Learning Where to Look: Self-supervised Viewpoint Selection for Active Localization using Geometrical Information

Luca Di Giammarino[1], Boyang Sun[2], Giorgio Grisetti[1], Marc Pollefeys[2,3], Hermann Blum[2,4], and Daniel Barath[2]

[1] Sapienza University of Rome [2] ETH Zürich [3] Microsoft [4] Uni Bonn

Abstract. Accurate localization in diverse environments is a fundamental challenge in computer vision and robotics. The task involves determining a sensor's precise position and orientation, typically a camera, within a given space. Traditional localization methods often rely on passive sensing, which may struggle in scenarios with limited features or dynamic environments. In response, this paper explores the domain of active localization, emphasizing the importance of viewpoint selection to enhance localization accuracy. Our contributions involve using a data-driven approach with a simple architecture designed for real-time operation, a self-supervised data training method, and the capability to consistently integrate our map into a planning framework tailored for real-world robotics applications. Our results demonstrate that our method performs better than the existing one, targeting similar problems and generalizing on synthetic and real data. We also release an open-source implementation to benefit the community at www.github.com/rvp-group/learning-where-to-look.

Keywords: Visual Localization, Active Sensing, Deep Learning

1 Introduction

Localization and mapping are fundamental building blocks of autonomous systems. Localization aims at determining the camera position within an environment, while mapping involves estimating a comprehensive representation of the surrounding space. These capabilities enable agents to navigate and operate effectively in unexplored and dynamic settings, with applications ranging from autonomous vehicles [4], precision agriculture [35], and augmented reality [17].

Decoupling localization and mapping is often advantageous due to the distinct nature of these processes, allowing for separate optimization. Offline mapping computation enhances accuracy by overcoming real-time processing constraints and generating a map before the actual operation. However, even with offline computation, discrepancies in accuracy may exist across the map, impacting certain areas more than others. For example, these differences can stem from

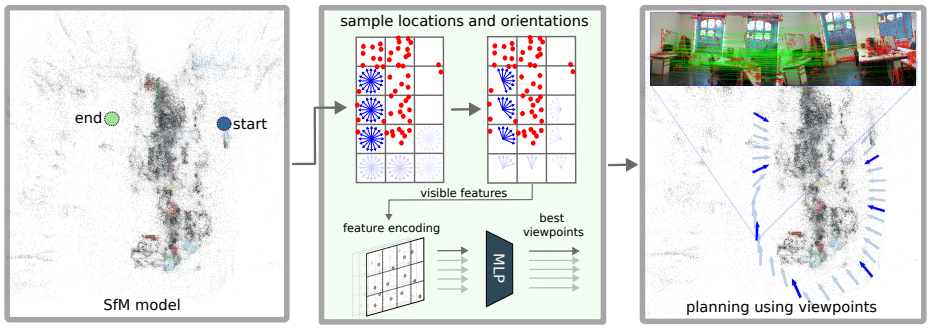


Fig. 1: Pipeline. Given a Structure-from-Motion (SfM) model, we aim to learn the camera viewpoint that can be employed to maximize the accuracy in visual localization. Our methodology requires first sampling the camera locations and orientation, calculating the best visibility orientation for each location, and learning active viewpoint through a Multi-layer Perceptron (MLP) encoder. The illustration above shows our full pipeline predicting active viewpoints for visual localization embedded into a planning framework.

non-uniformly distributed observations within the space. In such map regions characterized by heterogeneous accuracy, traditional localization methods may encounter challenges in delivering precise position information [55]. The difficulty lies in discerning meaningful features within the environment to facilitate accurate localization, giving rise to the concept of active localization [5].

Active localization entails purposeful selection and observation of specific environmental features to augment localization accuracy. In contrast to passive sensing, where an agent merely observes its surroundings, active localization allows the agent to actively seek out and concentrate on stable features. This approach significantly enhances the ability to determine an accurate position, particularly in complex or adverse environments.

This active approach to localization ensures adaptability to diverse surroundings and finds applications in collaborative scenarios. For instance, in human-robot collaboration, active localization may involve seeking input from human operators to identify critical features or areas of interest [17]. As technology progresses, active localization plays a pivotal role in enabling consistent robotic operation across a broad spectrum of environments, addressing challenges, and advancing the field of perception and robotics.

The concept of active localization encompasses various interpretations, ranging from enhancing map representation [17, 55] to optimizing localization accuracy through robot planning [10, 34]. In the context of map representation, studies have employed hand-crafted techniques, leveraging tools like the Fisher information matrix and optimality theory [55]. Conversely, approaches grounded in data-driven methodologies, which showed accurate results, necessitate collecting the training data manually [17]. This study addresses both aspects, proposing a map representation that identifies accurate active viewpoints. These viewpoints serve the dual purpose of enhancing localization accuracy and being readily ac-

cessible for efficient planning, thus encompassing the limitations observed in existing proposed works.

Inspired by [55] and [17], our primary focus is to extract valuable information from the environment's geometry, aiming to enhance active visual localization, enabling the integration of our map representation into a motion planning framework for robotics applications. The contributions of our work include a data-driven approach employing a compact architecture designed for real-time operation, a self-supervised data training method, a map representation facilitating multiple active viewpoints at specific locations in space, and the capability to compactly embed our map into a planning framework tailored for real-world robotics applications. A visual summary of our pipeline is depicted in Fig. 1.

2 Related Works

Visual localization describes the task of estimating the camera position and orientation for a query RGB/RGB-D image in a known scene (with databases). This task has gathered significant focus, particularly on improving localization accuracy from a certain perspective. Research within this field generally falls into one of two primary categories: the direct (or one-step) approach and the two-step approach. The first aims to directly estimate the camera pose from the query frame [8,11,28,43,45,46,50–52]. This is also known as pose regression. Learning-based models are increasingly being incorporated into this methodology, enhancing robustness and precision by integrating with conventional processes. The two-step approach initially identifies correspondences between the query frame and the database, followed by the estimation of the camera pose through optimization. These correspondences can be visual features [13,30,36–40,42], or dense correspondence between every pixel of the image [2,3,6,12,24,48,53].

