# LiteVLoc: Map-Lite Visual Localization for Image Goal Navigation

Jianhao Jiao[1], Jinhao He[2], Changkun Liu[3], Sebastian Aegidius[1], Xiangcheng Hu[4], Tristan Braud[3], Dimitrios Kanoulas[1,5]

***Abstract*— This paper presents LiteVLoc, a hierarchical visual localization framework that uses a lightweight topo-metric map to represent the environment. The method consists of three sequential modules that estimate camera poses in a coarse-to-fine manner. Unlike mainstream approaches relying on detailed 3D representations, LiteVLoc reduces storage overhead by leveraging learning-based feature matching and geometric solvers for metric pose estimation. A novel dataset for the map-free relocalization task is also introduced. Extensive experiments including localization and navigation in both simulated and real-world scenarios have validate the system's performance and demonstrated its precision and efficiency for large-scale deployment. Code and data will be made publicly available at `https://rpl-cs-ucl.github.io/LiteVLoc`.**

## I. Introduction

### A. Motivation

Imagine that you start the daily route to campus from the nearest subway station. Before setting out, it is essential to determine your relative position and orientation to the destination. As you commence, by iteratively repeating this process, you should refine and update your location as well as adjust the planned route. This strategy will ensure the efficient and reliable navigation to the destination.

The above process of determining the position and orientation of a body relative to a reference frame using images is known as visual localization (VLoc). Most of VLoc methods [1]–[5] follow a standard procedure that initially constructs a metric map of the environment and then predict the camera's poses within this map. To create a metrically precise map, VLoc methods typically employ off-the-shelf Structure-from-Motion (SfM) software [6], which ensures accurate mapping. Alternatively, less refined maps can be constructed in real time using SLAM, though this approach often comes at the cost of reduced accuracy. In either case, the resulting maps consume vast amounts of storage [7] (e.g., 2*GB* for an indoor scene of hLoc [5]), and present challenges for maintenance, particularly in the context of long-term navigation. As Heraclitus famously stated, 'You cannot step into the same river twice'; the concept is further underscored in a recent survey [8], highlighting that environments are subject to continual change in reality. This introduces a paradox: the more detailed and precise the description of an environment, the less adaptable it becomes when the environment inevitably shifts.

The aim of this paper is to develop a VLoc method that delivers camera poses at a metric level using a lightweight and sparse map. Previous research has demonstrated that the topo-metric map is a favorable representation due to its simplicity, flexibility, and scalability [9]–[14]. This map formulates the environment as a graph with nodes that contain images or visual features and spatial coordinates, and edges that represent node relationships. Despite these benefits, the adoption of topo-metric maps in real-world robotic systems remains limited, likely due to their sparse nature, which poses challenges for VLoc.

### B. Contributions

In this paper, we propose **LiteVLoc**, a hierarchical and map-lite VLoc method for robot navigation by exploiting the topo-metric map. LiteVLoc, which is inspired from the hierarchical framework of VLoc [5], estimates the camera's poses in a coarse-to-fine manner with three successive modules: *(1)* the Global Localization (GL) module initializes the camera's poses at a topological level, *(2)* the Local Localization (LL) module refines the camera's pose at a metric level, and *(3)* the Pose SLAM (PS) module integrates the low-rate VLoc's results with high-rate sensor-specific odometry to deliver real-time poses in the global coordinate for subsequent navigation tasks.

The LL module attempts to solve the challenge of map-free relocalization [15], which involves estimating a query pose from a single reference image. This task becomes particularly difficult when significant differences in appearance, viewpoint, or relative distance occur between the query and the reference images. We introduce a new dataset specifically for this task, along with a detailed evaluation of state-of-the-art (SoTA) feature matching models. Our findings reveal that the highest-scoring method can establish reliable correspondences between images across diverse environments on a zero-shot basis, significantly enhancing our method's generalization capabilities.

[1]The authors are with the Robot Perception and Learning Lab, Intelligent Robotics, Department of Computer Science, University College London, Gower Street, WC1E 6BT, London, UK. {ucacjji, d.kanoulas}@ucl.ac.uk.

[2]The author is with the Thrust of Robotics and Autonomous Systems, The Hong Kong University of Science and Technology (Guangzhou), Nansha District, Guangzhou, China.

[3]The authors are with the Department of Computer Science and Engineering, HKUST, Hong Kong, China.

[4]The author is with the Department of Electronic an Computer Engineering, HKUST, Hong Kong, China.

[5]Dimitrios Kanoulas is also with the AI Centre, Department of Computer Science, University College London, Gower Street, WC1E 6BT, London, UK and Archimedes/Athena RC, Greece.

This work was supported by the UKRI Future Leaders Fellowship [MR/V025333/1] (RoboHike). For the purpose of Open Access, the author has applied a CC BY public copyright license to any Author Accepted Manuscript version arising from this submission.

In addition to evaluating LiteVLoc on VLoc tasks, we have deployed it on a real-world legged robot to perform image-goal visual navigation (VNav) tasks. This integration enables better human-robot interaction by providing a more intuitive navigation goal to the robot, unlike the traditional method of specifying goals with coordinates. In general, this paper presents the following *contributions*:

1) We propose a general, scalable, and hierarchical VLoc framework utilizing a topo-metric map for robot navigation. This framework's *generalization* facilitates adaptation to various robots with specific vision-based approaches, while its *scalability* reduces reliance on specialized mapping devices, highlighting its potential for large-scale deployment.
2) Within this framework, we present a specific VLoc on a real-world robot. The map can be constructed from various geo-tagged image sources, including robot-mounted cameras and mobile devices like AR glasses [16], despite existing domain gaps. The map size can be reduced to $17MB$ covering a $200m$ route, with an average translation error less than $0.25m$.
3) We introduce a new map-free relocalization dataset, collected by mobile robots in environments ranging from simulated offices to real-world campuses. As a supplement to the existing dataset [15], ours offers greater diversity in viewpoints and relative distances. This resource is vital for evaluating map-free relocalizers for mobile robot applications.
4) We further implement and verify the image-goal VNav system in both simulated and real-world environments based on the map and VLoc method.

## II. Related Work

Environmental representations for VLoc can be broadly classified into categories based on whether the approaches employ a pre-built 3D metric map (i.e., with or without 3D representation) to estimate the camera's poses.

