FocusTune: Tuning Visual Localization through Focus-Guided Sampling

Son Tung Nguyen   Alejandro Fontan   Michael Milford   Tobias Fischer

Queensland University of Technology
Brisbane, Australia

sontung.nguyen@hdr.qut.edu.au

Abstract

We propose FocusTune, a focus-guided sampling technique to improve the performance of visual localization algorithms. FocusTune directs a scene coordinate regression model towards regions critical for 3D point triangulation by exploiting key geometric constraints. Specifically, rather than uniformly sampling points across the image for training the scene coordinate regression model, we instead re-project 3D scene coordinates onto the 2D image plane and sample within a local neighborhood of the re-projected points. While our proposed sampling strategy is generally applicable, we showcase FocusTune by integrating it with the recently introduced Accelerated Coordinate Encoding (ACE) model. Our results demonstrate that FocusTune both improves or matches state-of-the-art performance whilst keeping ACE's appealing low storage and compute requirements, for example reducing translation error from 25 to 19 and 17 to 15 cm for single and ensemble models, respectively, on the Cambridge Landmarks dataset. This combination of high performance and low compute and storage requirements is particularly promising for applications in areas like mobile robotics and augmented reality. We made our code available at https: //github.com/sontung/focus-tune.

1. Introduction

Visual localization remains a cornerstone task in computer vision, with substantial ramifications for applications in mobile robotics [22, 34] and augmented reality (AR) [1, 12]. The primary objective is to determine the pose – both position and orientation – of a query image relative to a pre-established environment map. Such maps typically comprise three components: a set of RGB images, a 3D point cloud approximating the environment's geometry, and individual camera poses corresponding to each image.

Historically, structured methods have been employed to solve the visual localization problem, utilizing feature correspondences between 2D pixels in the query image and 3D

points within the environmental map [33, 43, 44]. To manage the computational complexity, these methods often resort to clustering and quantizing the search space. While they offer excellent accuracy and low query times, their memory footprint can be prohibitive for large-scale environments. In this paper, we further close the performance gap to structured methods while retaining storage and training efficiency.

Modern regression-based techniques offer an alternative to structured methods by embedding the environmental map into the neural network's architecture, thus sidestepping the memory constraint. Although Absolute Pose Regression (APR) models [27, 31, 47, 52, 55], which directly regress the camera pose, have gained traction, they suffer from accuracy limitations, mainly due to unresolved questions around their optimal training objectives [8, 9, 26, 27].

Scene Coordinate Regression (SCR) models [3, 5, 7, 8, 13, 19, 23, 32, 48] represent a marked advancement in this regard, trading pose regression for scene coordinate regression and achieving higher accuracy. These models regress the 3D coordinates of the 2D pixels in an image, as opposed to directly regressing the camera pose as in APR models, which allows them to be trained via the re-projection error.

One high-performing popular SCR model is DSAC* [8], where SCR was combined with differentiable RANSAC. However, the training time for DSAC* [8] can be up to 15 hours for a given scene. Recently, the Accelerated Coordinate Encoding (ACE) model [3] emerged as a rapid training alternative, substantially reducing computational time to just 5 minutes by splitting the network into two parts: a scene-agnostic pre-trained backbone and scene-specific regression heads. ACE relies on a training buffer from which high-dimensional feature vectors are extracted for various regions in an image.

In this paper, we demonstrate that ACE's training buffer contains a considerable number of non-informative feature vectors, thus hindering the optimal training of the scene-specific regression heads. Our proposed FocusTune heuristic sampler guides the training process to focus on impor-
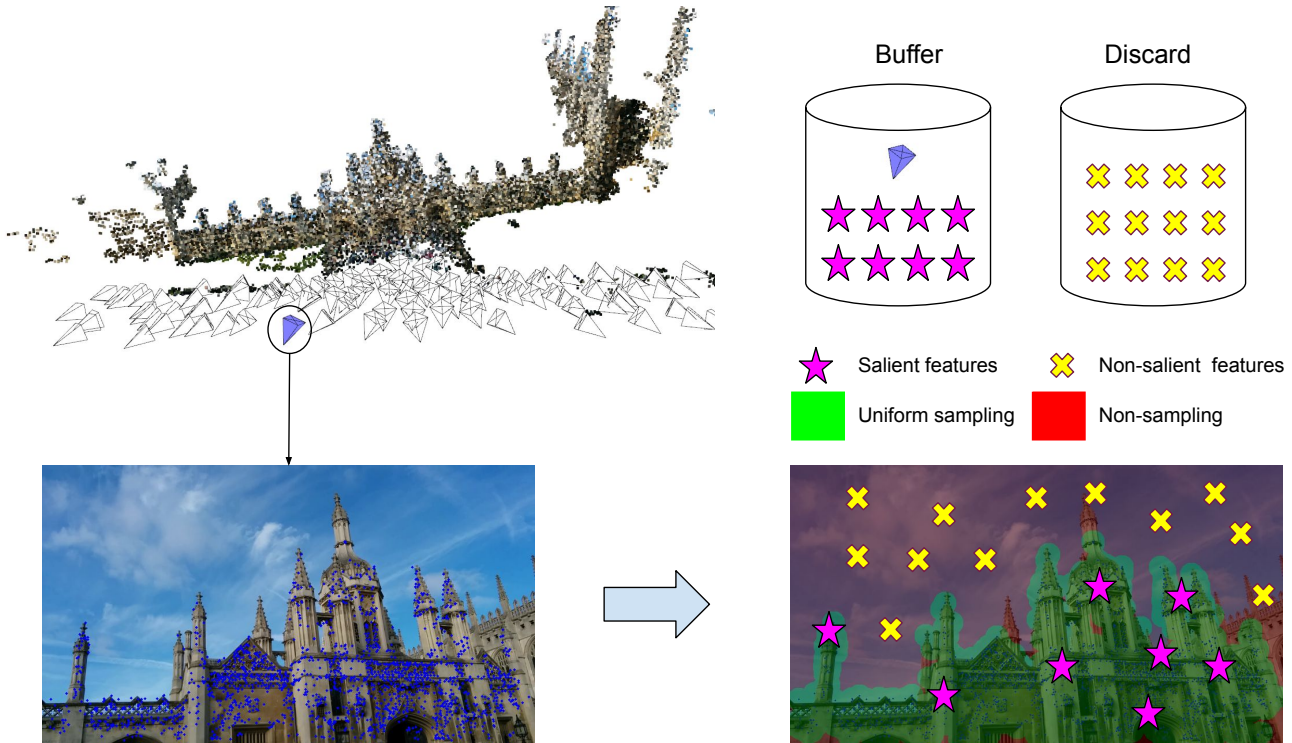
Figure 1. **System Overview.** For every training image, using the associated camera pose *(purple color)* we re-project the map *(top-left image)* onto that image to obtain a set of seed keypoints *(blue dots in the bottom-left image)*. We then sample training instances uniformly within a region of ρ pixels surrounding the seed keypoints *(green region of the bottom-right image)* and prohibit sampling from outside these circles *(red region of the bottom-right image)*. As a result, our training instances are sampled from important regions of the image *(pink stars in the bottom-right image)* rather than texture-less regions *(yellow crosses in the bottom-right image)*.