All the above approaches are *passive localization*, implying no active decision-making regarding the camera's viewpoint. Rather, the focus is on utilizing the captured image for accurate and efficient localization. Some studies choose a different perspective to improve the performance of visual perception, such as in the case of visual localization. The agent can adjust its sensors autonomously, aiming to enhance perception from alternative viewpoints. This approach is known as active perception, which has been an open research area for over twenty years. Particularly within the domain of mobile autonomous agents, active perception strategies are frequently combined with planning or navigation modules to improve outcomes in tasks like visual localization. This integration enables the agent not only to perceive its environment more effectively but also to make informed decisions about its movements to optimize perception results, e.g. [10,34,54], surface converging [15,22,25,31].

In the specific field of active localization, research has been relatively limited. A significant portion of the work in this area focuses on active viewpoint selection through various geometric evaluation metrics. These metrics assess the effectiveness of a selected viewpoint in terms of its impact on visual localization performance. A noteworthy example of such research is [55]. It proposes an ef-

ficient way of calculating the Fisher information in a 3D environment and uses this quantity to find camera poses that maximize the visibility and vicinity of feature landmarks. More recent works have tried to utilize additional information from visual appearances, such as semantics from the image [1]. Most of the works in this category design a hand-crafted metric that links the geometry and appearance with the performance, such as visual localization accuracy. In contrast, we propose to learn from the distribution of the 3D landmarks and their contribution to the visual localization task.

Learning-based approaches for active visual localization have recently started to be investigated but mostly rely on end-to-end approaches. PoseNet [21], and one of its extensions [9] use a convolutional network to implicitly represent the scene, mapping a single monocular image to a 3D pose (position and orientation). [7] proposed a perceptual model to estimate the belief of the current robot state and a policy model over the current belief to localize accurately. The authors in [14] propose an uncertainty-driven policy model to plan a camera path for localization. Differently, [26] focuses on safety guarantee and proposes a policy model that recommends a collision-free viewpoint while maximizing the information gained from the observation. Works using a reinforcement learning framework are usually tightly coupled with the action execution of the agent, which makes it hard to train and generalize. A recent interesting study to learn active viewpoints shows good results when the problem is cast to classification [17]. However, this work requires user-selected labels, which might be hard and time-consuming to obtain and does not consider an overall representation of the map or involve motion planning.

In this work, we focus on extracting useful information from the environment's geometry to enhance active visual localization with the possibility of embedding our map representation into a motion planning framework for robotics applications. The contributions of our work are as follows:

- a data-driven approach relying only on a compact architecture designed for real-time operation in known environments;
- a way to learn in a self-supervised manner;
- a map representation that allows for a set of locations in the space to have one or more active viewpoints available;
- an open-source implementation available at `www.github.com/rvp-group/learning-where-to-look`.

The proposed method can easily be embedded into a planning framework for real robotics applications.

3 Learning Where To Look

We present our viewpoint selection approach in this section. The core of our approach is a learning-based viewpoint evaluation model. The model predicts a visual localization score for an input viewpoint, indicating its efficacy for an accurate localization result. The whole approach follows a "sampling-and-evaluation"

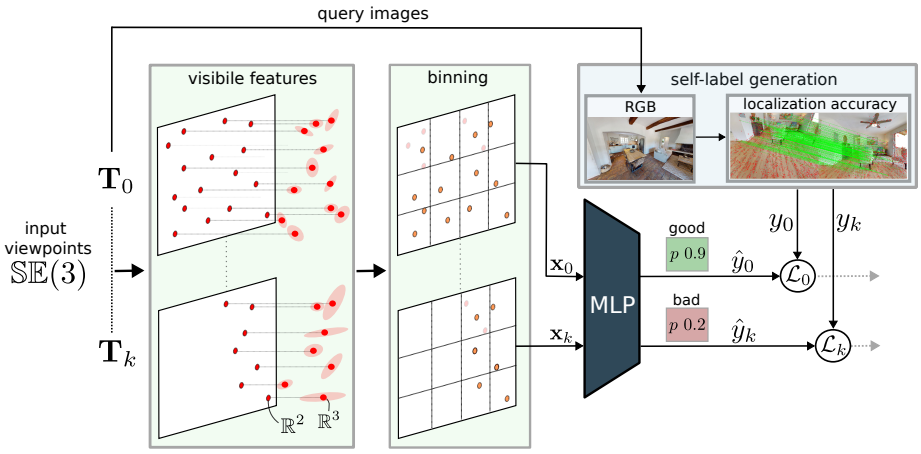


Fig. 2: Learning active viewpoints. Given a set of camera poses parameterized as homogeneous transformation matrices, obtained as explained in Sec. 3.1 and visibility information (3D landmarks and their projections), our goal is to develop a scoring function that discerns the suitability of a camera viewpoint for visual localization. We first identify visible data from each camera view to achieve this, as elaborated in Sec. 3.2. Subsequently, we encode this visible data through image binning for a fixed input size. The encoded information is then fed into a MLP encoder, which predicts the quality of the viewpoint for localization. This learning process, detailed in Sec. 3.3, is supervised by consistently providing the camera position, querying an RGB image through a simulator, and directly registering this image against a SfM model.

pipeline. We design a compact workflow to sample candidates' viewpoints from a given 3D environment and construct the input features of the certain viewpoint to pass into the model. A similar data acquisition structure is also applied during training to achieve self-supervision. This section starts with formulating the viewpoint selection task, followed by introducing our major components for building our data collection and learning pipeline, including initial viewpoint sampling, visibility check, and model training.

