Large-Scale Benchmarking of Vision-Based Indoor Localization Using Absolute Pose Regression

1st Jun Yu†, 2nd Yu Yang†, 3rd Vinod Namboodiri†,‡
†*Dept. of Computer Science & Engineering*, ‡*Dept. of Community & Population Health*
Lehigh University, Bethlehem, USA
{juy220, yuy421, vin423}@lehigh.edu

2025 International Conference on Indoor Positioning and Indoor Navigation (IPIN) | 979-8-3315-5680-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/IPIN66788.2025.11213320

***Abstract*—Navigating indoor environments is challenging due to the absence of satellite signals. Existing solutions often rely on auxiliary sensors such as Bluetooth or LiDAR to estimate user location and provide turn-by-turn navigation cues. However, these approaches typically require the deployment of beacons or transponders throughout a building, increasing installation costs and limiting scalability. In contrast, computer vision-based methods operate independently using only a camera on the user's device, capturing richer visual information than traditional sensors. This paper focuses on benchmarking absolute pose regression (APR) techniques, which predict a camera's absolute pose (position and orientation) from a single image in an end-to-end manner. Unlike conventional vision-based approaches, APR does not rely on matching to a 3D model or performing explicit geometric computations such as PnP. While prior work has largely evaluated APR in small-scale environments, we introduce a large-scale image dataset collected in a multi-floor research facility and conduct a comprehensive evaluation of state-of-the-art APR methods. Our results show that APR can achieve robust sub-meter localization accuracy in complex indoor settings, supporting its use as a practical backend for vision-based indoor navigation systems.**

***Index Terms*—dataset, benchmarking, absolute pose regression (APR), deep neural networks, computer vision**

I. Introduction

Indoor navigation remains a complex challenge due to the unavailability of satellite signals, such as GPS, for indoor localization. The dynamic and variable nature of indoor environment scenes further complicates the positioning [1]. While technologies like bluetooth low energy (BLE) beacons [2], Wi-Fi [3], radio frequency identification (RFID) [4], and ultra-wideband (UWB) [5] can improve localization accuracy, they typically require extensive hardware deployment and maintenance, limiting their scalability and practicality. These technologies generally work by measuring signal strength, time-of-flight, or proximity to fixed reference points to estimate a user's position within an indoor space. In contrast, computer vision-based techniques, as infrastructure-free alternatives, offer greater flexibility by leveraging visual data and have recently attracted growing attention [6]. Within this line of research, accurately estimating the camera pose—the position and orientation of the camera within a known environment—is a fundamental step in indoor localization and navigation, as it enables the system to determine the user's location relative to the surrounding space. Researchers have explored retrieval-based and 3D-2D matching approaches as backend localization models for practical indoor navigation systems. These two categories involve different trade-offs in hardware dependence, prediction accuracy, inference speed, and robustness. Retrieval-based methods estimate the camera pose by comparing a query image with a database of keyframes with known poses, selecting either the most similar pose or a weighted average of the top matches [7]. While conceptually simple, these methods require large storage for the image database and suffer from constrained query speeds as the dataset grows. To address these limitations, 3D reconstruction-based approaches incorporate geometric information by building spatial models from sparse features [8] or dense point clouds [9]. This added structure enables more accurate and scalable pose estimation, as the query image is matched to a 3D representation rather than a flat image database. Camera pose is then refined through geometric computations and 2D-3D matching [10]. However, these approaches typically involve significant memory overhead and slower processing, limiting their feasibility for real-time deployment.

Absolute pose regression (APR) [11] addresses these challenges by directly predicting the camera pose from a single image through deep neural networks. It offers fast inference and low memory demands, making it suitable for deployment on lightweight client devices. While there is a slight compromise in accuracy due to the camera relocalization, APR may provide sufficient precision for real-time navigation. Furthermore, its learning paradigm aligns well with other deep learning-based computer vision tasks such as image classification, semantic segmentation, and depth estimation [12]. However, existing APR methods have primarily been benchmarked on the outdoor urban dataset Cambridge Landmarks [11] and the small-scale indoor dataset 7 Scenes [13]. To date, there is little evidence or prior research demonstrating that APR can reliably operate in large and dynamic indoor environments. Its robustness remains uncertain, particularly in settings where scenes change over time and include numerous movable objects, such as people. In this paper, we extend the evaluation of APR methods to a complex, multi-floor indoor environment and demonstrate their potential as scalable and effective backend models for vision-based indoor navigation systems.

The main contribution of this paper is the creation of a large-scale image dataset from a four-floor research building,

This work was partially supported by the U.S. NSF through award #2345057. Yu Yang was supported in part by the U.S. NSF awards #2246080, #2318697, and #2427915.