tant parts of the scene, specifically those that form the 3D point cloud map. This results in marked performance improvements with minimal overhead to the training time. The sampling process is illustrated in Fig. 1.

We summarize our contributions as follows:

1. We introduce FocusTune, a focus-guided heuristic sampling strategy to avoid optimizing parts of the scene that are non-informative and instead leveraging geometrical constraints. FocusTune incurs minimal computational overhead.

2. We validate and implement FocusTune within the ACE framework [3], thereby exemplifying FocusTune's compatibility and effectiveness in a real-world setting.

3. We provide empirical evidence showing marked performance gains, including a reduction in translation error from 25cm to 19cm on the Cambridge Landmarks dataset when using a single model, approaching the performance of computationally intensive methods like DSAC*. When using an ensemble scheme, FocusTune results in a new state-of-the-art translation error of just 9cm across all prior visual localization techniques, including structured methods that require substantially more storage.

2. Related Works

2.1. Feature Detection

Feature detection serves as a foundational element in numerous geometric computer vision applications, such as Simultaneous Localization and Mapping (SLAM) [10] and Structure from Motion (SfM) [46]. These applications rely on the robust identification of pixels, or features, that exhibit consistent characteristics across varying lighting conditions and viewpoints. Such features typically originate from 2D pixels with distinct textural properties, as opposed to texture-deficient uniform areas. The conventional approach for feature detection employs the Difference-of-Gaussian technique [35]. This method involves iteratively smoothing the input image across multiple scales and computing the differences between successive scales. Pixels that remain sharp through this process are designated as features. Owing to its computational efficiency, this technique has been widely adopted in prevalent feature description algorithms [2, 35, 36].

More recent developments [18, 21, 38] have explored the use of neural networks for feature detection. For example, the method described in [38] utilizes an attention layer

to aggregate the semantic significance of local features, thereby facilitating the identification of the most relevant features while also providing reliable confidence scores. Conversely, [18] presents a learning-based corner detection model trained on a large synthetic dataset comprising basic shapes like triangles and cubes.

Our work builds upon this expansive body of literature, demonstrating that learning-based localization systems can also achieve enhanced performance by focusing on textured regions during training.

2.2. Structured Methods

Classical structured solutions [25, 33, 43–45] for visual localization tasks establish 2D-3D correspondences between query images and environment maps. The nearest-neighbor search is typically done by comparing the feature descriptors [35] of the query image and the descriptors of the 3D points. Therefore, structured methods require access to the descriptor database of the map at run time, making them unsuitable for memory-constrained devices. While GoMatch [55] eliminates the need for storing descriptors, this comes at the expense of reduced accuracy.

2.3. Absolute Pose Regression

The advent of deep learning has enabled absolute pose regression (APR) models to directly regress camera poses from query images [27, 31, 47, 52, 55]. APR models output the rotation (e.g. represented by a quaternion) and the translation vectors of the corresponding camera pose to the input image. While earlier methods [27] relied on a pre-trained GoogLeNet [49] as the backbone, more recent network architectures increase the performance of APR models with spatial LSTMs [52] or transformers [31, 47, 50]. However, despite being fast and requiring no access to maps at query time, the accuracy of APR models is still far behind classical structured methods.

2.4. Scene Coordinate Regression

Scene coordinate regression models predict the scene coordinates for the pixels in the query images [8,14,32,48,51]. SCR models can be implemented by random forests [48,51] or convolutional neural networks [3,5–8,19,23]. SCR models are often optimized via the re-projection error, which works well in practice, thus leading to a much higher accuracy than APR models that are competitive with structured methods in certain settings. Up until recently, the mapping time of SCR models was problematic, often requiring hours of mapping time [7, 8] for convergence.

2.5. Low Mapping Time Scene Coordinate Regression

Multiple approaches were proposed in the literature to reduce the mapping time of SCR models [3, 15, 20, 48, 54].

SCoRF [48] showed a random forest can localize well in just 10 minutes. [15] leverages depth information to further increase the accuracy. SAnet [54] requires access to database scene images to retrieve and interpolate database scene coordinates. Despite being scene-agnostic and having a short mapping time, its weaknesses include higher memory requirement and lower accuracy compared to classical structured methods.

[3,20] recently proposed to divide the SCR network into two parts: scene-agnostic pre-trained backbone and scene-specific regression heads. The backbone is trained once on a large dataset [16], whereas the regression heads are trained specifically for a given scene. [20] uses another convolutional layer to implement the regression head. This design choice cuts the mapping time to only 2 minutes, however, increases the memory footprint and decreases the accuracy. On the other hand, ACE [3] uses multi-layer perceptron layers to realize the regression heads, leading to state-of-the-art results on both indoor and outdoor settings with only RGB mapping images.

In this paper, we show that it is possible to further extend the accuracy of ACE models with a small overhead at training time by heuristic sampling of the training buffer.