Our goal is to find a set of camera viewpoints within a sparse set of landmarks or point cloud $\mathcal{P} = \{\mathbf{l}_0, \ldots, \mathbf{l}_L\}$ with each landmark $\mathbf{l} \in \mathbb{R}^3$ that when employed during localization allow to maximize its accuracy. We discretize the orientations $\mathcal{R} = \{\mathbf{R}_0, \ldots, \mathbf{R}_N\} \in \mathbb{SO}(3)^N$ and the locations $\mathcal{V} = \{\mathbf{t}_0, \ldots, \mathbf{t}_M\} \in \mathbb{R}^{3\times M}$, using spherical sampling and a voxel grid, respectively. Using the following discretization, we create the set of camera viewpoints parameterized as homogeneous transformation matrices where an element is represented by $\mathbf{T}_{ij}$ and is parameterized as follows:

$$\mathbf{T}_{ij} = \begin{bmatrix} \mathbf{R}_i & \mathbf{t}_j \\ \mathbf{0} & 1 \end{bmatrix}, \qquad \begin{matrix} i = 0, \ldots, N \\ j = 0, \ldots, M \end{matrix} \tag{1}$$

Given these two input sets, for each position $\mathbf{t}_j$ and orientation $\mathbf{R}_i$, we aim to learn a scoring function that evaluates their suitability for visual localization. Let

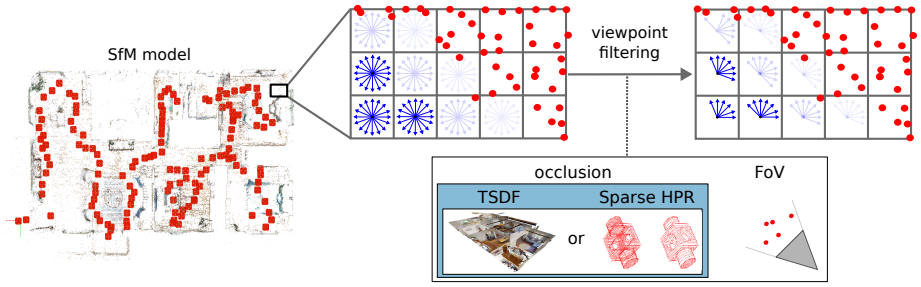


Fig. 3: Camera viewpoint generation. We represent our map as a discrete voxel grid $\mathcal{V}$ and a discrete set of orientations $\mathcal{R}$ constructed within the boundaries of a 3D reconstruction, e.g., coming from a SfM method. We filter the best directions from each camera location in the voxel grid based on visibility $\mathcal{Q}$, gradually removing occlusions. The illustration is done in 2D for ease of visualization.

this scoring function be $\mathbf{f}_{\mathcal{P}}(\mathbf{R}_i, \mathbf{t}_j) \rightarrow [0, 1]$. Note that this map representation is fundamental to self-generate data to implicitly learn $\mathbf{f}_{\mathcal{P}}(\mathbf{R}_i, \mathbf{t}_j)$ and can be employed for inference at constant time (i.e. for path planning). A summary of our learning strategy is illustrated in Fig. 2.

In the next two sections, we present how we sample the 3D locations and orientations composing our map (Sec. 3.1). After, we discuss how the best set of orientations $\mathcal{Q}$ for each camera location is selected based on landmarks visibility (Sec. 3.2). Finally, in Sec. 3.3 we present how we learn the function $\mathbf{f}_{\mathcal{P}}$ to select what viewpoints in the map are more suitable for localization, through a lightweight MLP.

3.1 Sampling

In this section, we detail the process of sampling locations $\mathcal{V}$ and orientations $\mathcal{R}$. This enables us to establish a discrete representation of the map, which is essential for localization and applies during both the training and inference phases. During training, this representation becomes necessary for facilitating the self-generation of data. During inference, this representation can be advantageous for planning; however, it is not strictly necessary, given that our learning strategy can effectively utilize any specified position $\mathbf{t}$ and orientation $\mathbf{R}$ of the camera. This procedure is detailed below and illustrated in Fig. 3.

Location. Given as input map a sparse set of landmarks or point cloud $\mathcal{P}$ and, the 3D boundary of the map, we construct a voxel grid $\mathcal{V}$ to sample camera positions $\mathbf{t}$ in the center of each voxel or camera bucket. The resolution of the voxel grid can be arbitrarily determined. In our experiments, we split the 3D space in a $8 \times 8 \times 8$ grid, representing each voxel as a cuboid. A mapping between the camera position and their location in memory is stored in a hash table, allowing $O(1)$ insertions and look-ups.

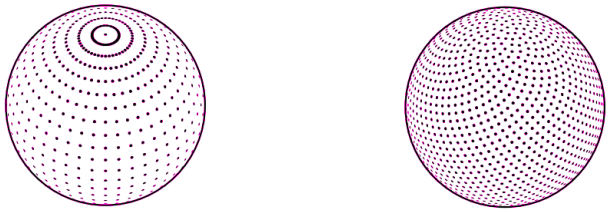


Fig. 4: Spherical sampling methods. The left plot shows classical azimuth-elevation sampling. The right one is Fibonacci sampling. We employed the technique on the right, given the more uniform distributed pattern.

Orientation. This step is required to generate the set of discrete orientations $\mathcal{R}$. We only consider azimuth θ and elevation ϕ of S^2 from $\mathbb{SO}(3)$ to determine each rotation, omitting the rotation around the camera optical-axes. For sampling, we employ the *spherical Fibonacci sampling* algorithm [16]. This sampling is organized in a closely wound generative spiral, where each point is positioned within the largest gap between the preceding points. Different from the classic azimuth–elevation sampling where equal angles of azimuth and elevation separate each point, this sampling exhibit a uniformly distributed pattern in a highly isotropic manner. The difference with a classical azimuth-elevation sampling can be appreciated in Fig. 4. Given the *lattice* nature, angles are incrementally generated in the following way:

$$\theta_i = \arccos\left(\frac{1-2i}{N}\right), \qquad \phi_i = i\pi\left(1+\sqrt{5}\right), \qquad i = 0,\ldots,N. \tag{2}$$

We calculate $\mathbf{R} = \mathbf{R}_\theta \mathbf{R}_\phi \in \mathbb{SO}(3)$ making a rotation along the x-axis, followed by a rotation along y-axis.

3.2 Visibility Check

After sampling the set of orientations $\mathcal{R}$ for each camera position $\mathbf{t}_j$, it becomes crucial to identify the subset $\mathcal{Q}_j$ that enables the visibility of 3D landmarks. It is important to discard viewpoints where none or only a minimal number of landmarks are visible. To accomplish this, we perform an initial visibility check to obtain $\mathcal{Q}_j \subseteq \mathcal{R}$ that maximizes visibility. Note that $\mathcal{Q}_j$ is the set calculated for each camera location j, hence $\{\mathcal{Q}_0,\ldots,\mathcal{Q}_M\} \subseteq \mathcal{Q}$.