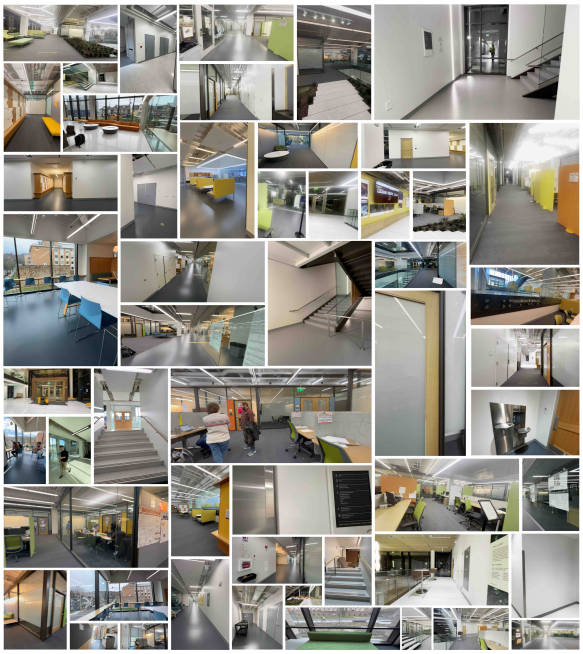

Fig. 1. Examples of raw images in our dataset collected from HST, showcasing typical indoor environments featuring dynamic elements, inadequate signage, and complex layouts.

which captures diverse and dynamic indoor conditions, and the demonstration that APR methods can achieve robust sub-meter localization accuracy in such a complex, real-world setting. This work lays the groundwork for deploying practical smartphone-based navigation systems. To that end, the dataset is designed to be scalable and easily replicable, relying solely on a commodity off-the-shelf (COTS) phone camera and requiring minimal manual effort for ground-truth annotation. All image positions are aligned with widely available digital floor plans to support visualization during navigation. In this paper, we demonstrate that APR models can accurately localize images on floor plans with high precision, even in dynamic indoor conditions. Our benchmarks showcase the potential of APR for indoor navigation, highlighting its robustness to diverse environments, scalability across scene sizes, and fast end-to-end inference—producing pose predictions from a single image within milliseconds. With the release of this dataset, we aim to foster further research into image-centric, real-time indoor navigation based on single-image inference.

II. Dataset Construction

The goal of this dataset is to support data-driven models aimed at solving the localization problem for future indoor navigation applications. Achieving this requires a large-scale dataset that accurately captures real-world indoor environments, including transient changes caused by moving people or furniture. To ensure practical applicability across different spaces, the data collection and annotation process must be scalable and minimize human labor, making the solution both affordable and replicable.

Algorithm 1 Camera Pose (6DoF) Annotation

Input: Video set $\mathcal{V} = \{V_1, \cdots, V_N\}$ by mobile cameras.
Output: Geo-registered ground-truth of camera poses (positions and orientations) on the floor plan for extracted video frames.

Initialize frame interval Δf. ▷ $\Delta f \approx 16$.
for $n = 1, \cdots, N$ **do**
  Sample the image sequences $\mathcal{I}_n = \{I_1, \cdots, I_{M_n}\}$ from video V_n every Δf frames.
  repeat
    Run SfM algorithm on image set $\mathcal{I}$. ▷ *Using COLMAP.*
    if Point clouds align **then**
      Obtain camera pose for image sequences $\mathcal{I}$.
    else
      Sample more image frames $\mathcal{I}'$ at breaking points. ▷ $\Delta f' = \frac{1}{3}\Delta f$
      Data augmentations: $\mathcal{I} = \mathcal{I} \cup \mathcal{I}'$.
    end if
  until Point clouds align.
end for

A. Collection

We employ simple smartphone cameras to capture videos (as opposed to LiDARs or more specialized cameras used in [7], [14]), addressing a significant need for images while streamlining the process to reduce human labor. This process has the potential for further automation using roaming robots in the future. In our study, we collected 400 videos in total, each lasting 2 to 4 minutes, across four floors of Lehigh's largest research facility—the Health, Science, and Technology (HST) Building[1]. We set the frame extraction interval to range between 0.2 to 0.3 seconds to conduct a dataset of approximately $300,000$ images. It should be noted that our simplistic way of expanding the dataset, either by recording longer videos or more videos, or by setting a narrower time interval to extract frames, facilitates easy and scalable deployment on other buildings as well. This adaptability makes it suitable for deployment in larger spaces or applications requiring higher localization accuracy. To enhance the variety of our dataset, we recorded videos at different times of day, with recording dates spanning from December 2023 to June 2024. Figure 1 displays example images from our dataset.

To enhance the robustness of algorithms developed from this dataset, videos were captured using four distinct smartphone models, in both landscape and portrait orientations. These recordings were made in two different holding ways: by human hand and using a smartphone gimbal stabilizer. To comprehensively capture the variability of indoor environments, recordings spanned various times of the day—sunrise, morning, noon, afternoon, sunset, and evening. Videos recorded after April are designated as testing set to ensure that models developed using this dataset exhibit generalizability. We acknowledge that despite our efforts to align this dataset closely with real-world scenarios, the dataset biases still exist and may be captured by models [15]. For example, our dataset lacks the coverage of the fall season.

B. Camera Pose Ground Truth

There are multiple ways to annotate camera images with accurate 6-DoF poses, but most rely on professional equipment

[1]HST is the largest building Lehigh has ever built. For more information, please click this link.