3. Preliminaries

3.1. Problem Statement

Given a dataset of N grayscale images $\mathcal{D} = \{I_1, I_2, \ldots, I_N\}$ and an associated ground-truth map produced via a Structure-from-Motion (SfM) algorithm [46, 53], the problem is to estimate the 6-DoF camera pose $H_i \in \text{SE}(3)$ for a given image $I_i \in \mathcal{D}$. Here, the camera pose H_i is an element of $\text{SE}(3)$ which refers to the special Euclidean group that represents rigid body transformations in 3D space, including both rotation and translation.

To accomplish this, we approximate a function $\tilde{f}_{(I,W)} = X$, where $X \in \mathbb{R}^{3 \times \frac{w}{8} \times \frac{h}{8}}$ contains the scene coordinates $\mathbf{x}_{ij} \in \mathbb{R}^3$ for every j-th pixel in downsampled image I_i, using a neural network parameterized by weights W. Therefore, f implements a mapping from grayscale images to 3D coordinates, $\tilde{f}_{(I,W)} : \mathbb{R}^{w \times h} \rightarrow \mathbb{R}^{3 \times \frac{w}{8} \times \frac{h}{8}}$. The network is trained on images $I_i \in \mathcal{D}$ and the corresponding ground-truth poses H_i by minimizing a re-projection objective given by:

$$r(\mathbf{x}_{ij}, H_i) = ||\hat{\mathbf{y}}_{ij} - K_i H_i^{-1} \hat{\mathbf{x}}_{ij}||_1, \tag{1}$$

where $K_i \in \mathbb{R}^{3 \times 3}$ is the camera calibration matrix, $\mathbf{y}_{ij} \in \mathbb{R}^2$ denotes the j-th pixel coordinate in the pixel grid of I_i and $\mathbf{x}_{ij} \in \mathbb{R}^3$ represents the j-th predicted scene coordinate. The hat operator $\hat{\cdot}$ denotes the homogeneous representation and $|| \cdot ||_1$ the L1-norm. While the fundamental objective remains consistent with Eq. 1, more complex objectives are used in practice; we refer the reader to [3, 8] for
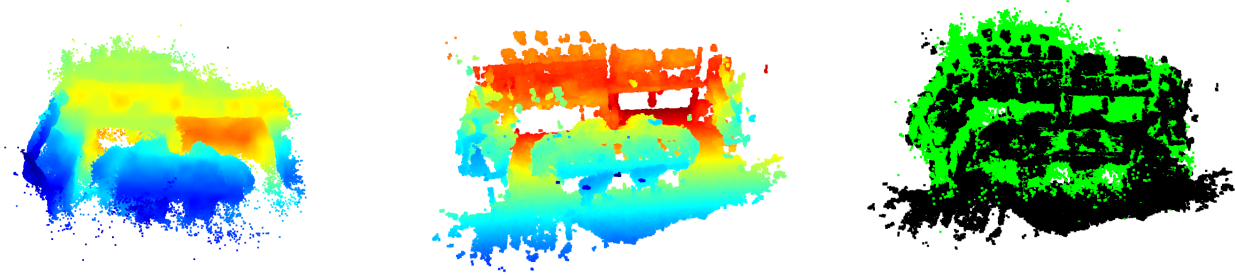
Figure 2. **Scene Geometry.** We plot the 3D geometry of the redkitchen sequence of the 7-scenes dataset [48]: *(left)* point cloud generated by the network after training, *(middle)* ground-truth point cloud, *(right)* ground-truth point cloud (black) overlaid on learned point cloud (green) in the same coordinate system. The fact that the learned geometry converged to the ground-truth geometry strongly suggests the existence of implicit triangulation among similar descriptors.

more details.

Finally, the camera pose can be computed given pairs of 2D-3D correspondences $\{(\mathbf{y}_{ij}, \mathbf{x}_{ij})\}$ using an off-the-shelf Perspective-n-Point solver [29, 39].

3.2. Accelerated Coordinate Encoding

While FocusTune is a generic method that is in principle compatible with a range of SCR methods, we have chosen to demonstrate our sampling technique within the Accelerated Coordinate Encoding (ACE) [3] framework that we briefly describe in this section. ACE partitions the neural network function $\tilde{f}_{(I,W)}$ into a convolutional backbone f_{B} (which only needs to be trained once and remains the same across all scenes) and a multi-layer perceptron (MLP) head f_{H} (which is scene specific). Formally, the convolutional backbone $f_{\mathrm{B}} : \mathbb{R}^{h \times w} \to \mathbb{R}^{512 \times \frac{h}{8} \times \frac{w}{8}}$, pre-trained on Scan-Net [16], generates dense local feature descriptors $\mathbf{d}_{ij} \in \mathbb{R}^{512}$ for a query image I_i. Subsequently, each descriptor $\mathbf{d}_{ij}$ is passed to the regression head $f_{\mathrm{H}} : \mathbb{R}^{512 \times \frac{h}{8} \times \frac{w}{8}} \to \mathbb{R}^{3 \times \frac{h}{8} \times \frac{w}{8}}$ to generate scene coordinates $\mathbf{x}_{ij} \in \mathbb{R}^3$. Every scene coordinate $\mathbf{x}_{ij}$ corresponds to the pixel coordinate $\mathbf{y}_{ij} \in \mathbb{R}^2$ of the descriptor $\mathbf{d}_{ij}$ to form a pair of 2D-3D correspondences $\{(\mathbf{y}_{ij}, \mathbf{x}_{ij})\}$ to re-localize the image I_i.

For training, ACE employs a buffer $\mathbf{B}$ containing sampled feature descriptors along with their geometric constraints. Each instance $\mathcal{G}_k \in \mathbf{B}$ is expressed as $(\mathbf{d}_k, \mathbf{y}_k, K_k, H_k)$, where $\mathbf{d}_k$ is the descriptor vector at pixel coordinate $\mathbf{y}_k$ in an image I with associated intrinsic matrix K_k and extrinsic matrix H_k. Note that the index notation changes from ij at the image level (Eq. 1) to k at the buffer level because a fixed amount of pixels is sampled from a particular training image for the buffer.