To perform visibility checks, we generate a virtual image for each view $\mathbf{T}_{ij}$, projecting each landmark in $\mathcal{P}$, given the camera intrinsics. A projection is a mapping $\pi : \mathbb{R}^3 \to \mathit{\Gamma} \subset \mathbb{R}^2$ from a landmark $\mathbf{l}$ to image coordinates $\mathbf{u} = [x_u, y_u]^T$. In our algorithm, we employ the pinhole projection model [18]. However, it can be straightforwardly replaced by any other model. For each camera location, we keep the best orientations that provide the highest number of visible landmarks. In order to detect occlusions, given the sparsity of $\mathcal{P}$, we perform one of the following steps based on the input (i.e., depending on whether dense depth data is available or not):

– with dense depth, we run a Truncated Signed Distance Function (TSDF) integration and store a dense model in a hash table similar to the one originally proposed in [29]. This assumes to have generated a dense model other than $\mathcal{P}$ during mapping. This model can be queried in $O(1)$ given $\mathbf{T}_{ij}$, and occlusions can be detected using a simple z-buffer algorithm;
– without dense depth, we employ a sparse hidden point removal technique [20] based only on sparse data $\mathcal{P}$. Given a viewpoint, this method first transforms the point cloud points to a new range-dependent domain and then constructs the convex hull in that domain.

While the second approach does not require additional input, it is expected to be less accurate due to having significantly less information regarding the scene.

Given the independence of each $\mathbf{T}_{ij}$, we implemented this initial visibility check on the GPU with CUDA for faster computation. After identifying the viewpoints that maximize the visibility of landmarks for each camera position $\mathbf{t}_j$, we sort the viewpoints in descending order, prioritizing visibility.

3.3 Training

Using the best set of orientations $\mathcal{Q}$ we can obtain the full set of viewpoints that maximizes the visibility. For each viewpoint, we compute a feature vector $\mathbf{x}_k$ that contains the visibility information and an expected value y_k representing if the viewpoint is good enough for localization. Therefore, our viewpoint evaluation model learns a scoring function $\mathbf{f}_{\mathcal{P}}(\mathbf{R}, \mathbf{t}) \rightarrow [0, 1]$, that given a set of input features $\mathbf{x}$, provides an estimate of the normalized localization quality. For learning, we employ a mean weighted binary-cross-entropy loss within a MLP encoder, casting the problem to classification:

$$\mathcal{L}(\mathbf{W}) = \frac{1}{|\mathcal{D}|} \sum_{k=1}^{|\mathcal{D}|} w_k (y_k \log p_k + (1 - y_k) \log(1 - p_k)), \tag{3}$$

where $\mathcal{D}$ is the dataset $\mathcal{D} = \{(\mathbf{x}_k, y_k)\}$. The term p_k denotes each sample's classifier output probability, y_k denotes the correspondence target label, and w_k is the weight balancing the negative and positive samples. Additionally, to address overfitting, we incorporate regularization terms into the loss function:

$$\mathcal{L}_{\text{total}}(\mathbf{W}) = \mathcal{L}(\mathbf{W}) + \lambda \sum_{l=1}^{L} \|\mathbf{W}_l\|_2^2, \tag{4}$$

where $\mathbf{W}_l$ represents the weights of layer l in the MLP, and λ controls the regularization strength.

Data. In order to learn our scoring function $\mathbf{f}_{\mathcal{P}}(\mathbf{R}, \mathbf{t})$, we provide $\mathbf{x}_k$ for each camera viewpoint $\mathbf{T}_k$. This involves only the set of visible landmarks $\in \mathbb{R}^{3H}$ and their reprojections in a virtual image $\pi(\mathbf{T}_k^{-1}\mathcal{P}) \in \mathbb{R}^{2H}$. In our experiments,

we try different input data, namely $\mathbf{x}_k \in \mathbb{R}^{2H}$: only landmarks projection, $\mathbf{x}_k \in \mathbb{R}^{3H}$: image projections and depth data, and $\mathbf{x}_k \in \mathbb{R}^{5H}$: image projections and 3D landmarks in camera frame ($\mathbf{T}_k^{-1}\mathcal{P}$). For now, we assume that our best model is the latter; however, in Tab. 3, we show numerically how this different information impacts localization score. A crucial aspect in both localization and image registration is the spatial distribution of features throughout the image [47]. The more uniform this distribution is across the image, the higher the likelihood of obtaining accurate results in the registration process and the less likely to have a degenerate solution [49]. To ensure a consistent and fixed input for our network, particularly given the variations in the number of observable landmarks across diverse viewpoints, we employ image binning (i.e., grouping the reprojected landmarks into discrete cells in the image). In our experiments, we set this grid dimension to be 30×30 bins. Within each bin, we calculate the mean of visible landmarks both in 3D and in 2D. This technique allows a consistent input that considers the distribution of the features in the image rather than performing, for instance, random sampling. Bins without visible landmarks are systematically assigned a zero value. The proposed strategy allows a uniformly distributed fixed input size for the MLP encoder without affecting the complexity of the model and/or additional preprocessing steps. Given the heterogenous input (2D projections, 3D landmarks, or depth landmark data) and to speed up the training process, we standardize the input data. For each feature x, we calculate its corresponding standardized value $z = \frac{x-\mu}{\sigma}$, where μ and σ are the mean and standard deviation of the kind of input. The benefits of centering the feature values around zero are described in [23].

Self-supervision. In our approach, we leverage our structured framework that involves inputting a SfM model and constructing a set of 3D locations denoted as $\mathcal{V}$ and a collection of orientations represented by $\mathcal{Q}$ as explained in Sec. 3.1. For each viewpoint $\mathbf{T}_k$ generated through our methodology, we use a simulator relying on Open3D [56] and Habitat - Matterport 3D meshes [33] to extract a dense RGB image. If the depth image can be queried, we use this to filter occlusions for processing the training set; otherwise, we rely on [20] (Sec. 3.2). This acquired image is fundamental to calculating the expected value y_k. We achieve this by employing COLMAP [44] in global localization mode, where the RGB image is localized against the SfM model. Subsequently, given the ground-truth pose obtained from the simulator, we assess the error in registering the image against the SfM model, considering both rotation and translation. In our setup, 3D landmarks are represented by triangulated SIFT features [27]. If this error falls below a predefined threshold, we label

Layer	Type
1	fully connected input-size $\times$ 300
1.1	ReLU
1.2	dropout 0.5
2	fully connected 300 $\times$ 300
2.1	ReLU
2.2	dropout 0.5
3	fully connected 300 $\times$ 1
3.1	Sigmoid

Table 1: The architecture employed to classify the quality of camera viewpoints.

y_k as a positive sample. Otherwise, it is set to negative. This process ensures autonomously the accuracy and reliability of our labeling mechanism, contributing to the overall effectiveness of our methodology.