TABLE I
SUMMARY OF CAMERA RELOCALIZATION DATASETS. THIS TABLE EXCLUSIVELY COMPARES DATASETS THAT ARE PUBLICLY AVAILABLE.

Dataset	Environment	Device	# Train / Test	Resolution	Scope	Ground Truth Tool	Error Level
Dubrovnik 6K [16]	Outdoor	–	6K / 0.8K	—	1.5×1.5 km^2	SIFT Matching	$\sim$ 10 m
7-Scenes [13]	Indoor	Kinect depth camera	16K / 17K	640×480	4×3 m^2	KinectFusion [17]	$<$ 10 cm
Cambridge [11]	Outdoor	Smartphone camera	8.4K / 4.8K	1920×1080	500×100 m^2	MVS [18]	$<$ 10 dm
12-Scenes [19]	Indoor	Structure.io depth sensor with iPad	240K / 6.7K	1296×968	80 m^3	VoxelHashing [20]	$\sim$ 10 cm
TUM-LSI [21]	Indoor	NavVis M3 trolley	875/220	4592×3448	5575 m^2	SLAM	$<$ 10 cm
InLoc [9]	Indoor	Faro 3D laser scanner	10K / 0.4K	$1600 \times 1,200$	186 m^2	LiDAR + Manual	$\sim$ 10 dm
LaMAR [22]	Indoor & Outdoor	Microsoft Hololens 2, Smartphone, iPad, NavVis M6 or VLX backpack	—	640×480	45,000 m^2	LiDAR + SfM + VIO [23]	$<$ 10 cm
360Loc [24]	Indoor & Outdoor	Velodyne lidar with 360° camera	9.3K / –	6144×3072	105×70 m^2	LiDAR + VIO	$<$ 10 cm
HST (Ours)	Indoor	Smartphone camera	212K / 88K	3840×2160	40×90 m^2	COLMAP [8]	$<$ 10 cm

such as the NavVis VLX 3D scanning backpack [21] or involve complex, often closed-source optimization pipelines that are difficult to reproduce [24]. Table I summarizes existing camera relocalization datasets across indoor and outdoor settings. Notably, the Cambridge Landmarks dataset [11] is the only one that employed a widely available device and open-source tools to generate 6-DoF ground truth, achieving decimeter-level accuracy outdoors. Inspired by this, we emphasize the importance of low-cost, reproducible methods for ground-truth annotation in indoor environments.

To this end, we initiate the process by executing **Algorithm 1** to derive camera poses in world coordinates automated by COLMAP[2]. Since the initial coordinate origins are arbitrarily set by the 3D reconstruction process for different videos, we redefine the world coordinate system by selecting the top-left pixel of the floor plan (following OpenCV conventions) as the unified origin. This geo-registration process involves manually anchoring approximately 10 reference images, which then enables real-time alignment of all subsequent images using COLMAP.

C. Annotation of PoIs

We assume the digital floor plan contains all PoIs necessary for navigation. To label regions of interest, we apply a sliding bounding box over the floor plan, as illustrated in Fig. 2. Both camera poses and PoIs are then defined in pixel coordinates and orientation angles relative to the floor plan, enabling the association of predicted poses with nearby PoIs without requiring physical tags.

III. BENCHMARKING

To evaluate the generalization ability across ever-changing indoor environments, we utilize images captured prior to April 15th as training dataset and images captured after May 1st as testing dataset throughout the benchmarking process.

A. Preliminary

For clarity and consistency, we represent the camera pose by a 2-element tuple $[\boldsymbol{x}, \boldsymbol{q}]$ according to [11], where $\boldsymbol{x} \in \mathbb{R}^3$ defines the position of camera center in 3D Cartesian coordinates and $\boldsymbol{q} \in \mathbb{R}^4$ is the unit quaternion encoding its orientation.

[2]COLMAP is available at this link.

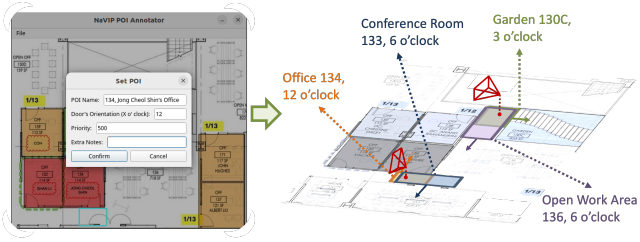

Fig. 2. The screenshot on the left displays the PoI annotator. Each PoI is annotated by selecting a rectangular bounding box (highlighted in cyan) on the floor plan and entering the PoI's name. The orientation of the door is determined using the clock position method, and priority numbers are assigned to manage overlapping PoIs at specific locations. This software allows users to easily add, delete, and reload PoIs. The right illustration shows how camera poses connect with PoI regions on a floor plan. Corresponding responses will be triggered when camera pose predictions fall within specific PoI regions.

Other variations focus on orientation representations, with BranchPoseNet [25] employing an Euler angle representation and MapNet [26] using a logarithm of the unit quaternion.