For larger scenes like those in the Cambridge Landmarks dataset [27], it is beneficial to divide the scene into multiple smaller sub-scenes m and train a separate f_{H_m} for each of them. At run time, all f_{H_m} models are queried and the prediction with the highest inlier count is selected [3].

4. Methodology

4.1. Implicit Triangulation

The effectiveness of Accelerated Coordinate Encoding (ACE) is inherently linked to its ability to infer ground-truth scene geometry with precision (see Fig. 2). While the re-projection error in Eq. 1 does not explicitly address the depth attributes of the descriptors, the network function f_{H} is conditioned to perform implicit triangulation through the training buffer $\mathbf{B}$. Specifically, f_{H} is required to produce consistent scene coordinates for similar descriptors $(\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3, \dots)$ by leveraging the geometric constraints such as $\{(\mathbf{y}_1, K_1, H_1), (\mathbf{y}_2, K_2, H_2), (\mathbf{y}_3, K_3, H_3), \dots\}$.

Triangulation becomes feasible only when the 3D lines parameterized by the geometrical constraints $\{(\mathbf{y}_k, K_k, H_k)\}$ intersect at a common 3D point. However, tracking the feasibility of the geometrical constraints is intractable since there is no easy way to know exactly which constraints take part in the triangulation of a particular point.

We made a key observation that triangulation is often possible for 3D points with unique repeatable appearances rather than texture-less or ambiguous regions (refer to Fig. 3). To identify such regions from the training images, a straightforward strategy might involve off-the-shelf feature detectors [18, 37] or foreground/background segmentation models [28, 30]. However, during our experiments, these methods did not consistently yield effective results. On the other hand, the SfM program already employed robust local feature methods [2, 18, 21, 35, 37] to identify these points (Fig. 3, top row) during the reconstruction phase. Therefore, instead of employing a third-party system to identify reliable features for the training buffer $\mathbf{B}$, we rely on the SfM map to reduce the likelihood of sampling infeasible triangulation (Fig. 3, bottom row).

Figure 3. **Examples of sampled instances with similar descriptors in the buffer B.** The green dot denotes the projection of the network prediction given the descriptor. *(Top row)* A unique appearance point leads to successful triangulation with an average re-projection error of 0.74 pixels. *(Bottom row)* An ambiguous appearance point leads to unsuccessful triangulation with an average re-projection error of 72.3 pixels.

4.2. Guided Buffer Sampling

To alleviate the issues associated with non-distinctive areas of the scene, we introduce a simple sampling strategy for populating the buffer $\mathbf{B}$. For each query image I_i, we employ a pre-defined set of seed keypoints $\mathcal{Y}_i$ that are generated by re-projecting the corresponding 3D points from the SfM map into the 2D image plane using the ground-truth camera poses (see Fig. 1).

To generate the set of seed keypoints $\mathcal{Y}_i$ for each image I_i, we initially retrieve the 3D coordinates $\mathcal{X}_i$ of the 3D points that are visible in the image I_i from the SfM map. These coordinates are re-projected into the 2D image plane using the ground-truth intrinsic and extrinsic matrices, resulting in $\mathcal{Y}_i = K_i H_i^{-1} \mathcal{X}_i$.

Due to random rotations and resizing applied to the camera matrices due to the data augmentation process during training time, some of the re-projected pixel coordinates may fall outside the image frame. Such invalid coordinates are then discarded, retaining only the valid set as the seed keypoints. Each seed keypoint forms a circle of radius ρ pixels around it. Within these circles, keypoints are sampled uniformly, while areas outside the circles are excluded from sampling. This targeted sampling technique allows the network to concentrate on semantically relevant portions of the image, such as architectural features, rather than non-distinctive, texture-less zones (e.g., sky). We refer the reader to Fig. 4 for a comparison of both sampling strategies.

5. Experimental Setup and Results

This section extensively evaluates the proposed Focus-Tune sampling method. We first provide details on the implementation in Section 5.1, followed by an overview of the evaluation datasets in Section 5.2. This is followed by a rigorous validation of FocusTune's effectiveness using two diverse datasets: the 7-scenes dataset for small-scale indoor environments and the Cambridge Landmarks Dataset for large-scale outdoor settings (Section 5.3). We then assess the sensitivity of our method to variations in the sampling radius in Section 5.4. Finally, Section 5.5 provides additional insights into the focus-guided sampling strategy. Overall, our findings demonstrate the significant localization accuracy of FocusTune over existing methods while retaining the computational efficiency of ACE models [3].

5.1. Implementation

We incorporate our FocusTune sampling approach into the ACE learning pipeline as described in Brachmann et al. [3]. Our training buffer consists of 8 million instances. Keeping consistent with ACE's original settings, we optimise f_H over 16 complete iterations using a batch size of 5120. For comprehensive details on other hyper-parameters, the reader is referred to [3]. Unless otherwise mentioned, we set the sampling radius $\rho = 5$ which is the only hyperparameter of our method.

5.2. Datasets

FocusTune requires ground-truth 3D models for training; therefore, our evaluations are based on two datasets that provide such ground-truth information:

Figure 4. **Sampling strategies comparison.** *(Left image)* An example of the sampled pixels (red dots) using random sampling. *(Right image)* An example of the sampled pixels (green dots) using our sampler. Our sampler focuses on important building structures instead of unimportant, texture-less regions (e.g., the sky) of the image.