4 Experiments

To show the effectiveness of our approach, we conducted different qualitative and quantitative experiments with simulated data and in real-world scenarios. We trained our model using 10 different meshes from Habitat - Matterport 3D [33], and we evaluated the generalization capabilities on 2 different meshes. The training set includes around 250k camera viewpoints, balancing negative and positive samples. In contrast, the test set, where we evaluate our approach, comprises around 92k viewpoints, which are never shown during training and contain around 70% of negative samples and 30% of positive samples. We train our best model of the size shown in Tab. 1, with Adam optimizer and learning rate initialized to 1e-3 for 300 epochs. Taking care of the binning with a grid of 30×30, we feed the network with a flattened input containing the projected landmark in 2D, its depth, or its 3D value. Hence, the total input length is 4220. All experiments and training were conducted on a machine equipped with an Intel Core i7-7700K CPU @ 4.20GHz, featuring 8 cores, and a GeForce GTX 1070 GPU.

4.1 Localization accuracy

To assess how accurately our method localizes, we use metrics from the Long-Term Visual Localization benchmark [41]. Originally designed for outdoor and large-scale settings, the benchmark defined three accuracy ranges (0.25m, 2° / 0.5m, 5° / 5m, 10°). Since we focus on indoor scenes, we include an additional finest range (0.05m, 0.4°) to ensure a meaningful evaluation in this environment. The localization results are reported as the percentage of query images localized within the four given translation and rotation thresholds for each condition.

In our experiments, we compared our method with Fisher Information Fields (FIF) [55] and a few other baseline approaches. FIF utilizes Fisher Information theory, treating the camera as a bearing vector to make information independent from rotation. A Gaussian Process regression is then applied to determine a visibility score. As commonly done in information theory [32], FIF evaluates the information based on the minimum eigenvalue, the determinant, and the trace.

We report all the results in Tab. 2. For baselines, we employed random sampling and reprojection with bins. In the first, we select the target viewpoint randomly, based on the distribution of the test set. In the reprojection with bins approach, we choose the viewpoint that maximizes the number of binned features, ensuring a uniform distribution of features in the image.

Compared to existing methods like FIF, the main advantage of our approach is that it allows multiple camera viewpoints to be considered "good" at each map location. This flexibility means that during planning, one can select the camera

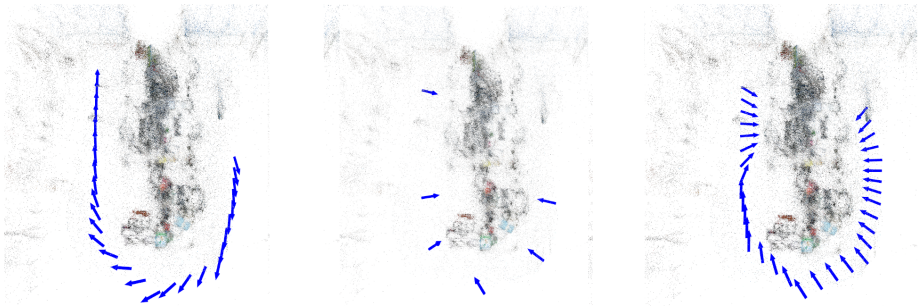
Method	0.05m, 0.4°	/ 0.25m, 2°	/ 0.5m, 5°	/ 5m, 10°
random	54.7	60.9	60.9	61.1
FIF mineig exact	60.1	67.7	69.0	69.6
FIF mineig app	57.3	64.7	66.2	66.4
FIF mineig GP app	59.8	65.4	65.7	66.8
FIF det exact	59.2	65.4	66.2	66.2
FIF det app	60.7	67.5	69.1	69.4
FIF det GP app	58.8	67.0	67.1	67.2
FIF trace app	55.4	61.1	61.8	62.0
FIF trace GP app	-	-	-	-
FIF trace appr worst	17.8	23.3	23.3	23.3
FIF trace appr zero deriv labels	12.5	16.7	17.3	17.3
FIF trace exact	-	-	-	-
reprojections with bins	68.7	71.7	71.7	71.9
proposed	**72.9**	**80.4**	**81.2**	**81.7**

Table 2: Quantitative results. The localization score values [%] on a test set of around 92k camera viewpoints, calculated following the Long-Term Visual Localization [41] benchmark with the addition of the finest scale to target specifically our indoor setting. We compare against FIF [55] using multiple modalities to evaluate the Fisher information matrix, random, and reprojection with bins. The proposed approach leads to the most accurate results. The fact that the proposed method, relying on point reprojections grouped into bins, significantly outperforms FIF demonstrates that the distribution of landmarks is more important than evaluating the Fisher information matrix in visual localization.

viewpoint that offers the most convenience—such as the shortest path—or the best balance between accuracy for localization and planning convenience. In Fig. 7, we show how predicting multiple viewpoints for each camera impacts the localization score of the Long-Term Visual Localization benchmark [41]. We specifically experimented with 1, 5, and 10 best directions for each camera location.

4.2 Planning experiment

We assess the planning quality qualitatively. For 3D planning, we utilized RRT* [19], a planner that iteratively grows the tree by sampling random configurations in the configuration space, with our state represented as $\mathbb{SE}(3)$. In our planning comparisons, we used Fisher information fields and a classic camera look-forward approach. During experiments, FIF fails to predict some viewpoints, resulting in a failure state. The conventional camera look-forward approach is a traditional method that overlooks the consideration of active viewpoints, and its drawbacks become evident as it fails to adjust the camera towards regions with higher landmark density, potentially pointing towards featureless areas. In contrast, our approach achieves meaningful planning, directing the camera towards regions with higher landmark density. These qualitative results are depicted in Fig. 5. We specifically use a noisy SfM model to show the robustness and adaptability of our approach, reflecting real case experiments. The SfM model has been generated



(a) Forward-facing camera **(b)** FIF [55] **(c)** Ours

Fig. 5: Qualitative planning experiments with self-recorded data. In this setup, FIF encounters challenges in predicting certain viewpoints, leading to failures in our planner. The conventional camera look-forward approach, a traditional method, neglects the consideration of active viewpoints. Its limitations become apparent as it neglects camera adjustments toward regions with higher landmark density, potentially directing it toward featureless areas. We specifically use a noisy SfM model to show the robustness and adaptability of our approach, reflecting real case experiments.

using only sparse COLMAP [44] reconstruction, without given poses, using only a set of images as input. We used markers to retrieve the original scale.

