that $\text{RPE}^{<apr,vio>} \leq d_{th}$ and $\text{ROE}^{<apr,vio>} \leq o_{th}$, predictions $\{\hat{p}_i\}_{i=j}^{j+N+1}$ of these $N+1$ consecutive images from I_j to I_{j+N+1} can be considered as accurate predictions. d_{th} and o_{th} are the relative pose checker's distance and orientation threshold to filter the inaccurate estimated poses. If the difference between $\hat{u}_{i,i+1}$ and $u_{i,i+1}^{vio}$ is less than d_{th} and o_{th} simultaneously, poses \hat{p}_i and \hat{p}_{i+1} become candidate reliable predictions (RPs). If the difference between $\hat{u}_{i,i+1}$ and $u_{i,i+1}^{vio}$ is larger than d_{th} or o_{th} , we assume \hat{p}_i is inaccurate and discard all previous candidate RPs. Upon getting $N+1$ candidate RPs, We perform geometric averaging from these RPs using Weiszfeld's algorithm. and [20] to obtain a *reference pose* \bar{P}^{apr} since we assume the pose error of APR is normally distributed in space. We perform the same geometric averaging from corresponding VIO poses to get the *reference pose* \bar{P}^{vio} in the VIO system. Once \bar{P}^{apr} and \bar{P}^{vio} are obtained, MobileARLoc moves to the *Pose Optimization* stage.

In **Pose Optimization** stage, the framework checks the current predicted pose against the previous pose as follows. For the subsequent predicted poses $\{\hat{p}_i\}_{i=j+N+2}$ and corresponding VIO poses $\{p_i^{vio}\}_{i=j+N+2}$, we get the odometry $\hat{u}_{i-1,i}$ of \hat{p}_{i-1} and \hat{p}_i in APR coordinates and odometry $\hat{u}_{i-1,i}^{vio}$ of p_{i-1}^{vio} and p_i^{vio} in VIO coordinates respectively. If $\text{RPE}_{i-1,i}^{<apr,vio>} \leq d_{th}$ and $\text{ROE}_{i-1,i}^{<apr,vio>} \leq o_{th}$ is satisfied, the APR output for this image is considered reliable as the difference of odometry between APR and VIO is small. The pose is considered reliable and can be output directly. Otherwise, the pose is optimized by $\text{OptimizePose}(p^{vio}, \bar{P}^{apr}, \bar{P}^{vio})$. We calculate the rigid transformation between VIO coordinates and world coordinates using \bar{P}^{apr} and \bar{P}^{vio} . Then we transform p^{vio} to world coordinates as p^{v2w} and replace the unreliable pose.

We call the $N+1$ consecutive predictions in Alignment stage for calculating the average reference pose and predictions pass the relative pose threshold in Pose Optimization stage as reliable predictions (RPs). To compensate for VIO drift, we detect the drift and make the system loop back to the *Alignment* stage adaptively: for each RP in *Pose Optimization* stage, we calculate the similarity between the $\hat{p} = \langle \hat{\mathbf{x}}, \hat{\mathbf{q}} \rangle$ and $p^{v2w} = \langle \mathbf{x}^{v2w}, \mathbf{q}^{v2w} \rangle$ from $\text{OptimizePose}(p^{vio}, \bar{P}^{apr}, \bar{P}^{vio})$:

$$S(\hat{p}, p^{v2w}) = \frac{(\frac{\hat{\mathbf{x}} \cdot \mathbf{x}^{v2w}}{\|\hat{\mathbf{x}}\|_2 \cdot \|\mathbf{x}^{v2w}\|_2} + \frac{\hat{\mathbf{q}} \cdot \mathbf{q}^{v2w}}{\|\hat{\mathbf{q}}\|_2 \cdot \|\mathbf{q}^{v2w}\|_2})}{2} \quad (8)$$

where $-0.5 < S(\hat{p}, p^{v2w}) < 1$. We use similarity here to balance the difference of translation and rotation because they have different units. If consecutive N reliable predictions have $S(\hat{p}, p^{v2w}) \leq \gamma$, it indicates drift happens and the system loops back to the *Alignment* since p^{v2w} should be very close to RPs. The local tracking of VIO system is used between the last pose in *Pose Optimization* stage until new RPs are found in the *Alignment* stage and new reference poses are updated.