Our purpose is to directly regress the camera pose $[\hat{\boldsymbol{x}}, \hat{\boldsymbol{q}}]$ from a single monocular image I using the trained function f. The standard objective loss function is defined as follows:

$$\mathcal{L}(I) = \|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_2 + \beta \cdot \left\| \hat{\boldsymbol{q}} - \frac{\boldsymbol{q}}{\|\boldsymbol{q}\|_2} \right\|_2, \tag{1}$$

where $[\boldsymbol{x}, \boldsymbol{q}] = f(I)$ represents the values predicted by our models. Notably, β is a hyperparameter introduced to balance the learning scales between position and orientation. Learnable PoseNet [27] captures homoscedastic uncertainty [28] between two tasks and omit β as

$$\mathcal{L}(I) = e^{-s_x} \cdot \|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_2 + e^{-s_q} \cdot \left\| \hat{\boldsymbol{q}} - \frac{\boldsymbol{q}}{\|\boldsymbol{q}\|_2} \right\|_2 + s_x + s_q, \tag{2}$$

where x_x and s_q are both learnable and only an approximate initial guess is required.

B. Settings

To ensure a fair comparison in benchmarking camera relocalization, we employed identical CNN architectures—ResNet-34 [38] and MobileNet-V3 [39]—across 14 models ranging from the pioneering PoseNet [11] to its latest advancements [37]. All the models above were trained for 200 epochs. Specifically, for the PoseNet-series models, we utilized the loss function defined in Eq.(1), setting β to e^5

TABLE II

MEAN AND MEDIAN ERRORS↓ OF MODELS ACROSS FOUR FLOORS (BASEMENT, LOWER LEVEL, LEVEL 1, AND LEVEL 2) IN OUR DATASET. BENCHMARKING IS LIMITED TO MODELS WITH PUBLICLY AVAILABLE CODE.