4.3 Runtimes

We adopt a simple model to learn the active camera viewpoints to make inferences promptly, designed for dedicated devices and robotics. Although a complete map representation can be pre-generated, during motion planning or localization, one can simply query the hash table based on the map location. One inference step of our model takes around 0.02 seconds with our setup. In addition, given the exhaustive approach adopted to subsample good camera viewpoints described in Sec. 3.1, and given the independence of each camera viewpoint, we make our implementation in CUDA leveraging parallelism.

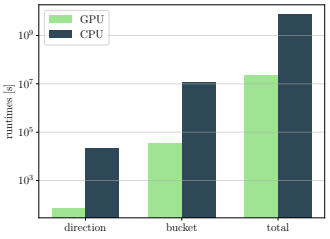


Fig. 6: Runtimes of our GPU sampling method compared with a single-threaded CPU implementation. These experiments are done by sampling from 648 camera locations with 24919 3D landmarks from the SfM model.

In Fig. 6, we illustrate the runtimes of our sampling compared to a single-threaded CPU implementation, presenting the plot in log scale for clarity. Numerically, the GPU implementation completes the initial sampling in around 23 seconds, while the CPU implementation takes approximately 7387 seconds. These experiments involve sampling from 648 camera locations with 24919 landmarks from the SfM model.

5 Ablation Study

Within this study, our primary focus is on extracting valuable insights from environmental geometry to advance active visual localization. For this, we introduce a data-driven approach employing a compact architecture designed for real-time operations, a novel self-supervised training method, the development of a unique map representation allowing specific voxel locations in space to possess one or more active viewpoints, and the possibility to integrate motion planning for robotics applications.

The discrete map representation presented in this work using a voxel grid, parameterizing each camera location as a voxel, is similar to the one proposed in [54]. The main difference is that within our active viewpoint selection, the independence of each viewpoint in the classification process allows the possibility of selecting more camera viewpoints for each voxel location. This makes things easier during motion planning, for example, because the planner can rely on the viewpoint in the location that minimizes the cost concerning its current state (i.e. shortest path). How predicting multiple directions within our methodology from each camera location impacts the localization score is analyzed in Fig. 7.

The results obtained in our experiments demonstrate how the distribution of observed landmarks in the image impacts visual localization tasks. This seems more important than exploiting the Fisher information matrix, which usually gives more importance to the vicinity, generally maximizing visibility, without considering the distribution of the observed map in the image.

In our experiments, we investigated how incorporating various types of information affects the quality of viewpoint selection. We explored scenarios with only 2D visible landmarks and introduced the full 3D landmarks (in relative camera frame) or only their depth. The detailed results are presented in Tab. 3. Notably, 3D information significantly enhances our model's accuracy. In general, depth information alone is sufficient to achieve good results. While incorporating the full 3D geometry can improve performance, it may also lead to overfitting, likely due to data quality issues or its limited usefulness for generalization.

We evaluated our model through quantitative testing on simulated data (Tab. 2) and qualitative analysis on real data (Fig. 5). It is worth noting that our approach never explicitly generates a complete RGB image during training (it is used just to calculate the expected value for supervision). Instead, it relies solely on geometric information, making it independent of specific digital details encoded in RGB images. This characteristic significantly enhances the model's generalization ability, enabling transitions from photorealistic simulated data to real-world scenarios. This capability is demonstrated in the planning experiments conducted with real data illustrated in Fig. 5.

We specifically use a simple MLP encoder since it does not require any preprocessing, unlike, for example, a Graph Neural Network (GNN), where a graph for each sample must be created, or a Convolutional Neural Network (CNN), where the full image should be convoluted. Also, training and inference are fast. While more complex models might offer better accuracy, exploiting the corre-

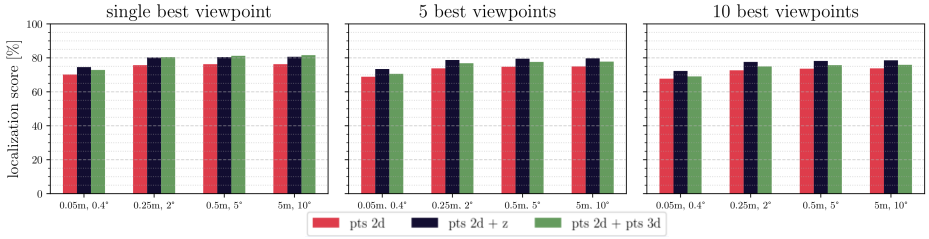


Fig. 7: Multiple viewpoints results. This figure demonstrates the impact of predicting multiple viewpoints on localization score according to [41], experimenting with 1, 5, and 10 directions per location. Compared to methods like FIF, our approach's main advantage is allowing multiple "good" camera viewpoints at each map location. This flexibility lets planners choose the most convenient viewpoint, such as the shortest path or the best balance between localization accuracy and convenience. In this experiment, we collected all the predicting viewpoints for each camera location, sorted the likelihood output from the MLP encoder in descending order, and took the best N elements for evaluation.

lation between landmarks, we aim to keep the approach simple to enhance its adaptability to real conditions.

approach	0.05m, 0.4° / 0.25m, 2°/ 0.5m, 5° / 5m, 10°			
ours (pts 2d)	70.2	75.7	76.2	76.2
ours (pts 2d + z)	**74.6**	80.3	80.4	80.6
ours (pts 2d + pts 3d)	72.9	**80.4**	**81.2**	**81.7**

Table 3: Impact of different geometrical information in our viewpoint selection strategy. We examined situations first with only 2D visible landmarks, then introduced landmark depth (along z-camera axis), and finally, considered the full 3D points. Intuitively, the 3D data improves localization scores compared to 2D-only cases. Localization score according to [41] is reported in %.