		Mapping w/ Mesh/Depth	Mapping Time	Map Size	7 Scenes							
					Pumpkin	Redkitchen	Stairs	Office	Heads	Chess	Fire	Average
Feature Matching	AS (SIFT) [44]	No		∼200MB	99.6%	99.8%	91.9%	98.6%	100%	99.9%	99.8%	98.5%
	D.VLAD+R2D2 [24]	No	∼1.5h	∼1GB	98.8%	98.4%	76.9%	99.7%	97.0%	100%	100%	95.7%
	hLoc (SP+SG) [40,41]	No		∼2GB	100%	98.6%	72.0%	100%	100%	100 %	99.4%	95.7%
SCR (w/Depth)	DSAC* [8]	Yes	15h	28MB	90.9%	96.4%	88.4%	95.3%	99.5%	99.6%	96.6%	95.3%
	SRC [20]	Yes	2 min	40MB	-	-	-	-	-	-	-	81.1%
SCR	DSAC* [8]	No	15h	28MB	99.0%	97.0%	**92.0%**	98.1%	99.8%	99.9%	98.9%	<u>97.8%</u>
	ACE [3]	No	5 min	4MB	**99.9%**	<u>98.5%</u>	83.1%	**100%**	**99.9%**	**100%**	<u>99.4%</u>	97.2%
	FocusTune (ours)	No	6 min	4MB	<u>99.7%</u>	**99.0%**	<u>87.1%</u>	<u>99.9%</u>	**99.9%**	**100%**	**99.5%**	**97.9%**

Table 1. **Indoor relocalization on 7-scenes [48].** We report the percentage of test frames below a 5 cm and 5 degrees pose error (higher is better). Best results are shown in **bold** for the Scene Coordinate Regression ("SCR") group, second best results are underlined. We improve upon the performance of ACE [3], especially in the difficult Stairs scene by an absolute 4.0% improvement, and achieve slightly higher performance than DSAC* [8] on average. The results for baseline methods were obtained from [3, 4].

	ACE [3] °/cm	FocusTune (ours) °/cm
Pumpkin	0.21 / 1.06	**0.20 / 1.00**
Redkitchen	0.20 / 0.77	**0.18 / 0.73**
Stairs	0.82 / 2.81	**0.76 / 2.60**
Office	0.28 / 1.04	**0.24 / 0.97**
Heads	0.33 / 0.53	**0.29 / 0.50**
Chess	0.18 / 0.54	**0.17 / 0.50**
Fire	0.32 / 0.82	**0.30 / 0.77**
Average	0.33 / 1.08	**0.30 / 1.01**

Table 2. **7-scenes dataset [48].** We report the median rotation (in degrees) and translation error (in cm) for the ACE baseline method [3] and our proposed FocusTune (lower is better). Best results are shown in **bold**. We consistently outperform ACE in all scenes, on average by 0.03 degrees and 0.07 cm.

The **7-scenes dataset** [48] provides ground-truth camera poses for small-scale indoor scenes using a Kinect RGB-D camera with 640×480 resolution. Even though the depth information is provided for each image, we only use RGB frames to train f_H. Seven sequences were captured for each scene by different users and then were split into separate sets for training and testing purposes. Ambiguities, such as repeated steps in the Stairs scene, as well as specularities like reflective cupboards in the RedKitchen scene, are present in both RGB and depth images. Other challenges include motion blur, different lighting conditions, flat surfaces, and sensor noise. The ground-truth camera poses and 3D point cloud for each sequence were obtained using COLMAP [46] and provided by [4]. With the sampling radius $\rho = 5$, our sampling method makes use of approx. 15.4% of the pixels in the training images on average.

The **Cambridge Landmarks Dataset** [27] was collected at 5 different sites at the University of Cambridge in a large-scale outdoor urban setting with a Google LG Nexus 5 smartphone into high-definition videos. These videos were subsampled at a 2 Hz rate to obtain RGB frames. Substantial urban obstacles like pedestrians and vehicles were evident, with data being gathered across various timeframes that reflected diverse lighting and weather situations. Training and testing sets came from separate walking trajectories rather than being selected from a single trajectory, thereby rendering it more challenging for the localization task. The ground-truth training and testing camera poses as well as the 3D models were reconstructed using VisualSfM [53] and made available with the dataset. With sampling radius

		Mapping w/ Mesh/Depth	Mapping Time	Map Size	Cambridge Landmarks					Average (cm / °)
					Court	King's	Hospital	Shop	St. Mary's	
Feature Matching	AS (SIFT) [44]	No		~200MB	24/0.1	13/0.2	20/0.4	4/0.2	8/0.3	14/0.2
	hLoc (SP+SG) [40, 41]	No		~800MB	16/0.1	12/0.2	15/0.3	4/0.2	7/0.2	11/0.2
	pixLoc [42]	No	~35min	~600MB	30/0.1	14/0.2	16/0.3	5/0.2	10/0.3	15/0.2
	GoMatch [55]	No		~12MB	N/A	25/0.6	283/8.1	48/4.8	335/9.9	N/A
	HybridSC [11]	No		~1MB	N/A	81/0.6	75/1.0	19/0.5	50/0.5	N/A
APR	PoseNet17 [26]	No	4 – 24h	50MB	683/3.5	88/1.0	320/3.3	88/3.8	157/3.3	267/3.0
	MS-Transformer [47]	No	~7h	~18MB	N/A	83/1.5	181/2.4	86/3.1	162/4.0	N/A
SCR w/Depth	DSAC* [8]	Yes	15h	28MB	49/0.3	15/0.3	21/0.4	5/0.3	13/0.4	21/0.3
	SANet [54]	Yes	~1min	~260MB	328/2.0	32/0.5	32/0.5	10/0.5	16/0.6	84/0.8
	SRC [20]	Yes	2 min	40MB	81/0.5	39/0.7	38/0.5	19/1.0	31/1.0	42/0.7
SCR	DSAC* [8]	No	15h	28MB	34/0.2	_18/0.3_	21/0.4	_5/0.3_	15/0.6	19/0.4
	ACE [3]	No	5 min	4MB	43/0.2	28/0.4	31/0.6	_5/0.3_	18/0.6	25/0.4
	FocusTune (ours)	No	6 min	4MB	38/0.1	19/0.3	18/0.4	6/0.3	15/0.5	19/0.3
	ACE (4 model ensemble) [3]	No	20 min	16MB	**28/0.1**	_18/0.3_	25/0.5	_5/0.3_	**9/0.3**	_17/0.3_
	FocusTune (4 model ensemble; ours)	No	24 min	16MB	_29/0.1_	**15/0.3**	**17/0.4**	**5/0.2**	**9/0.3**	**15/0.3**

Table 3. **Cambridge Landmarks [27] Results.** We report median rotation (in degrees) and position errors (in cm). Best results for the Scene Coordinate Regression (SCR) methods are shown in **bold**, and second best results are underlined. Our proposed FocusTune outperforms the direct competitor ACE in all but one scenes, both for a single model and a 4 model ensemble. The 4 model ensemble of FocusTune also consistently outperforms DSAC* which has a significantly higher mapping time, closing the gap to structured methods that employ feature matching. The results for baseline methods were obtained from [3].