Model	Backbone	Basement		Lower Level		Level 1		Level 2	
		Mean	Median	Mean	Median	Mean	Median	Mean	Median
PoseNet [11]	ResNet-34	0.52m, 5.50°	0.42m, 4.52°	0.96m, 7.14°	0.71m, 5.70°	0.91m, 7.65°	0.52m, 6.21°	1.12m, 6.78°	0.72m, 5.99°
	MobileNet-V3	0.60m, 5.78°	0.52m, 5.01°	0.95m, 7.10°	0.69m, 5.59°	0.90m, 7.94°	0.50m, 6.34°	1.11m, 6.91°	0.73m, 6.04°
Bayesian PoseNet [29]	ResNet-34	0.57m, 6.88°	0.51m, 5.70°	1.02m, 8.34°	0.75m, 6.04°	0.98m, 9.04°	0.61m, 7.38°	1.09m, 7.34°	0.71m, 6.51°
	MobileNet-V3	0.62m, 6.99°	0.60m, 5.98°	1.05m, 8.73°	0.79m, 6.48°	1.01m, 8.93°	0.72m, 6.76°	1.09m, 7.52°	0.72m, 6.58°
LSTM-PoseNet [21]	ResNet-34	0.49m, 6.20°	0.42m, 5.13°	0.92m, 7.43°	0.74m, 5.97°	0.81m, 7.54°	0.51m, 5.49°	0.92m, 6.17°	0.59m, 4.73°
	MobileNet-V3	0.52m, 6.49°	0.41m, 6.24°	0.94m, 7.83°	0.73m, 5.99°	0.84m, 7.62°	0.52m, 5.73°	0.95m, 6.21°	0.60m, 4.41°
Learnable PoseNet [27]	ResNet-34	0.39m, 3.41°	0.33m, 2.19°	0.76m, 3.24°	0.50m, 3.19°	0.70m, 4.30°	0.46m, 3.84°	0.87m, 3.90°	0.49m, **2.41**°
	MobileNet-V3	0.42m, 3.90°	0.34m, 2.81°	0.78m, **3.20**°	0.53m, 3.14°	0.75m, 4.76°	0.47m, 3.90°	0.91m, 4.74°	0.46m, 3.82°
Geometric PoseNet [27]	ResNet-34	0.41m, 4.45°	0.33m, 3.01°	0.80m, 3.41°	0.55m, 3.24°	0.73m, 4.28°	0.47m, 3.87°	0.90m, 5.21°	0.57m, 3.43°
	MobileNet-V3	0.46m, 3.89°	0.39m, 2.80°	0.83m, 3.40°	0.56m, 3.19°	0.77m, 4.91°	0.49m, 4.19°	0.94m, 5.32°	0.58m, 3.49°
Hourglass PoseNet [30]	ResNet-34	0.45m, 5.95°	0.34m, 4.91°	0.88m, 5.47°	0.62m, 4.79°	0.81m, 6.23°	0.49m, 4.02°	0.85m, 6.87°	0.49m, 3.96°
	MobileNet-V3	0.53m, 6.18°	0.48m, 6.03°	0.89m, 5.86°	0.60m, 4.79°	0.83m, 6.48°	0.50m, 4.18°	0.94m, 8.31°	0.52m, 4.37°
BranchNet-Euler6 [25]	ResNet-34	0.43m, 5.99°	0.32m, 4.72°	0.79m, 4.00°	0.48m, 3.25°	0.94m, 9.21°	0.65m, 6.27°	1.04m, 7.26°	0.78m, 6.13°
	MobileNet-V3	0.50m, 6.23°	0.41m, 4.82°	0.84m, 4.31°	0.51m, 3.36°	0.98m, 9.74°	0.66m, 6.40°	1.16m, 8.17°	0.82m, 7.10°
MapNet [31]	ResNet-34	0.39m, 3.41°	0.31m, 2.71°	0.70m, 3.75°	0.42m, 3.10°	0.65m, 4.58°	0.44m, 3.90°	0.82m, 5.43°	0.48m, 3.97°
	MobileNet-V3	0.40m, **3.37**°	0.29m, 2.65°	0.74m, 3.76°	0.43m, 3.22°	0.66m, 4.79°	0.41m, 4.21°	0.86m, 4.27°	0.50m, 3.21°
MSPN [32]	ResNet-34	0.39m, 3.40°	0.30m, 2.68°	0.68m, 3.70°	0.42m, 3.13°	0.69m, 5.17°	0.47m, 3.96°	0.91m, **3.71**°	0.48m, 2.53°
	MobileNet-V3	0.41m, 3.38°	0.30m, 2.58°	0.73m, 3.94	0.45m, 3.27°	0.71m, 5.32°	0.48m, 3.87°	0.92m, 4.14°	0.49m, 3.04°
Direct-PoseNet [33]	ResNet-34	0.35m, 3.94°	0.27m, 3.10°	0.69m, 3.77°	0.44m, 2.69°	0.63m, 4.74°	0.41m, 3.90°	0.84m, 3.94°	0.46m, 2.97°
	MobileNet-V3	0.39m, 3.49°	0.28m, 2.88°	0.70m, 3.71°	0.45m, 2.71°	0.61m, 4.82°	0.42m, 3.89°	0.88m, 4.10°	0.45m, 3.04°
MS-Transformer [34]	ResNet-34	**0.35m**, 4.47°	**0.26m**, 3.78°	0.63m, 3.74°	0.43m, 2.50°	**0.60m**, 4.66°	**0.41m**, 3.84°	0.83m, 4.21°	0.43m, 2.76°
	MobileNet-V3	0.44m, 5.26	0.36m, 4.19°	0.65m, 3.89°	**0.41m, 2.43**°	0.61m, **4.06**°	0.45m, **3.17**°	0.85m, 4.18°	0.43m, 2.96°
PAE [35]	ResNet-34	0.40m, 3.99°	0.27m, 2.87°	0.71m, 3.79°	0.45m, 2.57°	0.71m, 5.23°	0.49m, 4.15°	0.96m, 4.73°	0.51m, 3.25°
	MobileNet-V3	0.41m, 4.10°	0.27m, 2.93°	0.73m, 3.80°	0.47m, 2.63°	0.71m, 5.23°	0.49m, 4.15°	0.98m, 4.69°	0.52m, 3.09°
DFNet [36]	ResNet-34	0.36m, 3.49°	0.28m, 2.58°	**0.60m**, 3.80°	0.43m, 2.78°	0.64m, 4.82°	0.43m, 3.85°	0.81m, 4.37°	0.43m, 2.81°
	MobileNet-V3	0.40m, 3.56°	0.34m, 3.16°	0.61m, 3.72°	0.42m, 2.55°	0.67m, 5.01°	0.47m, 3.91°	**0.80m**, 4.26°	0.45m, 2.90°
marepo [37]	ResNet-34	0.37m, 3.40°	0.27m, **2.08**°	**0.60m**, 3.68°	0.43m, 2.46°	0.62m, 4.17°	**0.41m**, 3.26°	**0.80m**, 4.03°	**0.42m**, 2.65°
	MobileNet-V3	0.42m, 3.88°	0.35m, 4.01°	0.62m, 3.84°	0.45m, 2.79°	0.63m, 4.89°	0.43m, 3.65°	0.84m, 4.14°	0.44m, 2.91°

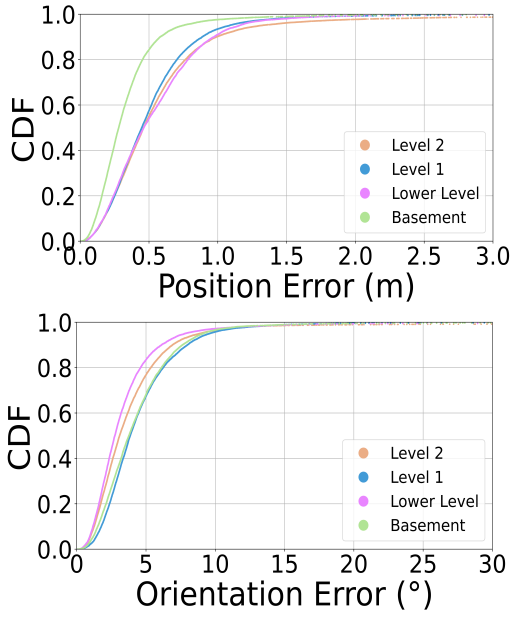

Fig. 3. CDF of errors on MS-Transformer across four floors.