6 Conclusion

This paper explores active localization, highlighting viewpoint selection's crucial role in refining localization accuracy. Our contributions involve a data-driven approach with a simple architecture designed for real-time operation, introducing a self-supervised data training method. We show the capabilities of our viewpoints map to be integrated into a planning framework for robotics applications. We conducted both qualitative and numerical experiments on simulated and real data. Our results demonstrate the performance of our method compared to existing solutions for similar challenges, proving its effectiveness. For the future, we envision a more robust model, developed ad-hoc for selecting the best camera viewpoints and an active localization benchmark to benefit the community.

Acknowledgments

This work has been partially supported by Sapienza University of Rome as part of the work for project *H&M: Hyperspectral and Multispectral Fruit Sugar Content Estimation for Robot Harvesting Operations in Difficult Environments*, Del. SA n.36/2022, by the Hasler Stiftung Research Grant via the ETH Zurich Foundation and an ETH Zurich Career Seed Award.

References

1. Bartolomei, L., Teixeira, L., Chli, M.: Semantic-aware Active Perception for UAVs using Deep Reinforcement Learning. Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS) pp. 3101–3108 (2021). https://doi.org/10.1109/IROS51168.2021.9635893 4
2. Brachmann, E., Rother, C.: Learning less is more-6d camera localization via 3d surface regression. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 4654–4662 (2018) 3
3. Brachmann, E., Rother, C.: Expert sample consensus applied to camera re-localization. In: Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV). pp. 7525–7534 (2019) 3
4. Brizi, L., Giacomini, E., Di Giammarino, L., Ferrari, S., Salem, O., De Rebotti, L., Grisetti, G.: Vbr: A vision benchmark in rome. In: Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA). IEEE (2024) 1
5. Burgard, W., Fox, D., Thrun, S.: Active mobile robot localization. In: Proc. of the Intl. Conf. on Artificial Intelligence (IJCAI). pp. 1346–1352. Citeseer (1997) 2
6. Cavallari, T., Golodetz, S., Lord, N.A., Valentin, J., Prisacariu, V.A., Di Stefano, L., Torr, P.H.: Real-time rgb-d camera pose estimation in novel scenes using a relocalisation cascade. IEEE Trans. on Pattern Analalysis and Machine Intelligence (TPAMI) **42**(10), 2465–2477 (2019) 3
7. Chaplot, D.S., Parisotto, E., Salakhutdinov, R.: Active neural localization. arXiv preprint arXiv:1801.08214 (2018) 4
8. Chen, S., Li, X., Wang, Z., Prisacariu, V.A.: Dfnet: Enhance absolute pose regression with direct feature matching. In: Proc. of the Europ. Conf. on Computer Vision (ECCV). pp. 1–17. Springer (2022) 3
9. Clark, R., Wang, S., Markham, A., Trigoni, N., Wen, H.: Vidloc: A deep spatio-temporal model for 6-dof video-clip relocalization. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 6856–6864 (2017) 4
10. Costante, G., Forster, C., Delmerico, J., Valigi, P., Scaramuzza, D.: Perception-aware path planning. arXiv preprint arXiv:1605.04151 (2016) 2, 3
11. Ding, M., Wang, Z., Sun, J., Shi, J., Luo, P.: Camnet: Coarse-to-fine retrieval for camera re-localization. In: Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV). pp. 2871–2880 (2019) 3
12. Dong, S., Wang, S., Zhuang, Y., Kannala, J., Pollefeys, M., Chen, B.: Visual localization via few-shot scene region classification. In: 2022 International Conference on 3D Vision (3DV). pp. 393–402. IEEE (2022) 3
13. Dusmanu, M., Rocco, I., Pajdla, T., Pollefeys, M., Sivic, J., Torii, A., Sattler, T.: D2-net: A trainable cnn for joint description and detection of local features. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 8092–8101 (2019) 3

14. Fang, Q., Yin, Y., Fan, Q., Xia, F., Dong, S., Wang, S., Wang, J., Guibas, L.J., Chen, B.: Towards accurate active camera localization. In: Proc. of the Europ. Conf. on Computer Vision (ECCV). pp. 122–139. Springer (2022) 4
15. Fontanelli, D., Salaris, P., Belo, F.A., Bicchi, A.: Visual appearance mapping for optimal vision based servoing. In: Experimental Robotics: The Eleventh International Symposium. pp. 353–362. Springer (2009) 3
16. González, Á.: Measurement of areas on a sphere using fibonacci and latitude–longitude lattices. Mathematical Geosciences **42**, 49–64 (2010) 7
17. Hanlon, M., Sun, B., Pollefeys, M., Blum, H.: Active visual localization for multi-agent collaboration: A data-driven approach. In: Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA). IEEE (2024) 1, 2, 3, 4
18. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge University Press (2003) 7
19. Karaman, S., Frazzoli, E.: Sampling-based algorithms for optimal motion planning. Intl. Journal of Robotics Research (IJRR) **30**(7), 846–894 (2011) 11
20. Katz, S., Tal, A., Basri, R.: Direct visibility of point sets. In: ACM Trans. on Graphics, pp. 24–es (2007) 8, 9
21. Kendall, A., Grimes, M., Cipolla, R.: Posenet: A convolutional network for real-time 6-dof camera relocalization. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 2938–2946 (2015) 4
22. Kim, A., Eustice, R.M.: Perception-driven navigation: Active visual slam for robotic area coverage. In: Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA). pp. 3196–3203 (2013). https://doi.org/10.1109/ICRA.2013.6631022 3
23. LeCun, Y., Bottou, L., Orr, G.B., Müller, K.R.: Efficient backprop. In: Neural networks: Tricks of the trade, pp. 9–50. Springer (2002) 9
24. Li, X., Wang, S., Zhao, Y., Verbeek, J., Kannala, J.: Hierarchical scene coordinate classification and regression for visual localization. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 11983–11992 (2020) 3
25. Lim, J., Lawrance, N., Achermann, F., Stastny, T., Bähnemann, R., Siegwart, R.: Fisher information based active planning for aerial photogrammetry. In: Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA). pp. 1249–1255 (2023). https://doi.org/10.1109/ICRA48891.2023.10161136 3
26. Lodel, M., Brito, B., Serra-Gómez, A., Ferranti, L., Babuska, R., Alonso-Mora, J.: Where to look next: Learning viewpoint recommendations for informative trajectory planning. In: Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA). pp. 4466–4472. IEEE (2022) 4
27. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Intl. Journal of Computer Vision (IJCV) **60**, 91–110 (2004) 9
28. Moreau, A., Piasco, N., Tsishkou, D., Stanciulescu, B., de La Fortelle, A.: Lens: Localization enhanced by nerf synthesis. In: Conference on Robot Learning. pp. 1347–1356. PMLR (2022) 3
29. Nießner, M., Zollhöfer, M., Izadi, S., Stamminger, M.: Real-time 3d reconstruction at scale using voxel hashing. ACM Trans. on Graphics **32**(6), 1–11 (2013) 8
30. Panek, V., Kukelova, Z., Sattler, T.: Meshloc: Mesh-based visual localization. In: Proc. of the Europ. Conf. on Computer Vision (ECCV). pp. 589–609. Springer (2022) 3
31. Papachristos, C., Khattak, S., Alexis, K.: Uncertainty-aware receding horizon exploration and mapping using aerial robots. In: Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA). pp. 4568–4575 (2017). https://doi.org/10.1109/ICRA.2017.7989531 3