$\rho = 5$, our sampling method makes use of approximately 16% of the pixels in the training images on average.

5.3. Comparison to State-of-the-Art

5.3.1 7-scenes Dataset

We now compare FocusTune to the state-of-the-art on the 7-scenes indoor dataset. Employing FocusTune resulted in models that consistently outperform the ACE baseline [3] across all tested indoor scenes, highlighted by either higher percentages of test images localized within $5°$ and 5 cm of the ground truth poses (Table 1) or lower median errors when the percentages are comparable (Table 2). Notably, in texture-deficient scenes such as the Stairs sequence, we observed an improvement of approximately 4% over ACE. However, for easier scenes, our sampling method also leads to even more accurate localizers with lower median errors across all test sequences (Table 2).

5.3.2 Cambridge Landmarks Dataset

As shown in Table 3, compared to ACE [3], FocusTune excelled on the Cambridge landmarks dataset with a significant median error reduction of $0.1°$ in rotation and 6 cm in translation for a single model. The improvement was even more pronounced in challenging scenes like the King (translation error reduction of 9 cm) and Hospital (translation error reduction of 13 cm) sequences.

Similar observations hold for the 4 model ensemble variants, where FocusTune improved upon ACE's performance by on average 2 cm. In scenes like the King and Hospital

sequences, our method again reduces the translation errors from 18 cm to 15 cm and 25 cm to 17 cm, respectively. FocusTune thus closes the gap further to structured methods relying on feature matching [40–42, 44], while retaining the significantly low storage requirements of ACE [3], and performs better than DSAC* [8].

5.4. Sampling Radius

In this section, we assess the sensitivity of our method to variations in the sampling radius ρ. The bottom plot in Fig. 5 illustrates the median translation error for the sequences from the Cambridge Landmarks dataset [27].

To determine the operational radius for our experiments, we aggregate the results from the five sequences using the following approach: we individually normalize the results of each sequence with its corresponding minimum error. For each radius value, we calculate the mean of the normalized errors across the different sequences and add one standard deviation. The upper plot in Fig. 5 presents the resulting normalized error, which combines the outcomes from all five sequences.

For the remainder of our experiments, we select a sampling radius of $\rho = 5$, which corresponds to the point where our approach demonstrates optimal performance with the lowest error. We observe that larger values result in decreased performance because the buffer becomes populated with keypoints from non-textured areas. Note that $\rho \to \infty$ falls back to the default ACE random sampling. Conversely, smaller values lead to reduced performance due to a decrease in the number of keypoints.
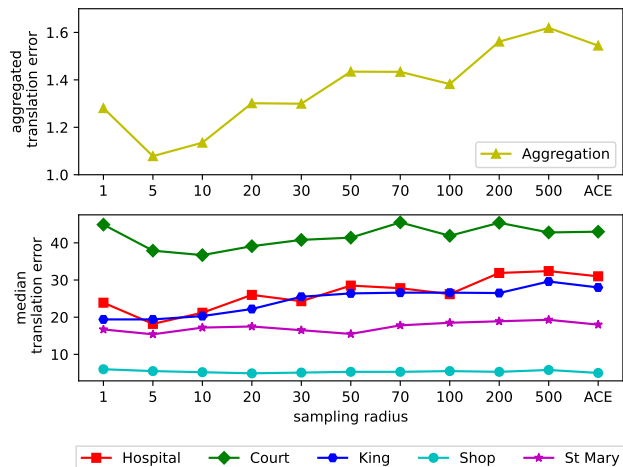
Figure 5. **Ablation studies on** ρ. We plot the normalized aggregated (top) and absolute individual (bottom) median translation errors for our method, using various sampling radii ρ, across the five sequences of the Cambridge Landmarks dataset [27]. A sampling radius of $\rho = 5$ leads to highest performance, with smaller values degrading in performance as the number of keypoints reduces too much. Larger sampling radii include areas that are not informative, and similarly result in reduced performance.

5.5. Sampling Strategy Analysis

To elucidate the efficiency of our sampling method, we compared two training buffers: $\mathbf{B}_{ACE}$ using the random sampling strategy employed by ACE and $\mathbf{B}_{FocusTune}$ using our sampling strategy. Both mean and median re-projection errors dropped markedly across all scenes when employing our sampling strategy, suggesting more effective dataset utilization (Fig. 6).

6. Conclusions and Future Works

In this paper, we introduced FocusTune, a novel focus-guided sampling strategy designed to improve the performance of visual localization algorithms. By leveraging key geometric constraints to guide the scene coordinate regressor, FocusTune advances the state-of-the-art in visual localization accuracy while reducing computational and storage requirements. Notably, we demonstrated these benefits by integrating FocusTune into the recently developed Accelerated Coordinate Encoding (ACE) model, achieving reductions in translation errors on indoor and outdoor benchmark datasets.