TABLE I
DATASET DETAILS AND STATISTICS FOR HLoc (IMAGE RETRIEVAL WITH TOP-20 RECALL), PN, AND MS-T.

	Scenes	Dataset quantity		Spatial Extent (m)	Storage (MB)			Runtime (ms)		
		Train	Test		HLoc	PN	MS-T	HLoc	PN	MS-T
Outdoor	Square	2058	1023	40×25	6500	85	71	4965	4	15
	Church	1643	853	50×40	6600	85	71	6659	4	15
	Bar	1834	838	55×35	6600	85	71	6230	4	15
Indoor	Stairs	873	222	5.5 × 4.5 × 6	1500	85	71	2263	4	15
	Office	1479	635	7.5 × 4	2300	85	71	5722	4	15
	Atrium	1694	441	30 × 50	5100	85	71	4500	4	15

IV. DATASET

We collect a dataset of image and VIO data using iPhone 14 Pro Max, the flagship ARKit 6 phone at the time. The resolution of all images is 1920×1440 . All images are fed into an SfM framework using COLMAP [21] to get the GT. We compare the GT with the pose labels of VIO. Our dataset highlights the low drift of current mobile VIO solutions, such as ARKit, supporting our assumption that VIO system can reinforce absolute pose estimation. Table I provides summary statistics, as well as the storage requirements and runtime per request for HLoc, PN, and MS-T. (see Section VI-A).

V. IMPLEMENTATION

A. Desktop Implementation

There are four hyperparameters. d_{th} and o_{th} are the relative pose checker’s distance and orientation thresholds to filter inaccurate pose estimations. If among $N+1$ consecutive images, N image pairs pass the relative pose checker, we consider these pose estimations to be accurate. We set $\gamma = 0.99$, $d_{th} = 0.4m$ and $o_{th} = 4^\circ$ over all datasets. We set $N = 2$, with one frame processed per second for the AR application and all the experiments in this paper. We implement our framework over two APR models:

PN. PoseNet (PN) is the baseline method. Since there is no open source code for PoseNet [12], we follow [11], [13] and use ResNet34 [22] as the backbone network.

MS-T. MS-Transformer [8] (MS-T) extends the single-scene paradigm of APR to learning multiple scenes in parallel and is one of the most recent APRs with official opensource code. Therefore, we trained one MS-T for all outdoor scenes and one MS-T for all indoor scenes using the official code².

We note APR methods integrated into our MobileARLoc framework as APR^{vio}. During training, all input images are resized to 256×256 and then randomly cropped to 224×224 . For both PN and MS-T, we set an initial learning rate of $\lambda = 10^{-4}$. All experiments for evaluation in Section VII are performed on an NVIDIA GeForce GTX 3090 GPU.

B. Application Implementation

We implement MobileARLoc as a mobile AR app using Unity and ARKit to run on an iPhone 14 Pro Max. We convert the pre-trained PN to ONNX format and incorporated it into

²<https://github.com/yolish/multi-scene-pose-transformer>

a Unity application. We use OpenCVforUnity³ for processing query images and use Barracuda transferring resized images to tensor as the input of the network.

VI. SYSTEM PERFORMANCE

A. Desktop Implementation

As mentioned in Section I, structure-based methods tend to require significant resources that are not available on mobile devices. Table I shows the performance of HLoc [2] pipelines⁴ prevents one-device camera relocalisation. In MobileARLoc, the APR only requires storing neural network weights. The memory requirement of the APR thus remains constant, between 71 and 85 MB depending on the backbone model. Meanwhile, HLoc pipelines require 1) a pre-built 3D model; 2) an image database; 3) a local descriptor database; and 4) the models for image retrieval and feature extraction. The memory represents between 1.5 and 6.6 GB per scene. APRs only require a single forward pass for each query image, leading to a runtime between 4 and 15 ms. In contrast, HLoc pipelines take up to 6.7 s on larger scenes.