across all four floors[3]. In the Bayesian PoseNet, uncertainty is incorporated only before the layers with randomly initialized weights, following [29]. LSTM-PoseNet models have the hidden size of all LSTM units set at 256 to achieve optimal results [21]. Learnable PoseNet starts with initial guesses of s_x and s_q set to 0 and -5, respectively, for all scenes [27], and then Geometric PoseNet continues training these models using geometric reprojection data to balance positional and rotational errors per image. We employed feature map concatenation in ResNet-34 and element-wise summation in MobileNet-V3 to combine features from the front layers and achieve the optimal results in Hourglass PoseNet [30]. For BranchNet-Euler6 [25], the networks are split before the final convolutional block to facilitate multi-task learning. MapNet [31] configurations avoid updating model weights with unlabeled data to maintain fairness in APR comparisons. Both MapNet and MSPN [32] adopt log-quaternion for rotation representation. Direct-PoseNet maintains a direct matching ratio of photometric difference to the loss function Eq.(1) at 3/7 [33]. MS-Transformer [34] and marepo [37] replaces only the convolutional backbones to extract activation maps while preserving the encoder-decoder architecture of Transformers. PAE embeds $\boldsymbol{x}$ and $\boldsymbol{q}$ using Fourier Features with expanded dimensions of 12 ($L = 6$ in [35, Eq.(5)]). For DFNet, results are confined to single-frame APR [36].

[3]Specifically tuning β for each floor may yield better results.

C. Results

Table II shows the mean and median errors for both camera positions and orientations. Notably, even the pioneering work of PoseNet [11] from 2015 can achieve sub-meter accuracy, meeting the requirements for indoor navigation. We observe

that the best performance in average is achieved by MS-Transformer [34], which benefits from its model capacity enabled by the use of Transformers and a multi-scene mixing training strategy. Figure 3 illustrates the cumulative distribution function (CDF) of MS-Transformer's predictions regarding mean errors in positions and orientations across four floors.

IV. Discussion

A. Potential Applications

1) APR methods development: Popular datasets commonly used for benchmarking APR methods include 7-Scenes [13] and Cambridge Landmarks [11]. These datasets, however, are limited in scope and size, which can lead to the overfitting phenomenon already observed in previous research [11], [27]. This limitation complicates the performance assessment of new proposed models, particularly in an era dominated by deep learning and foundation models. For instance, Bayesian PoseNet [29] demonstrates inferior performance compared to the vanilla PoseNet on HST. This discrepancy arises because any dropout rate applied to PoseNet constrains its capacity rather than serving its intended purpose of regularization.

2) Floor plan navigation development: Each image in our dataset is geo-registered to a corresponding floor plan, facilitating the development of learning-based methods that utilize only RGB images for localization. While LiDAR-based localization techniques have been explored extensively in recent research [40]–[43], their practical application is often constrained by the hardware capabilities of commonly used mobile devices. In contrast, we anticipate that purely image-based floor plan navigation will become more viable with the availability of this large indoor navigation dataset. This could pave the way for more accessible and widely deployable navigation solutions that leverage visual data alone.

3) Deployment and test anywhere.: We will provide comprehensive details regarding the setup and organization of our construction within the Lehigh HST building. The data collection process requires only a mobile phone, and to simplify this process, we will also release the corresponding code, including data pre-processing and annotation. Our team has also tested APR methods in other buildings and deployed the trained models as the localization engine for a smartphone-based navigation app. We encourage other research groups seeking an affordable and intelligent indoor navigation solution to adopt and adapt this pipeline for use in their own environments.

B. Limitations

1) How to merge 3D reconstruction losslessly?: In our dataset, we utilize COLMAP to reconstruct the 3D models of indoor environments along each video path. Given that indoor environments can be ever-changing, there currently lacks a robust algorithm to effectively merge these 3D point clouds from different time. As a workaround, we utilize human supervision to pinpoint several anchor images directly onto the floor plan. These annotated points facilitate the geo-registration process built in COLMAP to transform the coordinates of all 3D point clouds. While this alignment method is not lossless–resulting in an inevitable system error in camera poses of approximately 0.5 meters–we will release both the sparse and dense models generated by COLMAP, before and after geo-registration, to facilitate further research in this area. This contribution is expected to further enhance the accuracy of positioning system in indoor navigation.

2) Handling multi-floor transitions in indoor navigation: While our experimental evaluation primarily focuses on a single floor, practical indoor navigation requires seamless localization across multiple levels. In our dataset, each floor is independently reconstructed and geo-registered to its corresponding floor plan. Although camera height information is recorded (currently set to zero per floor), vertical connectivity features such as stairwells or elevators could be exploited for continuous height tracking. Nevertheless, repetitive architectural features across floors introduce additional challenges for vision-based localization models. As a practical solution, we treat floor transitions as discrete localization resets, where the system detects a change in floor level (e.g., through visual or sensor cues) and reinitializes pose estimation within the new floor's localization system. While this approach incurs some overhead from backend model switching, it remains effective given the lack of robust multi-floor localization algorithms. Future work should aim to develop integrated methods for continuous, multi-floor spatial mapping to further improve indoor navigation performance.