### A. VLoc without 3D Representation

The simplest form of a map consists of a database of visual features or mapping images and their corresponding poses. When the image nodes are connected, this map can be structured as a topological or topo-metric map. Given a query image, the most similar images from the map are retrieved [17]–[21], and the query pose is estimated based on the poses of the top retrieved images [22]–[24]. These methods, commonly referred to as image retrieval or visual place recognition (VPR), discretize the environment into distinct places but cannot provide fine-grained estimates of the 6-DOF pose, often resulting in large error margins (e.g., 5 m and 90° as reported in [18]). An alternative approach, pose regression [22], [25]–[30], trains scene-specific neural networks to predict the absolute or relative pose directly from a query image using the database. While pose regression is computationally efficient and fast, it generally lacks the accuracy and generalization ability of structure-based methods [22], [31].

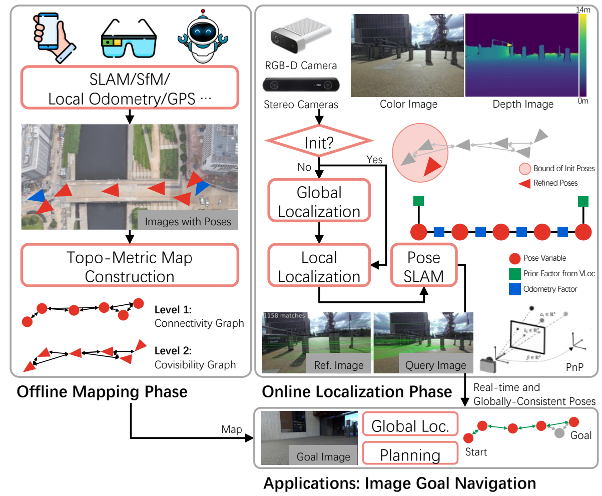



Fig. 1. Pipeline of LiteVLoc. The mapping phase builds a topo-metric map from a set of images with poses (Section III-B). Selected cameras are marked in red; otherwise, they are marked in blue. The map includes two levels, with edges marked in black: level-1 for planning and level-2 for VLoc. The localization phase estimates camera's poses in a coarse-to-fine manner (Section III-C). The map and real-time localization results can be applied to various applications such as image goal navigation (Section III-D). The image to illustrate PnP is derived from [45].

### B. VLoc with 3D Representation

Structure-based approaches traditionally rely on sparse 3D models that are generated from SfM to establish 2D-3D correspondences [5], [32]–[35]. After obtaining 2D-3D correspondences, the camera pose can be estimated via Perspective-n-Point (PnP) [36]. Alternatives to SfM models include meshes [37], Neural Radiance Fields (NeRFs) [38]–[40], and 3D Gaussian Splatting [41]. However, generating high-quality 3D representation in complex environments is often difficult and time-consuming [7]. Scene coordinate regression (SCR) methods directly predict 3D coordinates for given 2D pixel positions in the query image [7], [42]–[44], which can alleviate this issue. Despite this, SCR requires training a scene-specific model and are typically insufficiently robust or accurate for large-scale scenes [7].

LiteVLoc falls within the VLoc without 3D representation category, utilizing a topo-metric map to represent the environment. This eliminates the need to construct a 3D metrically-precise map for localization and streamlines the mapping process for large-scale navigation tasks. Different from VPR, LiteVLoc employs robust feature matching to establish correspondences between the query image and the top-retrieved image. By using depth information from stereo cameras, our approach can achieve localization results with accuracy comparable to that of using dense 3D representations. In contrast to pose regression and SCR methods, LiteVLoc is scene-agnostic and can be applied to unseen environments without requiring network fine-tuning.

## III. Methodology

This section presents the detailed methodology of the LiteVLoc framework and the VNav system derived from it.

### A. Preliminary

*1) Problem Definition:* Let $\mathbf{x} = [\mathbf{t}, \mathbf{q}]$ represent the camera's pose, where $\mathbf{t} = [x, y, z]^\top$ is the position, and $\mathbf{q} = [qw, qx, qy, qz]^\top$ is the quaternion representing rotation. Without loss of generality, we consider that the robot frame coincides with the camera frame, using "camera pose" to refer to the "robot pose." The camera's pose in the world frame at time step $k$ is denoted as $\mathbf{x}_k$. Given the pre-built and unchanged map $\mathcal{M}$, visual observations $\mathcal{Z}_k$, interoceptive sensor measurements $\mathcal{U}_k$ (e.g., IMU data), and the prior pose $\hat{\mathbf{x}}_{k-1}$ from the last time step, we formulate the localization as an optimization problem [46]

$$\arg\min_{\mathbf{x}_k} \; \underbrace{\left[f_u(\mathbf{x}_k, \mathcal{U}_k, \hat{\mathbf{x}}_{k-1}) + f_z(\mathbf{x}_k, \mathcal{M}, \mathcal{Z}_k)\right]}_{f(\mathbf{x}_k, \mathcal{M}, \mathcal{Z}_k, \mathcal{U}_k, \hat{\mathbf{x}}_{k-1})}. \quad (1)$$

where $f(\cdot)$ is decomposed into two objectives: $f_u(\cdot)$, which measures the deviation between the propagated pose and the variable, and $f_z(\cdot)$, which quantifies the deviation between the predicted measurements and practical observations. But solving problem (1) is challenging due to its non-linear and non-convex nature. To make it tractable, this paper approximates the solution by first addressing a vision-based localization and then tackling a pose SLAM problem.

*2) LiteVLoc Framework:* Figure 1 illustrates the pipeline of the LiteVLoc's framework. The process begins with the GL module, which initializes the camera's coarse position in the global coordinate system defined by the map. It is triggered when the mission starts or the robot loses its position (e.g., sufficient feature correspondences cannot be established over an extended period). Once initialized, the LL module refines the camera's pose at a metric level. It consist of three steps: searching for the most similar reference node, establishing correspondences between the reference image and the observation, and estimating the camera's pose. Finally, the PS module fuses low-rate VLoc's results with local odometry (e.g., proprioceptive odometry), which provides high-rate and accurate pose estimates.