B. Mobile Implementation

We assess the performance of our framework on an iPhone 14 Pro Max device on the setup described in Section V-B. We measure each parameter over 200 samples. The average processing time per image is 37 ms while the average time for PN to infer an image is 39.5 ms. Under current ARKit’s implementation, VIO runs in a parallel loop every frame. Therefore, our pipeline can perform absolute camera localization on a mobile device in less than 80 ms. The ResNet34-based PN requires only 85 MB for weight storage.

VII. DATASET EVALUATION

We evaluate our framework on the datasets described in Section IV and Section V-A. Due to the disparity in computational scale between our approach and structure-based methods like HLoc (refer to Table I), we do not present results for HLoc.

We evaluate the performance of APR and our framework through two primary metrics. We consider the mean and median APE and AOE in Tables II as shown in Equation (9) and Equation (10) for all test frames. We also evaluate the percentage of test images with pose predicted with high ($0.25m, 2^\circ$), medium ($0.5m, 5^\circ$), and low ($5m, 10^\circ$) accuracy levels proposed by [23] in Table III. The higher the percentage of each accuracy level, the better the performance.

$$\text{APE}^{\langle apr, GT \rangle} = \|\hat{\mathbf{x}}_i - \mathbf{x}_i\|_2 \quad (9)$$

$$\text{AOE}^{\langle apr, GT \rangle} = 2 \arccos |\mathbf{q}_i^{-1} \hat{\mathbf{q}}_i| \frac{180}{\pi} \quad (10)$$

³<https://assetstore.unity.com/packages/tools/integration/opencv-for-unity-21088>

⁴<https://github.com/cvg/Hierarchical-Localization>

TABLE II

MEAN AND MEDIAN ABSOLUTE TRANSLATION/ROTATION ERRORS IN $m/^\circ$. THE RATIO OF RPs AND OPT. POSES ARE PROVIDED IN TABLE III.