V. Conclusion

In this paper, we introduced a large-scale image dataset collected from a four-floor research facility to support the development and evaluation of vision-based indoor localization methods. Departing from traditional retrieval or matching-based approaches, our work benchmarks APR techniques that directly predict camera positions and orientations from single images using deep neural networks. Through comprehensive experiments, we demonstrated that APR models can achieve robust sub-meter localization accuracy in complex, real-world indoor environments, validating their potential for scalable and lightweight navigation systems. Our dataset emphasizes low-cost and replicable data collection, relying solely on COTS phone cameras and minimizing manual annotation via 3D reconstruction. Camera positions are aligned with digital floor plans to fully leverage the rich spatial information associated with PoIs. The dataset enables rigorous benchmarking across dynamic scenes and varied architectural configurations, making it a valuable resource for the research community. Future work will extend the dataset to additional buildings with more diverse layouts, while also improving annotation precision through unsupervised learning and reinforcement learning from human feedback (RLHF). The trained APR model will serve as the backend for a smartphone-based navigation app, which will be evaluated through extensive user studies. We hope this dataset catalyzes further research in image-based indoor navigation using end-to-end learning approaches.

REFERENCES

[1] Zeev Volkovich, Elena V Ravve, and Renata Avros. Indoor navigation in facilities with repetitive structures. *Sensors*, 24(9):2876, 2024.

[2] Seyed Ali Cheraghi, Vinod Namboodiri, and Laura Walker. Guidebeacon: Beacon-based indoor wayfinding for the blind, visually impaired, and disoriented. In *2017 IEEE PerCom*, pages 121–130. IEEE, 2017.

[3] Roshan Ayyalasomayajula, Aditya Arun, Chenfeng Wu, Sanatan Sharma, Abhishek Rajkumar Sethi, Deepak Vasisht, and Dinesh Bharadia. Deep learning based wireless localization for indoor navigation. In *Proceedings of the 26th MobiCom*, pages 1–14, 2020.

[4] Tan Kim Geok, Khaing Zar Aung, Moe Sandar Aung, Min Thu Soe, Azlan Abdaziz, Chia Pao Liew, Ferdous Hossain, Chih P Tso, and Wong Hin Yong. Review of indoor positioning: Radio wave technology. *Applied Sciences*, 11(1):279, 2020.

[5] Fuhu Che, Qasim Zeeshan Ahmed, Pavlos I Lazaridis, Pradorn Sureephong, and Temitope Alade. Indoor positioning system (ips) using ultra-wide bandwidth (uwb)—for industrial internet of things (iiot). *Sensors*, 23(12):5710, 2023.

[6] Dawar Khan, Zhanglin Cheng, Hideaki Uchiyama, Sikandar Ali, Muhammad Asshad, and Kiyoshi Kiyokawa. Recent advances in vision-based indoor navigation: A systematic literature review. *Computers & Graphics*, 104:24–45, 2022.

[7] Anbang Yang, Mahya Beheshti, Todd E Hudson, Rajesh Vedanthan, Wachara Riewpaiboon, Pattanasak Mongkolwat, Chen Feng, and John-Ross Rizzo. Unav: An infrastructure-independent vision-based navigation system for people with blindness and low vision. *Sensors*, 22(22):8894, 2022.

[8] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE CVPR*, pages 4104–4113, 2016.

[9] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. Inloc: Indoor visual localization with dense matching and view synthesis. In *Proceedings of the IEEE Conference on CVPR*, pages 7199–7209, 2018.

[10] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *IEEE transactions on PAMI*, 39(9):1744–1756, 2016.

[11] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE ICCV*, pages 2938–2946, 2015.

[12] Haoxuan Qu, Hossein Rahmani, Li Xu, Bryan Williams, and Jun Liu. Recent advances of continual learning in computer vision: An overview. *IET Computer Vision*, 19(1):e70013, 2025.

[13] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE conference on CVPR*, pages 2930–2937, 2013.

[14] Jennifer Palilonis, Caitlin Cambron, and Mianda Hakim. Challenges, tensions, and opportunities in designing app-based orientation and mobility tools for blind and visually impaired students. In *Int. Conf. on HCI*, pages 372–391. Springer, 2023.

[15] Zhuang Liu and Kaiming He. A decade's battle on dataset bias: Are we there yet? *arXiv preprint arXiv:2403.08632*, 2024.

[16] Yunpeng Li, Noah Snavely, Dan Huttenlocher, and Pascal Fua. Worldwide pose estimation using 3d point clouds. In *European conference on computer vision*, pages 15–29. Springer, 2012.

[17] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE ISMAR*, pages 127–136. IEEE, 2011.

[18] Yasutaka Furukawa, Brian Curless, Steven M Seitz, and Richard Szeliski. Towards internet-scale multi-view stereo. In *2010 IEEE computer society conference on CVPR*, pages 1434–1441. IEEE, 2010.