Even though SfM maps often come with ground-truth camera poses in common capturing processes [46, 53] and SfM models are highly useful for downstream applications (e.g., augmented reality), access requirements for these models might render our method inapplicable for scenarios in which the mapping poses are captured via depth SLAM [17]. However, our experiments show that SCR
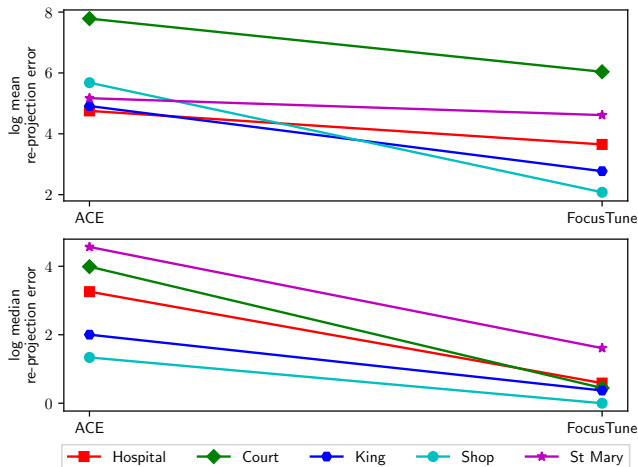

Figure 6. **Training efficiency.** We sample two buffers from the Cambridge Landmarks dataset [27]: $\mathbf{B}_{ACE}$ using random sampling as in ACE [3] (*left*) and $\mathbf{B}_{FocusTune}$ using our proposed focus-guided sampling strategy (*right*). We compute the mean and median re-projection errors of all the instances after training and plot the log of the errors. The decrease in training re-projection errors suggests that f_H is able to triangulate more points from $\mathbf{B}_{FocusTune}$ than $\mathbf{B}_{ACE}$, highlighting a better use of the dataset.

models benefit from training with salient image regions that can still be discovered using an off-the-shelf feature detector [2, 18, 21, 35, 37].

We suggest a range of promising directions to investigate for future works. Exploring adaptive sampling techniques could provide an increased level of adaptability to varying scene complexities. Whilst FocusTune is highly performant in itself, by integrating FocusTune's heuristic sampling strategy with other localization methods it may be possible to obtain further performance improvements, where this is practically acceptable. Finally, the exploration of geometrically-informed loss functions could further optimize the triangulation process, leading to more robust performance across a wider range of scenarios.

We believe that FocusTune represents a significant advancement in the field, offering both high performance and efficiency, thus making it particularly appealing for applications in mobile robotics and augmented reality.

Acknowledgements

References

[1] Clemens Arth, Manfred Klopschitz, Gerhard Reitmayr, and Dieter Schmalstieg. Real-time self-localization from

panoramic images on mobile devices. In *IEEE International Symposium on Mixed and Augmented Reality*, pages 37–46, 2011. 1

[2] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: speeded up robust features. In *Eur. Conf. Comput. Vis.*, pages 404–417, 2006. 2, 4, 8

[3] Eric Brachmann, Tommaso Cavallari, and Victor Adrian Prisacariu. Accelerated coordinate encoding: Learning to relocalize in minutes using rgb and poses. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5044–5053, 2023. 1, 2, 3, 4, 5, 6, 7, 8

[4] Eric Brachmann, Martin Humenberger, Carsten Rother, and Torsten Sattler. On the limits of pseudo ground truth in visual camera re-localisation. In *IEEE Int. Conf. Comput. Vis.*, pages 6198–6208, 2021. 6

[5] Eric Brachmann, Frank Michel, Alexander Krull, Michael Ying Yang, Stefan Gumhold, et al. Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3364–3372, 2016. 1, 3

[6] Eric Brachmann and Carsten Rother. Learning less is more-6d camera localization via 3d surface regression. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4654–4662, 2018. 3

[7] Eric Brachmann and Carsten Rother. Expert sample consensus applied to camera re-localization. In *IEEE Int. Conf. Comput. Vis.*, pages 7525–7534, 2019. 1, 3

[8] Eric Brachmann and Carsten Rother. Visual camera re-localization from RGB and RGB-D images using DSAC. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(9):5847–5865, 2022. 1, 3, 6, 7

[9] Samarth Brahmbhatt, Jinwei Gu, Kihwan Kim, James Hays, and Jan Kautz. Geometry-aware learning of maps for camera localization. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2616–2625, 2018. 1

[10] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Trans. Robot.*, 32(6):1309–1332, 2016. 2

[11] Federico Camposeco, Andrea Cohen, Marc Pollefeys, and Torsten Sattler. Hybrid scene compression for visual localization. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7653–7662, 2019. 7

[12] Robert Castle, Georg Klein, and David W Murray. Video-rate localization in multiple maps for wearable augmented reality. In *IEEE International Symposium on Wearable Computers*, pages 15–22, 2008. 1

[13] Tommaso Cavallari, Luca Bertinetto, Jishnu Mukhoti, Philip H. S. Torr, and Stuart Golodetz. Let's take this online: Adapting scene coordinate regression network predictions for online RGB-D camera relocalisation. In *Int. Conf. 3D Vision*, pages 564–573, 2019. 1

[14] Tommaso Cavallari, Stuart Golodetz, Nicholas A Lord, Julien Valentin, Luigi Di Stefano, and Philip HS Torr. On-the-fly adaptation of regression forests for online camera relocalisation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4457–4466, 2017. 3

[15] Tommaso Cavallari, Stuart Golodetz, Nicholas A Lord, Julien Valentin, Victor A Prisacariu, Luigi Di Stefano, and Philip HS Torr. Real-time rgb-d camera pose estimation in novel scenes using a relocalisation cascade. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(10):2465–2477, 2019. 3

[16] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5828–5839, 2017. 3, 4

[17] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Trans. Graph.*, 36(4):1, 2017. 8

[18] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, pages 224–236, 2018. 2, 3, 4, 8

[19] Tien Do, Ondrej Miksik, Joseph DeGol, Hyun Soo Park, and Sudipta N Sinha. Learning to detect scene landmarks for camera localization. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 11132–11142, 2022. 1, 3

[20] Siyan Dong, Shuzhe Wang, Yixin Zhuang, Juho Kannala, Marc Pollefeys, and Baoquan Chen. Visual localization via few-shot scene region classification. In *Int. Conf. 3D Vision*, pages 393–402, 2022. 3, 6, 7

[21] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable CNN for joint description and detection of local features. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8092–8101, 2019. 2, 4, 8