Only RPs Mean				Only RPs Median				
Outdoor	PN	PN ^{vio} (ours)	MS-T	MS-T ^{vio} (ours)	PN	PN ^{vio} (ours)	MS-T	MS-T ^{vio} (ours)
Square	2.17/0.07	1.17/3.57	2.50/4.14	1.64/2.46	1.11/3.61	0.76/3.04	1.52/14	0.81/1.78
Church	1.53/8.04	0.82/4.0	1.91/11.9	0.65/3.0	0.73/3.97	0.59/3.0	0.71/3.08	0.49/2.01
Bar	1.44/4.03	0.78/2.89	1.65/2.82	0.86/1.97	0.66/2.82	0.52/2.38	0.69/1.65	0.52/1.47
average	1.69/6.38	0.92/3.49	2.02/6.29	1.05/2.48	0.83/3.47	0.62/2.81	0.97/2.29	0.61/1.75
Only Opt. Mean				Only Opt. Median				
Square	2.8/10.5	1.4/2.14	3.25/5.54	1.27/2.12	1.45/4.38	0.98/2.16	2.22/2.44	1.04/1.89
Church	2.07/11.2	1.08/1.98	2.85/18.6	1.1/1.86	0.93/5.04	0.89/1.87	1.13/4.74	0.93/1.70
Bar	2.18/5.35	1.07/2.12	2.42/3.74	0.96/1.56	0.94/3.55	0.86/2.18	0.99/1.75	0.86/1.53
average	2.35/9.02	1.18/2.08	2.84/9.29	1.1/1.85	1.11/4.32	0.91/2.07	1.45/2.98	0.94/1.65
RPs + Opt. Mean				RPs + Opt. Median				
Square	2.17/0.07	1.3/2.8	2.49/4.14	1.44/2.27	1.11/3.61	0.84/2.36	1.52/14	0.96/1.87
Church	1.53/8.04	0.96/2.88	1.91/11.9	0.92/2.34	0.73/3.97	0.74/2.13	0.71/3.08	0.73/1.79
Bar	1.44/4.03	0.92/2.53	1.65/2.82	0.91/1.78	0.66/2.82	0.69/2.26	0.69/1.65	0.71/1.44
average	1.69/6.38	1.06/2.74	2.02/6.29	1.08/2.13	0.83/3.47	0.76/2.25	0.97/2.29	0.8/1.7
Only RPs Mean				Only RPs Median				
Indoor	PN	PN ^{vio} (ours)	MS-T	MS-T ^{vio} (ours)	PN	PN ^{vio} (ours)	MS-T	MS-T ^{vio} (ours)
Stairs	0.35/7.51	0.27/5.25	0.26/6.8	0.22/4.56	0.27/4.89	0.25/4.88	0.18/4.33	0.16/3.6
Office	0.55/11.7	0.45/7.3	0.48/13.3	0.40/7.36	0.36/6.21	0.32/5.55	0.35/4.66	0.30/4.11
Atrium	2.71/10.2	1.79/6.14	3.66/9.08	1.95/5.03	1.78/5.54	1.2/4.3	2.03/9.8	1.35/3.23
average	1.2/9.8	0.84/6.23	1.47/9.73	0.86/5.65	0.80/5.55	0.59/4.91	0.84/4.32	0.6/3.65
Only Opt. Mean				Only Opt. Median				
Stairs	0.50/11.76	0.21/4.88	0.32/11.9	0.19/3.5	0.40/4.98	0.20/4.96	0.25/7	0.13/3.74
Office	0.9/25.2	0.51/6.49	0.8/31.3	0.33/4.27	0.79/12.2	0.36/6.1	0.6/12.8	0.21/3.34
Atrium	3.13/13.2	2.26/12.9	4.6/13.5	1.2/1.95	2.53/7.3	2.09/4.9	2.6/6	1.16/1.6
average	1.51/16.73	0.99/8.09	1.91/18.9	0.57/3.24	1.24/8.16	0.88/5.32	1.15/8.6	0.5/2.89
RPs + Opt. Mean				RPs + Opt. Median				
Stairs	0.35/7.51	0.27/5.25	0.26/6.8	0.22/4.56	0.27/4.89	0.22/4.90	0.18/4.33	0.15/3.68
Office	0.55/11.7	0.47/7.13	0.48/13.3	0.39/6.89	0.36/6.21	0.34/5.8	0.35/4.66	0.28/4.0
Atrium	2.71/10.2	1.99/9.0	3.66/9.08	1.59/3.52	1.78/5.54	1.57/4.81	2.03/9.8	1.21/2.61
average	1.2/9.8	0.91/7.13	1.47/9.73	0.73/4.99	0.80/5.55	0.71/5.17	0.84/4.32	0.55/3.43

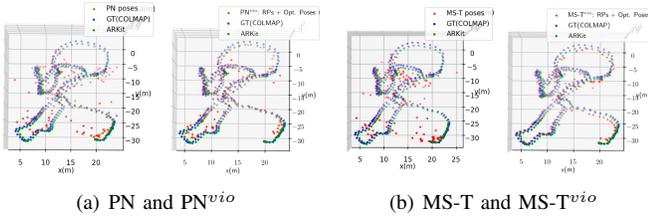


Fig. 2. Pose predictions for APR (left) and APR^{vio} (right) for one test sequence in the Square scene. Using MobileARLoc significantly decreases the number of predictions with large error as well as the noisiness of pose estimation error compared to APR alone.

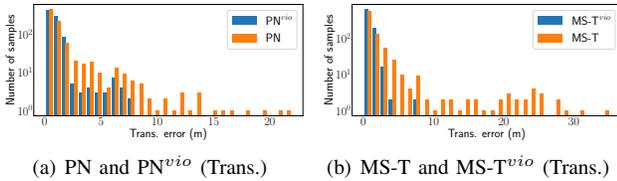


Fig. 3. Pose error distribution for Church scene. Using MobileARLoc significantly decreases the number of predictions with large error.