### B. Map Construction

*1) Map Representation:* We design a two-level topo-metric map to support VLoc and VNav. Each level is represented as an undirected graph $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$, where $\mathcal{N}$ is the set of nodes and $\mathcal{E}$ is the set of edges depicting nodes' relationships. The first level, the *connectivity graph* (CnG), is used for motion planning. Nodes are connected if they are navigable from each other, and each node stores the pose. The second level, the *covisibility graph* (CvG), is used for VLoc. Nodes are connected if they visual overlaps with each other, and in addition to the pose, each node stores visual VPR features and raw images captured in this place.

*2) Keyframe Selection:* For simplicity, we assume the CnG is equal to the CvG. We use the CvG as an example to explain how the map is constructed. We assume that a set of visual observations and their associated poses which are denoted by $\mathcal{C} = \{(\mathcal{Z}_i, \mathbf{x}_i)\}$ are available. $\mathcal{C}$ can be obtained from various sources, such as single-/multi-agent SLAM or SFM algorithms [47], [48], cameras mounted on robots with odometry (e.g., wheel encoders, GPS) [49], or internet databases (e.g., Google Street View) [50]. Covisibility between frames is determined by checking for sufficient feature correspondences. To reduce the storage requirement, we select a subset of keyframes $\mathcal{C}^{\#}$, while maximumly preserving the environmental information. These keyframes with edges construct the CvG. We formulate the keyframe selection as an optimization problem:

$$\arg\max_{\mathcal{C}^{\#}} \; h(\mathcal{C}^{\#}), \quad \text{subject to } |\mathcal{C}^{\#}| \leq M, \; \mathcal{C}^{\#} \subseteq \mathcal{C}, \quad (2)$$

where $h(\cdot)$ quantifies the information and $M$ is a hyper-parameter. We apply the greedy-based method [51] to approximate $\mathcal{C}^{\#}$ within a reasonable computation time. If visual observations include depth images, we represent the environment using a 2D occupancy map constructed from selected keyframes, with $h(\cdot)$ defined as the number of occupied grid cells. If depth images are not available, we use a simple approach by treating frames as point clouds and downsampling them at a fixed resolution to select keyframes.

### C. Coarse-to-Fine Visual Localization

LiteVLoc is algorithm-agnostic, meaning that it can be applied to virtually any VPR and feature matching model. In this section, we present a specific solution.

*1) Global Localization:* A key assumption in VPR is that geographically close places share similar layouts, which can be partially observed by cameras [8]. This *similarity* can be reflected in the visual resemblance between two images and can be quantified by the relative distance between their *global descriptors* (describing overall features of an image). Following this assumption, we can initialize the camera's pose using that of the top-1 retrieved image if the current observation is similar to it. We use the pre-trained CosPlace [19] model for VPR due to its high performance and simple architecture. CosPlace consists of a convolutional encoder and a generalized mean pooling (GeM) layer to compute the global descriptor. We use CosPlace with this configuration in experiments: ResNet-18 [52] as the backbone; the dimension of the descriptor is 256.

*2) Local Localization:* Using the pose from the previous time step as a prior, we can retrieve nearby candidate nodes by first finding the closest map node. Then, using covisibility cues from the CvG, we gather the map and its neighbors to form a set of candidates. The most similar node from the set is selected as the reference node.

The following steps is to first establish local feature correspondences between the reference image (denoted by $\mathbf{I}_{ref}$) and the current observation, and then estimate the camera's pose. The observation, denoted by $\mathcal{Z}_k = \{\mathbf{I}_k, \mathbf{D}_k\}$, offers a color and depth image. Dense 2D-2D pixel correspondences between $\mathbf{I}_{ref}$ and $\mathbf{I}_k$ are obtained using the MASt3R model [53]. MASt3R, which was trained on extensive data, directly predicts 3D geometry from image pairs and has demonstrated strong robustness in our tests. We then obtain 2D-3D correspondences by lifting 2D pixels to 3D points using the depth image $\mathbf{D}_k$ and the camera intrinsics. Finally, the camera's pose is estimated by feeding these correspondences into a PnP solver [36] with a RANSAC loop [54]. The number of inliers can be used to check whether the LL is successful.

*3) Pose SLAM:* VLoc's performance may be affected by several factors such as *(1)* large domain gaps and viewpoint differences between the map and observations, *(2)* textureless or repetitive scenes, *(3)* noise in depth data, and *(4)* time delays in visual processing. While using inlier counts to filter out unreliable results can mitigate these issues, it reduces the update rate. To address this, we designed a PS module that integrates outputs from VLoc and local odometry, which enhances the system's overall reliability and performance.

Local odometry can come from possible sources that provide cumulative motion estimates, such as wheel encoders or visual odometry. In simulations, we implement an ICP algorithm for depth point cloud [55], while in real-world environments, we use proprioceptive odometry [56] that mixes IMU, encoder and robot contact data. We model the PS problem as a factor graph (Fig. 1) and solve it using the GTSAM library [57] from the optimization perspective. With camera's poses as variables, the cost function includes error terms from both prior and odometry constraints. The prior constraints, provided by VLoc, offer initial estimates of camera's poses relative to the map, while the odometry constraints, derived from local odometry, provide estimates of pose changes at each time step. Optimization is triggered with new VLoc results; otherwise, the pose is directly propagated by the local odometry.

### *D. Closed-Loop Visual Navigation*

We integrate LiteVLoc with motion planning to achieve the closed-loop image goal VNav (see Fig. 1). Unlike point-goal navigation, where the target is specified by coordinates, image-goal navigation directs the robot to a target image [49], enhancing better human-robot interaction. Although this paper shows the image-goal visual navigation, the proposed approach can be extended to tasks such as object-goal or instruction-based navigation. By leveraging the map and VLoc method, we can localize the goal using VPR.

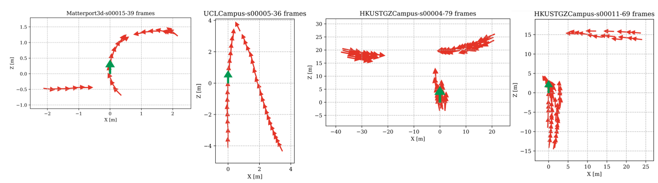
Planning is hierarchical, with global planning using Dijkstra algorithm to find the shortest path from the start to the goal node on the CnG. Traversed nodes are stored as subgoals. By projecting the nearest subgoal's coordinates onto the robot frame, local motion planning generates a collision-free path to this goal using single-frame depth measurements and the robot's kinematics, In simulations, we use iPlanner [58]. To avoid fine-tuning in real-world tests, we modify the SoTA nonlearning motion primitives planner (Falco) [59], which requires only a single-frame depth point cloud as input and eliminates the need to build a local map.