[22] Lionel Heng, Benjamin Choi, Zhaopeng Cui, Marcel Geppert, Sixing Hu, Benson Kuan, Peidong Liu, Rang Nguyen, Ye Chuan Yeo, Andreas Geiger, et al. Project autovision: Localization and 3d scene perception for an autonomous vehicle with a multi-camera system. In *IEEE Int. Conf. Robot. Autom.*, pages 4695–4702, 2019. 1

[23] Zhaoyang Huang, Han Zhou, Yijin Li, Bangbang Yang, Yan Xu, Xiaowei Zhou, Hujun Bao, Guofeng Zhang, and Hongsheng Li. Vs-net: Voting with segmentation for visual localization. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6101–6111, 2021. 1, 3

[24] Martin Humenberger, Yohann Cabon, Nicolas Guerin, Julien Morat, Jérôme Revaud, Philippe Rerole, Noé Pion, Cesar de Souza, Vincent Leroy, and Gabriela Csurka. Robust image retrieval-based visual localization using Kapture, 2020. 6

[25] Arnold Irschara, Christopher Zach, Jan-Michael Frahm, and Horst Bischof. From structure-from-motion point clouds to fast location recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2599–2606, 2009. 3

[26] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5974–5983, 2017. 1, 7

[27] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *IEEE Int. Conf. Comput. Vis.*, pages 2938–2946, 2015. 1, 3, 4, 6, 7, 8

[28] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.04643*, 2023. 4

[29] Viktor Larsson. PoseLib - Minimal Solvers for Camera Pose Estimation, 2020. 4

[30] Boyi Li, Kilian Q. Weinberger, Serge J. Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. In *Int. Conf. Learn. Represent.*, 2022. 4

[31] Xinyi Li and Haibin Ling. GTCaR: graph transformer for camera re-localization. In *Eur. Conf. Comput. Vis.*, pages 229–246, 2022. 1, 3

[32] Xiaotian Li, Shuzhe Wang, Yi Zhao, Jakob Verbeek, and Juho Kannala. Hierarchical scene coordinate classification and regression for visual localization. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 11983–11992, 2020. 1, 3

[33] Yunpeng Li, Noah Snavely, and Daniel P. Huttenlocher. Location recognition using prioritized feature matching. In *Eur. Conf. Comput. Vis.*, pages 791–804, 2010. 1, 3

[34] Hyon Lim, Sudipta N Sinha, Michael F Cohen, and Matthew Uyttendaele. Real-time image-based 6-dof localization in large-scale environments. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1043–1050, 2012. 1

[35] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.*, 60(2):91–110, 2004. 2, 3, 4, 8

[36] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *Int. J. Comput. Vis.*, 60:63–86, 2004. 2

[37] Anastasiia Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *Adv. Neural Inform. Process. Syst.*, 2017. 4, 8

[38] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *IEEE Int. Conf. Comput. Vis.*, pages 3456–3465, 2017. 2

[39] Mikael Persson and Klas Nordberg. Lambda twist: An accurate fast robust perspective three point (P3P) solver. In *Eur. Conf. Comput. Vis.*, pages 318–332, 2018. 4

[40] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 12716–12725, 2019. 6, 7

[41] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4938–4947, 2020. 6, 7

[42] Paul-Edouard Sarlin, Ajaykumar Unagar, Mans Larsson, Hugo Germain, Carl Toft, Viktor Larsson, Marc Pollefeys, Vincent Lepetit, Lars Hammarstrand, Fredrik Kahl, et al. Back to the feature: Learning robust camera localization from pixels to pose. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3247–3257, 2021. 7

[43] Torsten Sattler, Michal Havlena, Filip Radenovic, Konrad Schindler, and Marc Pollefeys. Hyperpoints and fine vocabularies for large-scale location recognition. In *IEEE Int. Conf. Comput. Vis.*, pages 2102–2110, 2015. 1, 3

[44] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Improving image-based localization by active correspondence search. In *Eur. Conf. Comput. Vis.*, volume 7572, pages 752–765, 2012. 1, 3, 6, 7

[45] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(9):1744–1756, 2016. 3

[46] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4104–4113, 2016. 2, 3, 6, 8

[47] Yoli Shavit, Ron Ferens, and Yosi Keller. Learning multi-scene absolute pose regression with transformers. In *IEEE Int. Conf. Comput. Vis.*, pages 2713–2722, 2021. 1, 3, 7

[48] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew W. Fitzgibbon. Scene coordinate regression forests for camera relocalization in RGB-D images. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2930–2937, 2013. 1, 3, 4, 6

[49] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1–9, 2015. 3

[50] Shitao Tang, Sicong Tang, Andrea Tagliasacchi, Ping Tan, and Yasutaka Furukawa. Neumap: Neural coordinate mapping by auto-transdecoder for camera localization. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 929–939, 2023. 3

[51] Julien P. C. Valentin, Matthias Nießner, Jamie Shotton, Andrew W. Fitzgibbon, Shahram Izadi, and Philip H. S. Torr. Exploiting uncertainty in regression forests for accurate camera relocalization. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4400–4408, 2015. 3

[52] Florian Walch, Caner Hazirbas, Laura Leal-Taixe, Torsten Sattler, Sebastian Hilsenbeck, and Daniel Cremers. Image-based localization using lstms for structured feature correlation. In *IEEE Int. Conf. Comput. Vis.*, pages 627–637, 2017. 1, 3

[53] Changchang Wu. Towards linear-time incremental structure from motion. In *Int. Conf. 3D Vision*, pages 127–134, 2013. 3, 6, 8

[54] Luwei Yang, Ziqian Bai, Chengzhou Tang, Honghua Li, Yasutaka Furukawa, and Ping Tan. Sanet: Scene agnostic network for camera localization. In *IEEE Int. Conf. Comput. Vis.*, pages 42–51, 2019. 3, 7

[55] Qunjie Zhou, Sérgio Agostinho, Aljoša Ošep, and Laura Leal-Taixé. Is geometry enough for matching in visual localization? In *Eur. Conf. Comput. Vis.*, pages 407–425, 2022. 1, 3, 7