A. Results

Reliable predictions: For outdoor scenes, Table III demonstrates that selecting RPs comprising 41.8% to 51.3% of total PN predictions and 41.6% to 51% of total MS-T predictions significantly increases the percentage of each accuracy level. Consequently, pose estimates below the low accuracy threshold (5m, 10°) are greatly reduced in all three outdoor scenes. Moreover, RPs selected by PN^{vio} and MS-T^{vio} exhibit higher mean and median accuracy across all predictions compared to PN and MS-T (Table II). The mean accuracy improvement surpasses the median accuracy improvement. In the three

outdoor scenes, PN^{vio}-preserved RPs reduce mean translation error by 44% to 46% and median translation error up to 33%. Mean rotation accuracy improves by 28% to 50%, while median rotation accuracy improves up to 22%. MS-T^{vio}-preserved RPs reduce mean translation error by 34% to 66% and median translation error by 11% to 32%. Mean rotation accuracy improves by 30% to 75%, and median rotation error improves by 11% to 35%. The improvement for indoor scenes follows a similar pattern. Table III shows that the percentage of each accuracy level is greatly increased by selecting the RPs that amount for 48.5% to 76.2% of the total PN pose estimations and 46.3% to 78.7% of the total MS-T pose estimations. $APE_{\langle apr, GT \rangle}$ and $AOE_{\langle apr, GT \rangle}$ are greatly reduced as shown in Table II. These outcomes indicate that a significant portion of predictions with large errors contribute to the lower accuracy. Our framework effectively identifies RPs using VIO.

Optimized Poses: Tables II and III provide confirmation that a portion of the pose estimates that fail the relative pose checker in the *pose optimization* stage exhibit larger errors compared to the median and mean pose errors of all predictions. The difference in accuracy is even more pronounced when compared to the selected RPs. These unreliable poses significantly impact overall accuracy. For outdoor scenes, PN^{vio} improves mean translation accuracy by 48% to 51% and median translation accuracy by up to 32%. It also enhances mean rotation accuracy by 60% to 82% and median rotation accuracy by 39% to 63%. Similarly, MS-T^{vio} improves mean translation accuracy by 60% to 61% and median translation accuracy by 13% to 53%. It also enhances mean rotation accuracy by 58% to 90% and median rotation accuracy by 23% to 64%. Pose estimates below the low accuracy level are significantly reduced for both PN^{vio} and MS-T^{vio}. In the Church and Bar scenes, no optimized pose estimates of MS-T^{vio} exceed 5 meters and 10 degrees. For indoor scenes, 20.2% and 36.7% of PN pose estimates and 14.2% to 44.7% of the MS-T pose estimates that do not pass the relative pose threshold in the *pose optimization* stage have larger error compared with the median and mean pose error of all predictions. The optimized accuracy has experienced a substantial improvement.

Total results: As shown in Table III, less than 10 percent of the predictions in test set are filtered out as unreliable predictions in *alignment* stage except for atrium. MobileARLoc effectively optimizes unreliable predictions in *pose optimization* stage, which leads to a much higher mean accuracy than original APRs on both translation and rotation for all three scenes. Figure 2 and Figure 3 show MobileARLoc improves accuracy by reducing the incidence of outliers with large error and the noisiness of APR predictions.

B. Analysis

Table III show that PN and MS-T have predictions that are very inaccurate with large errors more than 5 meters and 10 degrees in both outdoor and indoor scenes. By calculating the reference pose with RPs identified by VIO system, unreliable predictions are optimized, resulting in significant improve-

TABLE III

PERCENTAGE (%) OF RPS, OPTIMIZED POSES AND TOTAL POSES PREDICTED WITH HIGH (0.25m, 2°), MEDIUM (0.5m, 5°), AND LOW (5m, 10°) ACCURACY [23] (HIGHER IS BETTER). THE VALUE IN PARENTHESES REPRESENTS THE RATIO (%) OF RPS, OPT. POSES, AND RPS + OPT. POSES IN THE TEST SET.