## IV. Experiment

### *A. Overview*

*1) Settings:* We conduct experiments from three perspectives, with increasing task complexity. All experiments are evaluated using both simulated and real-world data.


(a) Sample images of different scenarios: (top) Matterport3D, (middle) UCL Campus, and (bottom two rows) HKUSTGZ Campus.




(b) Poses of cameras (arrows indicate the camera's front view). Green and red arrows represent reference images and query images, respectively.

Fig. 2. Sample images and the distribution of camera poses in the proposed dataset for evaluating map-free relocalization methods.

- **Map-free Relocalization** (Section IV-B): We benchmark 13 SoTA feature matchers coupled with a PnP + RANSAC solver using a new dataset to evaluate pose estimation accuracy against severe viewpoint changes, following the setting in [15].
- **Visual Localization** (Section IV-C): We evaluate the complete LiteVLoc system with sequences captured from both simulated and real-world environments.
- **Image Goal Navigation** (Section IV-D): We test the VNav system on both simulated and real-world robots, providing both quantitative and qualitative results.

*2) Implementation Details:* LiteVLoc is implemented in Python. All algorithms, except for real-world VNav experiments, are run on an i9 desktop PC with an Nvidia GeForce RTX 4090 GPU. Real-world VNav experiments are conducted on ANYmal-D [69] quadruped robots, equipped with a front-view ZED2 stereo camera ($540\times960$ resolution, generating reliable depth up to $15m$), an Intel NUC computer (for planning), and a NVIDIA Jetson Orin (for visual processing). Ground-truth (GT) poses are generated using a Livox Mid360 LiDAR with the Fast-LIO2 algorithm [70]. All learning-based models were used without fine-tuning.

### *B. Map-free Relocalization*

*1) Dataset:* Related datasets, such as the map-free relocalization benchmark [15] and several SfM datasets, have limitations compared to ideal robot datasets. They typically focus on a specific object in circular motion, whereas robots often navigate in straight paths and operate in unstructured environments. To address this, we propose a new dataset for this task, including both simulated and real-world data: *(1)* **Simulated Matterport3D** [71]: An indoor dataset featuring similar scenes and occluded viewpoint, with 27

TABLE I

PERFORMANCE OF RELATIVE POSE ESTIMATION WITH DIFFERENT FEATURE MATCHERS. $[Xcm, Y^\circ]$ INDICATES THE PRECISION OF POSE ERROR WITHIN THIS RANGE. THE FIRST AND SECOND BEST RESULTS ARE HIGHLIGHTED IN RED AND BLUE.

| Types | Methods | Max $E_t[m]/E_r[deg]$ ↓ | Avg. Median $E_t[m]/E_r[deg]$ ↓ | $[5cm, 5^\circ]$ ↑ | $[25cm, 5^\circ]$ ↑ | $[1m, 10^\circ]$ ↑ | % Estimated ↑ | Time$[ms]$ ↓ |
|---|---|---|---|---|---|---|---|---|
| **Simulated Matterport3D + UCL Campus Dataset** | | | | | | | | |
| Dense | RoMa [60] | 6.52/169.75 | 0.26/0.73 | 62.44 | 75.58 | 86.85 | **100** | 321 |
| | DUSt3R [61] | 6.55/**43.85** | **0.16**/1.25 | 40.84 | **77.46** | **94.83** | 99.53 | 669 |
| | MASt3R [53] | 5.78/**14.85** | **0.06**/**0.64** | **66.19** | **80.75** | **94.83** | 97.18 | 101 |
| Semi-Dense | LoFTR [35] | **4.65**/179.31 | 0.42/4.38 | 53.05 | 69.95 | 81.22 | **100** | 20 |
| | Efficient-LoFTR [62] | 7.01/178.94 | 0.45/8.56 | 56.80 | 69.01 | 80.75 | **100** | 16 |
| | MatchFormer [63] | **4.73**/178.70 | 0.48/7.04 | 46.00 | 67.60 | 77.93 | **100** | 36 |
| | XFeat* [64] | 8.31/178.71 | 0.57/10.10 | 28.16 | 63.38 | 76.52 | **100** | **12** |
| Sparse | SuperPoint-LightGlue [32], [33] | 8.61/178.93 | 0.44/5.95 | 49.29 | 69.01 | 80.75 | 98.59 | 21 |
| | GIM-LightGlue [33], [65] | 9.17/177.81 | 0.29/1.41 | 36.15 | 63.38 | 78.40 | 92.95 | 28 |
| | GIM-DKM [65], [66] | 7.94/175.43 | 0.20/**0.69** | 65.72 | 79.81 | 90.61 | **100** | 760 |
| | XFeat [64] | 5.66/176.88 | 0.57/11.58 | 32.39 | 60.09 | 73.23 | **100** | 16 |
| | (Non-Learning) SIFT-NN [67] | 427.31/171.37 | 2.93/6.49 | 42.25 | 63.38 | 75.11 | 90.14 | 24 |
| | (Non-Learning) ORB-NN [68] | 207.71/179.52 | 0.49/8.11 | 30.04 | 55.86 | 70.89 | 90.14 | **11** |
| **HKUSTGZ Campus Dataset** | | | | | | | | |
| **Types** | **Methods** | **Max $E_t[m]/E_r[deg]$ ↓** | **Avg. Median $E_t[m]/E_r[deg]$ ↓** | **$[5cm, 5^\circ]$ ↑** | **$[25cm, 5^\circ]$ ↑** | **$[1m, 10^\circ]$ ↑** | **% Estimated ↑** | **Time$[ms]$ ↓** |
| Dense | RoMa [60] | 114.49/179.90 | 2.69/8.80 | 5.13 | 46.15 | **70.00** | **100** | 312 |
| | DUSt3R [61] | **58.38**/**175.23** | **1.27**/**1.44** | 3.85 | 37.95 | 67.44 | 98.72 | 361 |
| | MASt3R [53] | **52.65**/**174.91** | **0.39**/**1.05** | **5.39** | **47.44** | **74.36** | 96.92 | 99 |
| Semi-Dense | LoFTR [35] | 178.50/180 | 3.20/10.86 | 5.13 | 38.97 | 64.87 | **100** | 22 |
| | Efficient-LoFTR [62] | 86.32/179.58 | 2.96/11.08 | 4.36 | 41.28 | 65.13 | 99.49 | 18 |
| | MatchFormer [63] | 189.29/179.84 | 3.03/8.65 | 3.33 | 37.95 | 65.13 | 98.97 | 37 |
| | XFeat* [64] | 262.07/179.90 | 3.17/10.35 | 1.80 | 22.56 | 55.90 | 94.62 | **14** |
| Sparse | SuperPoint-LightGlue [32], [33] | 204.06/179.94 | 1.82/7.28 | 3.59 | 39.49 | 63.33 | 91.54 | 23 |
| | GIM-LightGlue [33], [65] | 148.17/179.64 | 2.24/7.04 | 2.56 | 35.64 | 62.05 | 89.49 | 33 |
| | GIM-DKM [65], [66] | 63.23/179.63 | 2.36/6.40 | **5.64** | **47.18** | **70.00** | **100** | 800 |
| | XFeat [64] | 84.16/179.84 | 4.36/17.30 | 3.08 | 27.69 | 56.41 | **100** | 19 |
| | (Non-Learning) SIFT-NN [67] | 594.87/178.03 | 2.96/6.12 | 2.05 | 22.05 | 45.39 | 64.62 | 23 |
| | (Non-Learning) ORB-NN [68] | 607.47/179.84 | 1.55/1.84 | 1.54 | 16.15 | 42.56 | 69.49 | **13** |