32. Placed, J.A., Strader, J., Carrillo, H., Atanasov, N., Indelman, V., Carlone, L., Castellanos, J.A.: A survey on active simultaneous localization and mapping: State of the art and new frontiers. IEEE Trans. on Robotics (TRO) (2023) 10
33. Ramakrishnan, S.K., Gokaslan, A., Wijmans, E., Maksymets, O., Clegg, A., Turner, J., Undersander, E., Galuba, W., Westbury, A., Chang, A.X., et al.: Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. arXiv preprint arXiv:2109.08238 (2021) 9, 10
34. Roy, N., Burgard, W., Fox, D., Thrun, S.: Coastal navigation-mobile robot navigation with uncertainty in dynamic environments. In: Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA). vol. 1, pp. 35–40. IEEE (1999) 2, 3
35. Saraceni, L., Motoi, I.M., Nardi, D., Ciarfuglia, T.A.: Agrisort: A simple online real-time tracking-by-detection framework for robotics in precision agriculture. In: Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA). IEEE (2024) 1
36. Sarlin, P.E., Cadena, C., Siegwart, R., Dymczyk, M.: From coarse to fine: Robust hierarchical localization at large scale. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 12716–12725 (2019) 3
37. Sarlin, P.E., Unagar, A., Larsson, M., Germain, H., Toft, C., Larsson, V., Pollefeys, M., Lepetit, V., Hammarstrand, L., Kahl, F., et al.: Back to the feature: Learning robust camera localization from pixels to pose. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 3247–3257 (2021) 3
38. Sattler, T., Leibe, B., Kobbelt, L.: Fast image-based localization using direct 2d-to-3d matching. In: Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV). pp. 667–674. IEEE (2011) 3
39. Sattler, T., Leibe, B., Kobbelt, L.: Improving image-based localization by active correspondence search. In: Proc. of the Europ. Conf. on Computer Vision (ECCV). pp. 752–765. Springer (2012) 3
40. Sattler, T., Leibe, B., Kobbelt, L.: Efficient & effective prioritized matching for large-scale image-based localization. IEEE Trans. on Pattern Analalysis and Machine Intelligence (TPAMI) **39**(9), 1744–1756 (2016) 3
41. Sattler, T., Maddern, W., Toft, C., Torii, A., Hammarstrand, L., Stenborg, E., Safari, D., Okutomi, M., Pollefeys, M., Sivic, J., et al.: Benchmarking 6dof outdoor visual localization in changing conditions. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 8601–8610 (2018) 10, 11, 14
42. Sattler, T., Torii, A., Sivic, J., Pollefeys, M., Taira, H., Okutomi, M., Pajdla, T.: Are large-scale 3d models really necessary for accurate visual localization? In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 1637–1646 (2017) 3
43. Sattler, T., Zhou, Q., Pollefeys, M., Leal-Taixe, L.: Understanding the limitations of cnn-based absolute camera pose regression. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 3302–3312 (2019) 3
44. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 4104–4113 (2016) 9, 12
45. Shavit, Y., Ferens, R., Keller, Y.: Learning multi-scene absolute pose regression with transformers. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 2733–2742 (2021) 3
46. Shavit, Y., Keller, Y.: Camera pose auto-encoders for improving pose regression. In: Proc. of the Europ. Conf. on Computer Vision (ECCV). pp. 140–157. Springer (2022) 3
47. Shi, J., et al.: Good features to track. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 593–600. IEEE (1994) 9

48. Tang, S., Tang, C., Huang, R., Zhu, S., Tan, P.: Learning camera localization via dense scene matching. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 1831–1841 (2021) 3
49. Torr, P.H., Zisserman, A., Maybank, S.J.: Robust detection of degenerate configurations while estimating the fundamental matrix. Computer Vision and Image Understanding **71**(3), 312–333 (1998) 9
50. Wang, B., Chen, C., Lu, C.X., Zhao, P., Trigoni, N., Markham, A.: Atloc: Attention guided camera localization. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 10393–10401 (2020) 3
51. Xue, F., Wu, X., Cai, S., Wang, J.: Learning multi-view camera relocalization with graph neural networks. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 11372–11381. IEEE (2020) 3
52. Yan, Q., Zheng, J., Reding, S., Li, S., Doytchinov, I.: Crossloc: Scalable aerial localization assisted by multimodal synthetic data. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 17358–17368 (2022) 3
53. Yang, L., Bai, Z., Tang, C., Li, H., Furukawa, Y., Tan, P.: Sanet: Scene agnostic network for camera localization. In: Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV). pp. 42–51 (2019) 3
54. Zhang, Z., Scaramuzza, D.: Perception-aware receding horizon navigation for mavs. In: Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA). pp. 2534–2541. IEEE (2018) 3, 13
55. Zhang, Z., Scaramuzza, D.: Beyond point clouds: Fisher information field for active visual localization. In: Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA). pp. 5986–5992. IEEE (2019) 2, 3, 10, 11, 12
56. Zhou, Q.Y., Park, J., Koltun, V.: Open3d: A modern library for 3d data processing. arXiv preprint arXiv:1801.09847 (2018) 9