Only Reliable Predictions					
Dataset	Scenes	PN	PN ^{opt} (ours)	MS-T	MS-T ^{opt} (ours)
Outdoor	Square	3.2/18.4/87	5.6/27.2/96.4 (41.8)	2.9/17.9/81.1	3.8/26.9/91.9 (43.2)
	Church	2.2/21.9/82.1	3.8/33.4/93.0 (43.5)	6.9/29.7/79.0	12.4/48.7/96.1 (41.6)
	Bar	4.1/31.7/89.5	6.4/39.6/7 (51.3)	8.5/34.6/90.8	12.2/46.4/97.4 (51)
Only Optimization					
Outdoor	Square	1.8/13.7/79 (49.9)	4.1/20.8/94.3 (49.9)	2/10.1/72 (53.1)	1.5/12.7/97.8(53.1)
	Church	1.1/12.7/73.6 (55.6)	3.6/21.9/97.5 (55.6)	3.1/15.3/66.3 (57.4)	1.8/15.7/100 (57.4)
	Bar	1.8/19.5/81 (45.9)	3.1/16.8/100 (45.9)	4.8/23.0/85.3 (44.6)	4/15.8/100 (44.6)
Reliable Predictions + Optimization					
Outdoor	Square	3.2/18.4/87.0	4.8/23.7/95.3 (91.7)	2.9/17.9/81.1	2.5/19.1/95.1 (96.3)
	Church	2.2/21.9/82.1	3.7/27.0/95.5 (99.1)	6.9/29.7/79.0	6.3/29.6/98.3 (97)
	Bar	4.1/31.7/89.5	4.7/30.7/98.3 (97.2)	8.5/34.6/90.8	8.4/32.1/98.6 (92)
Only Reliable Predictions					
Dataset	Scenes	PN	PN ^{opt} (ours)	MS-T	MS-T ^{opt} (ours)
Indoor	Stairs	5.4/48.2/87.4	7.7/51.4/93.7 (64)	18.5/58.1/86.9	22.9/69.3/94.1 (68.9)
	Office	3.8/31.0/72.8	4.3/36/80.8 (76.2)	7.1/46.0/80	8.8/54.7/90.2 (78.7)
	Atrium	0/5.4/71	0/8.4/86 (48.5)	0.4/7.3/66.7	1/13.7/88.7 (46.3)
Only Optimization					
Indoor	Stairs	1.3/42.9/75.3 (36.7)	0/53.2/100 (36.7)	9.1/34.8/71.2 (29.7)	34.8/69.7/100 (29.7)
	Office	1.6/14.8/46.9 (20.2)	9.4/18/89.8 (20.2)	1.1/15.6/41.1 (14.2)	23.3/51.1/94.4 (14.2)
	Atrium	0/2.5/64.6 (35.8)	0/5.1/75.3 (35.8)	0/1.5/46.7 (44.7)	0/5.4/100 (44.7)
Reliable Predictions + Optimization					
Indoor	Stairs	5.4/48.2/87.4	5/52.1/95.9 (90.7)	18.5/58.1/86.9	26.5/69.4/95.9 (98.6)
	Office	3.8/31.0/72.8	5.4/32.2/82.7 (96.4)	7.1/46.0/80	11.5/41.1/90.8 (92.9)
	Atrium	0/5.4/71.0	0/6.9/81.5 (84.3)	0/7.3/66.7	0.7/9/94.3 (91)

ment. Outdoor, our framework improves the accuracy of MS-T by 47% on mean translation error and 66% on mean rotation error over all scenes average. Indoor, it improves the accuracy of MS-T by 50% on mean translation error and 49% on mean rotation error on all scenes average. Compared to the median accuracy, our method provides a greater improvement in mean accuracy. This is because the accuracy of our optimization depends on the accuracy of the reference pose, which is basically about the same error as the median accuracy. Outdoor datasets yield better results due to the lower accuracy of the APR in indoor scenes. This leads to misclassification of inaccurate poses during *alignment* (case (3) in Section III-B), making the calculation of the reference pose challenging and introducing further inaccuracies during *pose optimization*.