TABLE II

ABSOLUTE TRAJECTORY ERROR ON THE SIMULATED SEQUENCES.

| Env. | Map Size* | Seq. (Length) | ICP (15) | VLoc (1) | PS (15) | PS-Opt (1) |
|---|---|---|---|---|---|---|
| *Env*0 | $\|\mathcal{N}\| = 20$ | Seq0 (34.70$m$) | 1.400 | **0.013** | 0.045 | 0.036 |
| | $\|\mathcal{E}\| = 71$ | Seq1 (27.10$m$) | 0.836 | **0.016** | 0.049 | 0.090 |
| | 21*KB*+1.4*MB* | Seq2 (24.40$m$) | 0.747 | **0.010** | 0.080 | 0.051 |
| *Env*1 | $\|\mathcal{N}\| = 26$ | Seq3 (51.10$m$) | 2.031 | **0.029** | 0.065 | 0.057 |
| | $\|\mathcal{E}\| = 104$ | Seq4 (38.70$m$) | 0.173 | **0.042** | 0.043 | 0.039 |
| | 27*KB*+2.3*MB* | Seq5 (58.30$m$) | 4.589 | **0.025** | 0.104 | 0.075 |
| *Env*2 | $\|\mathcal{N}\| = 63$ | Seq6 (206.5$m$) | 22.921 | **0.042** | 0.143 | 0.134 |
| | $\|\mathcal{E}\| = 242$ | Seq7 (124.9$m$) | 5.716 | **0.055** | 0.757 | 0.295 |
| | 65*KB*+4.5*MB* | Seq8 (182.7$m$) | 6.953 | **0.073** | 2.480 | 0.926 |

* Number of nodes, edges, and storage of VPR descriptors and images.

TABLE III

ABSOLUTE TRAJECTORY ERROR ON THE REAL-WORLD SEQUENCES.

| Map Size | $\|\mathcal{N}\| = 81$, $\|\mathcal{E}\| = 383$, Storage: 83*KB*+17*MB* | | |
|---|---|---|---|
| **Methods** | **Seq0 (198m)** | **Seq1 (172m)** | **Seq2 (129m)** |
| RobotOdom (20) | 1.193 | 1.129 | 0.536 |
| RobotOdom-VLoc (0.5) | **0.238** | **0.217** | 0.191 |
| RobotOdom-PS (20) | 0.548 | 0.557 | 0.300 |
| RobotOdom-PS-Opt (0.5) | **0.238** | 0.242 | **0.170** |
| AirSLAM (10) | 43.483 | 20.881 | 17.704 |
| AirSLAM-VLoc (0.5) | **0.233** | **0.226** | **0.152** |
| AirSLAM-PS (10) | 8.791 | 5.231 | 1.040 |
| AirSLAM-PS-Opt (0.5) | 2.913 | 1.101 | 0.386 |

reference images and 703 query images. *(2)* **UCL Campus**: Captured by the ANYmal-D quadruped robot between two campus buildings. The environment is crowded and structureless, with 9 reference images and 290 query images. *(3)* **HKUSTGZ Campus**: Generaetd from sequences presented in [72], with depth provided by LiDAR map processing. This dataset features significant appearance, viewpoint, and position changes, with 30 reference images and 1897 query images. Several examples are shown in Fig. 2.

*2) Performance:* We report the results of relative pose estimation in Table I. These metrics reflect different aspects of a method's performance: maximum error and success rate indicate reliability, median error reflects accuracy, precision of pose error within $[Xcm, Y^\circ]$ shows the error bound and robustness, and time cost reflects efficiency. MASt3R [53] outperforms other matchers across all datasets while maintaining competitive computation time compared to its closest competitors, DUSt3R [53] and GIM-DKM [65], [66]. Therefore, we select MASt3R as the matcher for VLoc.

## C. Visual Localization

*1) Dataset:* We generate 3 simulated sequences for each of 3 Matterport3D environments using the autonomous navigation stack [73] (given a set of waypoints): *17DRP5sb8fy*, *EDJbREhghzL*, and *B6ByNegPMK* (denoted as *Simu-Seq*0-8 for *Env*0-2), with increasing environment size. Some sequences include motion into regions not covered by the map can be used to show the feasibility of our method in unseen areas. We also manually control the robot to collect 3 real-world outdoor sequences, where some regions are crowded with people and others on a bridge are structureless. *Simu-Seq*0, *Simu-Seq*3, *Simu-Seq*6, and *Real-Seq*0, containing geo-tagged images, are used to build the topo-metric map.