[19] Julien Valentin, Angela Dai, Matthias Nießner, Pushmeet Kohli, Philip Torr, Shahram Izadi, and Cem Keskin. Learning to navigate the energy landscape. In *2016 Fourth 3DV*, pages 323–332. IEEE, 2016.

[20] Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Marc Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (ToG)*, 32(6):1–11, 2013.

[21] Florian Walch, Caner Hazirbas, Laura Leal-Taixe, Torsten Sattler, Sebastian Hilsenbeck, and Daniel Cremers. Image-based localization using lstms for structured feature correlation. In *Proceedings of the IEEE ICCV*, pages 627–637, 2017.

[22] Paul-Edouard Sarlin, Mihai Dusmanu, Johannes L Schönberger, Pablo Speciale, Lukas Gruber, Viktor Larsson, Ondrej Miksik, and Marc Pollefeys. Lamar: Benchmarking localization and mapping for augmented reality. *Computer Vision–ECCV 2022*, 13667:686–704, 2022.

[23] Kenji Koide, Shuji Oishi, Masashi Yokozuka, and Atsuhiko Banno. General, single-shot, target-less, and automatic lidar-camera extrinsic calibration toolbox. In *2023 ICRA*, pages 11301–11307. IEEE, 2023.

[24] Huajian Huang, Changkun Liu, Yipeng Zhu, Hui Cheng, Tristan Braud, and Sai-Kit Yeung. 360loc: A dataset and benchmark for omnidirectional visual localization with cross-device queries. *arXiv preprint arXiv:2311.17389*, 2023.

[25] Jian Wu, Liwei Ma, and Xiaolin Hu. Delving deeper into convolutional neural networks for camera relocalization. In *2017 ICRA*, pages 5644–5651. IEEE, 2017.

[26] Samarth Brahmbhatt, Jinwei Gu, Kihwan Kim, James Hays, and Jan Kautz. Geometry-aware learning of maps for camera localization. In *IEEE Conference on CVPR*, 2018.

[27] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *Proceedings of the IEEE Conference on CVPR*, pages 5974–5983, 2017.

[28] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on CVPR*, pages 7482–7491, 2018.

[29] Alex Kendall and Roberto Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *2016 ICRA*, pages 4762–4769. IEEE, 2016.

[30] Iaroslav Melekhov, Juha Ylioinas, Juho Kannala, and Esa Rahtu. Image-based localization using hourglass networks. In *Proceedings of the IEEE ICCV workshops*, pages 879–886, 2017.

[31] Samarth Brahmbhatt, Jinwei Gu, Kihwan Kim, James Hays, and Jan Kautz. Geometry-aware learning of maps for camera localization. In *Proceedings of the IEEE conference on CVPR*, pages 2616–2625, 2018.

[32] Hunter Blanton, Connor Greenwell, Scott Workman, and Nathan Jacobs. Extending absolute pose regression to multiple scenes. In *Proceedings of the IEEE/CVF Conference on CVPR Workshops*, pages 38–39, 2020.

[33] Shuai Chen, Zirui Wang, and Victor Prisacariu. Direct-posenet: Absolute pose regression with photometric consistency. In *2021 Int. Conf. on 3DV)*, pages 1175–1185. IEEE, 2021.

[34] Yoli Shavit, Ron Ferens, and Yosi Keller. Learning multi-scene absolute pose regression with transformers. In *Proceedings of the IEEE/CVF ICCV*, pages 2733–2742, 2021.

[35] Yoli Shavit and Yosi Keller. Camera pose auto-encoders for improving pose regression. In *European Conference on Computer Vision*, pages 140–157. Springer, 2022.

[36] Shuai Chen, Xinghui Li, Zirui Wang, and Victor A Prisacariu. Dfnet: Enhance absolute pose regression with direct feature matching. In *European Conference on Computer Vision*, pages 1–17. Springer, 2022.

[37] Shuai Chen, Tommaso Cavallari, Victor Adrian Prisacariu, and Eric Brachmann. Map-relative pose regression for visual re-localization. In *Proceedings of the IEEE/CVF Conference on CVPR*, pages 20665–20674, 2024.

[38] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on CVPR*, pages 770–778, 2016.

[39] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF ICCV*, pages 1314–1324, 2019.

[40] Federico Boniardi, Tim Caselitz, Rainer Kümmerle, and Wolfram Burgard. Robust lidar-based localization in architectural floor plans. In *IEEE/RSJ Int. Conf. on IROS*, pages 3318–3324. IEEE, 2017.

[41] Federico Boniardi, Tim Caselitz, Rainer Kümmerle, and Wolfram Burgard. A pose graph-based localization system for long-term navigation in cad floor plans. *Robotics and Autonomous Systems*, 112:84–97, 2019.

[42] Zhikai Li, Marcelo H Ang, and Daniela Rus. Online localization with imprecise floor space maps using stochastic gradient descent. In *IEEE/RSJ Int. Conf. on IROS*, pages 8571–8578. IEEE, 2020.

[43] Oscar Mendez, Simon Hadfield, Nicolas Pugeault, and Richard Bowden. Sedar: reading floorplans like a human—using deep learning to enable human-inspired localisation. *IJCV*, 128(5):1286–1310, 2020.