VIII. CONCLUSION

This paper introduces MobileARLoc, a framework that combines an APR with a local VIO tracking system to improve the accuracy and stability of localization for markerless mobile AR. The VIO evaluates and optimizes the APR's accuracy while the APR corrects VIO drift, resulting in improved positioning. We evaluate MobileARLoc through dataset simulations. MobileARLoc improves the position accuracy by up to 50% and rotation by up to 66% for different APRs. The mobile app can perform pose estimation in less than 80 ms with minimal storage and energy consumption. Due to its low system footprint, high accuracy, and robustness, MobileARLoc enables pervasive markerless mobile AR at a large scale.

REFERENCES

[1] 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/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8092–8101.

[2] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[3] 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.

[4] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han, "Large-scale image retrieval with attentive deep local features," in *IEEE International Conference on Computer Vision*, 2017, pp. 3456–3465.

[5] A. Moreau, T. Gilles, N. Piasco, D. Tsishkou, B. Stanculescu, and A. de La Fortelle, "Imposing: Implicit pose encoding for efficient visual localization," in *IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 2892–2902.

[6] T. Braud, Z. Pengyuan, J. Kangasharju, and H. Pan, "Multipath computation offloading for mobile augmented reality," in *2020 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2020, pp. 1–10.

[7] C. B. Fernandez, T. Braud, and P. Hui, "Implementing gdpr for mobile and ubiquitous computing," in *23rd Annual International Workshop on Mobile Computing Systems and Applications*, 2022.

[8] Y. Shavit, R. Ferens, and Y. Keller, "Learning multi-scene absolute pose regression with transformers," in *IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2733–2742.

[9] T. Sattler, Q. Zhou, M. Pollefeys, and L. Leal-Taixe, "Understanding the limitations of cnn-based absolute camera pose regression," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[10] T. Scargill, G. Premsankar, J. Chen, and M. Gorlatova, "Here to stay: A quantitative comparison of virtual object stability in markerless mobile ar," in *2022 2nd International Workshop on Cyber-Physical-Human System Design and Implementation (CPHS)*. IEEE, 2022, pp. 24–29.

[11] S. Brahmhatt, 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.

[12] 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.

[13] A. Kendall and R. Cipolla, "Geometric loss functions for camera pose regression with deep learning," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5974–5983.

[14] S. Chen, X. Li, Z. Wang, and V. A. Prisacariu, "Dfnet: Enhance absolute pose regression with direct feature matching," in *ECCV 2022. Tel Aviv, Israel, October 23–27, 2022, Part X*. Springer, 2022.

[15] A. Kendall and R. Cipolla, "Modelling uncertainty in deep learning for camera relocalization," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016.

[16] Z. Huang, Y. Xu, J. Shi, X. Zhou, H. Bao, and G. Zhang, "Prior guided dropout for robust visual localization in dynamic environments," in *IEEE/CVF International Conference on Computer Vision*, 2019.

[17] A. Moreau, N. Piasco, D. Tsishkou, B. Stanculescu, and A. de La Fortelle, "Coordinet: uncertainty-aware pose regressor for reliable vehicle localization," in *IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022.

[18] M. Bui, T. Birdal, H. Deng, S. Albarqouni, L. Guibas, S. Ilic, and N. Navab, "6d camera relocalization in ambiguous scenes via continuous multimodal inference," in *ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Part XVIII 16*. Springer, 2020.

[19] F. Zangeneh, L. Bruns, A. Dekel, A. Pieropan, and P. Jensfelt, "A probabilistic framework for visual localization in ambiguous scenes," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 3969–3975.

[20] C. Gramkow, "On averaging rotations," *Journal of Mathematical Imaging and Vision*, vol. 15, no. 1-2, pp. 7–16, 2001.

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

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[23] 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.