*2) Performance:* We report the absolute trajectory error (ATE) of modules in LiteVLoc in Tables II and III, along with a description of the map. The number following each

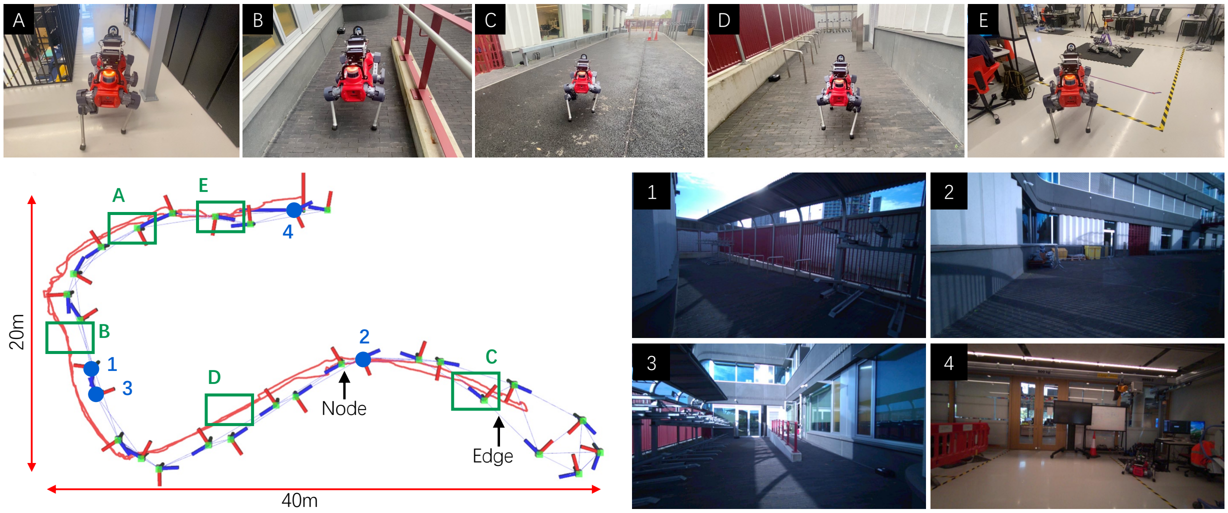



Fig. 3. Real-world experiment with a legged robot, with the red curve showning the robot's trajectory estimated by our VLoc method. The nodes (visualized as axes: red-X, green-Y, and blue-Z) and connected edges indicate the topo-metric map's structure. The robot is guided by a sequence of goal images that were captured by the AR glasses (right). It starts inside a room, navigates outside, follows a circular route, and returns. It totally traverses $94.5m$ in $314s$ with an average speed of $0.3m/s$. Green boxes A-E highlight key planning events along the route: (A) a narrow pathway, (B) descending slope, (C and D) outdoor pathways, and (E) the lab region. Refer to the video for a full demonstration of the navigation process.

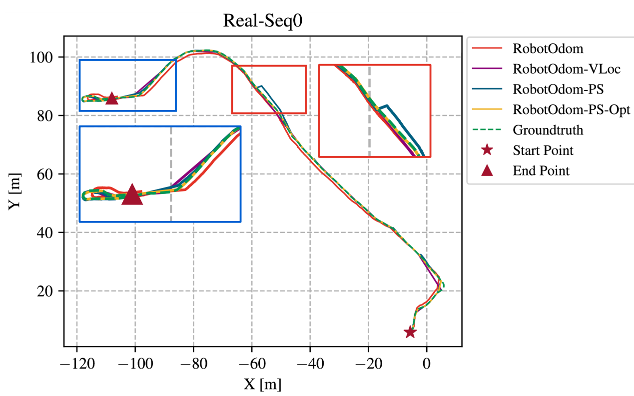



Fig. 4. Estimated trajectories on a real-world sequence traversing a bridge. VLoc's trajectory closely aligns with the ground truth, demonstrating high accuracy. The sudden changes in both *RobotOdom* and *RobotOdom-PS* are due to corrections from VLoc's results.

method indicates the frequency of its output. In Table III, *RobotOdom* refers to the proprioceptive odometry provided by the robot, and *X-PS-Opt* denotes the batch optimization results, where historical poses are optimized using additional measurements. We also replace *RobotOdom* with AirSLAM stereo camera odometry [74] for similar experiments. In Fig. 4, we visualize the trajectories of our method on *Real-Seq0*. These results confirm our hypothesis: *(1)* low-rate VLoc can effectively correct the accumulative drift from local odometry, *(2)* the localization accuracy is sufficient for navigation, and *(3)* batch optimization results provide good initial guess for accurate mapping missions.

### D. Image Goal Visual Navigation

We evaluate the VNav system using three simulated tests and one real-world test. In each experiment, the robot is sequentially given five goal images and tasked with completing the navigation process. Table IV presents the results of the simulated experiments, comparing the VNav system to the SoTA LiDAR-based navigation system [73] (GT localization + Falco) as a baseline. Due to the limited field of view of the camera, VNav sometimes encounters local minima, requiring the robot to rotate more frequently. This increases the time taken to reach the goal. However, the length of the navigation path does not increase significantly, indicating that VLoc consistently tracks the robot's position accurately.

TABLE IV

NAVIGATION PERFORMANCE IN SIMULATED ENVIRONMENTS.

| | Navigation Time $[s]$ and Path Length $[m]$ | | |
|---|---|---|---|
| **Methods** | *Env*0 | *Env*1 | *Env*2 |
| GTLoc + Falco [59] | $40.3s/25.5m$ | $42.7s/28.3m$ | $266.9s/220.7m$ |
| LiteVLoc + iPlanner [58] | $50.0s/28.6m$ | $52.9s/29.1m$ | $280.3s/223.7m$ |

Fig. 3 shows the results of the VNav system in real-world environments. We use the AR glasses [16] that provide geo-tagged images for the map construction. The real-world test further demonstrates the flexibility of our method, even in the presence of domain gaps and viewpoint differences between the reference images and the robot's observations (see Fig. 2 and 3). The robot successfully navigates from an lab to an outdoor pathway, guided by a sequence of goal images. It is also capable of autonomously navigating through narrow doorways and avoiding obstacles. The $94.5m$-route was completed in $314s$, with an average speed of $0.3m/s$. The indoor and outdoor environments present different lighting conditions, resulting in varying noise levels in the RGB and depth measurements. A video of the experiment provides a clearer visualization of this process.

## V. Conclusion

This paper proposes a simple yet effective VLoc method using topo-metric maps for robot navigation. The map-lite feature of ours is different from mainstream methods that using detailed 3D maps to represent environments. Experiments demonstrate that while the proposed LiteVLoc has much room for improvement, it can be used for many non-safety-critical applications. In addition, LiteVLoc reduces the strict mapping requirements for navigation. As spatially aware mobile devices become more prevalent, the gap between scalable navigation and real-world deployment will be narrowed. Future work will focus on optimizing real-time performance and further integrating with downstream navigation to handle more complex tasks in dynamic environments.

## References


[1] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han, "Large-scale image retrieval with attentive deep local features," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3456–3465.

[2] 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 *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 2019, pp. 8092–8101.

[3] T. Sattler, B. Leibe, and L. Kobbelt, "Efficient and effective prioritized matching for large-scale image-based localization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 9, pp. 1744–1756, 2016.

[4] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7199–7209.

[5] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *CVPR*, 2019.

[6] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[7] E. Brachmann, T. Cavallari, and V. A. Prisacariu, "Accelerated coordinate encoding: Learning to relocalize in minutes using rgb and poses," *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5044–5053, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:258841110

[8] P. Yin, J. Jiao, S. Zhao, L. Xu, G. Huang, H. Choset, S. Scherer, and J. Han, "General place recognition survey: Towards real-world autonomy," *arXiv preprint arXiv:2405.04812*, 2024.

[9] F. Dayoub, T. Morris, B. Upcroft, and P. Corke, "Vision-only autonomous navigation using topometric maps," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 1923–1929.

[10] T. Roussel, P. Chakravarty, G. Pandey, T. Tuytelaars, and L. Van Eycken, "Deep-geometric 6 dof localization from a single image in topometric maps," *arXiv preprint arXiv:2002.01210*, 2020.

[11] A. Rosinol, A. Violette, M. Abate, N. Hughes, Y. Chang, J. Shi, A. Gupta, and L. Carlone, "Kimera: From slam to spatial perception with 3d dynamic scene graphs," *The International Journal of Robotics Research*, vol. 40, no. 12-14, pp. 1510–1546, 2021.

[12] Y. He, I. Fang, Y. Li, R. B. Shah, and C. Feng, "Metric-free exploration for topological mapping by task and motion imitation in feature space," in *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023.

[13] N. Kim, O. Kwon, H. Yoo, Y. Choi, J. Park, and S. Oh, "Topological semantic graph memory for image-goal navigation," in *Conference on Robot Learning*. PMLR, 2023, pp. 393–402.

[14] N. Hughes, Y. Chang, S. Hu, R. Talak, R. Abdulhai, J. Strader, and L. Carlone, "Foundations of spatial perception for robotics: Hierarchical representations and real-time systems," *The International Journal of Robotics Research*, p. 02783649241229725, 2024.

[15] E. Arnold, J. Wynn, S. Vicente, G. Garcia-Hernando, A. Monszpart, V. Prisacariu, D. Turmukhambetov, and E. Brachmann, "Map-free visual relocalization: Metric pose relative to a single image," in *European Conference on Computer Vision*. Springer, 2022, pp. 690–708.

[16] J. Engel, K. Somasundaram, M. Goesele, A. Sun, A. Gamino, A. Turner, A. Talattof, A. Yuan, B. Souti, B. Meredith *et al.*, "Project aria: A new tool for egocentric multi-modal ai research," *arXiv preprint arXiv:2308.13561*, 2023.

[17] R. Arandjelović, P. Gronát, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5297–5307, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:44604205

[18] M. Xu, N. Sünderhauf, and M. Milford, "Probabilistic visual place recognition for hierarchical localization," *IEEE Robotics Autom. Lett.*, vol. 6, pp. 311–318, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:229643627

[19] G. Berton, C. Masone, and B. Caputo, "Rethinking visual geo-localization for large-scale applications," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4878–4888.

[20] G. Berton, G. Trivigno, B. Caputo, and C. Masone, "Eigenplaces: Training viewpoint robust models for visual place recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 11 080–11 090.

[21] A. Gordo, J. Almazan, J. Revaud, and D. Larlus, "End-to-end learning of deep visual representations for image retrieval," *International Journal of Computer Vision*, vol. 124, no. 2, pp. 237–254, 2017.

[22] T. Sattler, Q. Zhou, M. Pollefeys, and L. Leal-Taixé, "Understanding the limitations of cnn-based absolute camera pose regression," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3297–3307, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:81979654

[23] S. Dong, S. Liu, H. Guo, B. Chen, and M. Pollefeys, "Lazy visual localization via motion averaging," *arXiv preprint arXiv:2307.09981*, 2023.

[24] M. Zaffar, L. Nan, and J. F. Kooij, "On the estimation of image-matching uncertainty in visual place recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 17 743–17 753.

[25] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2938–2946.

[26] 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.

[27] 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.

[28] 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.

[29] V. Balntas, S. Li, and V. Prisacariu, "Relocnet: Continuous metric learning relocalisation using neural nets," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 751–767.

[30] Z. Laskar, I. Melekhov, S. Kalia, and J. Kannala, "Camera relocalization by computing pairwise relative poses using convolutional neural network," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 929–938.

[31] C. Liu, S. Chen, Y. Zhao, H. Huang, V. Prisacariu, and T. Braud, "Hr-apr: Apr-agnostic framework with uncertainty estimation and hierarchical refinement for camera relocalisation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 8544–8550.

[32] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 224–236.

[33] P. Lindenberger, P.-E. Sarlin, and M. Pollefeys, "LightGlue: Local feature matching at light speed," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 17 627–17 638.

[34] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 4938–4947.

[35] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, "LoFTR: Detector-free local feature matching with transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 8922–8931.

[36] X. Gao, X. Hou, J. Tang, and H.-F. Cheng, "Complete solution classification for the perspective-three-point problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, pp. 930–943, 2003. [Online]. Available: https://api.semanticscholar.org/CorpusID:15869446

[37] V. Panek, Z. Kukelova, and T. Sattler, "Meshloc: Mesh-based visual localization," in *European Conference on Computer Vision*, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:251018193

[38] A. Moreau, N. Piasco, M. Bennehar, D. V. Tsishkou, B. Stanciulescu, and A. de La Fortelle, "Crossfire: Camera relocalization on self-supervised features from an implicit representation," *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 252–262, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:257427144

[39] J. Liu, Q. Nie, Y. Liu, and C. Wang, "Nerf-loc: Visual localization with conditional neural radiance field," *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9385–9392, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:258180042

[40] Q. Zhou, M. Maximov, O. Litany, and L. Leal-Taixé, "The nerfect match: Exploring nerf features for visual localization," *arXiv preprint arXiv:2403.09577*, 2024.

[41] C. Liu, S. Chen, Y. Bhalgat, S. Hu, Z. Wang, M. Cheng, V. A. Prisacariu, and T. Braud, "Gsloc: Efficient camera pose refinement via 3d gaussian splatting," 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:271916288

[42] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother, "Dsac — differentiable ransac for camera localization," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2492–2500, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:4001530

[43] E. Brachmann and C. Rother, "Visual camera re-localization from RGB and RGB-D images using DSAC," *TPAMI*, 2021.

[44] ——, "Learning less is more - 6D camera localization via 3D surface regression," in *CVPR*, 2018.

[45] R. Sheffer and A. Wiesel, "Pnp-net: A hybrid perspective-n-point network," *arXiv preprint arXiv:2003.04626*, 2020.

[46] T. D. Barfoot, *State estimation for robotics*. Cambridge University Press, 2024.

[47] H. Xu, P. Liu, X. Chen, and S. Shen, "D2SLAM: Decentralized and distributed collaborative visual-inertial slam system for aerial swarm," *IEEE Transactions on Robotics*, 2024.

[48] C. Liu, Y. Zhao, and T. Braud, "Marvin: Mobile ar dataset with visual-inertial data," in *2024 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. IEEE, 2024, pp. 532–538.

[49] D. Shah, A. Sridhar, A. Bhorkar, N. Hirose, and S. Levine, "GNM: A general navigation model to drive any robot," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 7226–7233.

[50] D. Anguelov, C. Dulong, D. Filip, C. Frueh, S. Lafon, R. Lyon, A. Ogale, L. Vincent, and J. Weaver, "Google street view: Capturing the world at street level," *Computer*, vol. 43, no. 6, pp. 32–38, 2010.

[51] J. Jiao, Y. Zhu, H. Ye, H. Huang, P. Yun, L. Jiang, L. Wang, and M. Liu, "Greedy-based feature selection for efficient lidar slam," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 5222–5228.

[52] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[53] V. Leroy, Y. Cabon, and J. Revaud, "Grounding image matching in 3d with MASt3R," *arXiv preprint arXiv:2406.09756*, 2024.

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

[55] S. Rusinkiewicz and M. Levoy, "Efficient variants of the icp algorithm," in *Proceedings third international conference on 3-D digital imaging and modeling*. IEEE, 2001, pp. 145–152.

[56] M. Bloesch, "State estimation for legged robots-kinematics, inertial sensing, and computer vision," Ph.D. dissertation, ETH Zurich, 2017.

[57] F. Dellaert and G. Contributors, "borglab/gtsam," May 2022. [Online]. Available: https://github.com/borglab/gtsam)

[58] F. Yang, C. Wang, C. Cadena, and M. Hutter, "iplanner: Imperative path planning," *arXiv preprint arXiv:2302.11434*, 2023.

[59] J. Zhang, C. Hu, R. G. Chadha, and S. Singh, "Falco: Fast likelihood-based collision avoidance with extension to human-guided navigation," *Journal of Field Robotics*, vol. 37, no. 8, pp. 1300–1313, 2020.

[60] J. Edstedt, Q. Sun, G. Bökman, M. Wadenbäck, and M. Felsberg, "RoMa: Robust dense feature matching," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 19 790–19 800.

[61] S. Wang, V. Leroy, Y. Cabon, B. Chidlovskii, and J. Revaud, "Dust3r: Geometric 3d vision made easy," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 697–20 709.

[62] Y. Wang, X. He, S. Peng, D. Tan, and X. Zhou, "Efficient LoFTR: Semi-dense local feature matching with sparse-like speed," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 21 666–21 675.

[63] Q. Wang, J. Zhang, K. Yang, K. Peng, and R. Stiefelhagen, "MatchFormer: Interleaving attention in transformers for feature matching," in *Proceedings of the Asian Conference on Computer Vision*, 2022, pp. 2746–2762.

[64] G. Potje, F. Cadar, A. Araujo, R. Martins, and E. R. Nascimento, "XFeat: Accelerated features for lightweight image matching," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 2682–2691.

[65] X. Shen, Z. Cai, W. Yin, M. Müller, Z. Li, K. Wang, X. Chen, and C. Wang, "Gim: Learning generalizable image matcher from internet videos," *arXiv preprint arXiv:2402.11095*, 2024.

[66] J. Edstedt, I. Athanasiadis, M. Wadenbäck, and M. Felsberg, "Dkm: Dense kernelized feature matching for geometry estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 765–17 775.

[67] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, pp. 91–110, 2004.

[68] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*. Ieee, 2011, pp. 2564–2571.

[69] M. Hutter, C. Gehring, D. Jud, A. Lauber, D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, R. Diethelm, S. Bachmann, A. Melzer, and M. A. Höpflinger, "Anymal - a highly mobile and dynamic quadrupedal robot," *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 38–44, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:19007565

[70] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "FAST-LIO2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.

[71] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3d: Learning from rgb-d data in indoor environments," *arXiv preprint arXiv:1709.06158*, 2017.

[72] J. He, H. Huang, S. Zhang, J. Jiao, C. Liu, and M. Liu, "Accurate prior-centric monocular positioning with offline lidar fusion," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 11 934–11 940.

[73] C. Cao, H. Zhu, F. Yang, Y. Xia, H. Choset, J. Oh, and J. Zhang, "Autonomous exploration development environment and the planning algorithms," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 8921–8928.

[74] K. Xu, Y. Hao, S. Yuan, C. Wang, and L. Xie, "Airslam: An efficient and illumination-robust point-line visual slam system," *arXiv preprint arXiv:2408.03520*, 2024.