A2-GNN: Angle-Annular GNN for Visual Descriptor-free Camera Relocalization

Yejun Zhang[1] Shuzhe Wang[1] Juho Kannala[1,2]
[1]Aalto University [2]University of Oulu
firstname.lastname@aalto.fi

Abstract

Visual localization involves estimating the 6-degree-of-freedom (6-DoF) camera pose within a known scene. A critical step in this process is identifying pixel-to-point correspondences between 2D query images and 3D models. Most advanced approaches currently rely on extensive visual descriptors to establish these correspondences, facing challenges in storage, privacy issues and model maintenance. Direct 2D-3D keypoint matching without visual descriptors is becoming popular as it can overcome those challenges. However, existing descriptor-free methods suffer from low accuracy or heavy computation. Addressing this gap, this paper introduces the Angle-Annular Graph Neural Network (A2-GNN), a simple approach that efficiently learns robust geometric structural representations with annular feature extraction. Specifically, this approach clusters neighbors and embeds each group's distance information and angle as supplementary information to capture local structures. Evaluation on matching and visual localization datasets demonstrates that our approach achieves state-of-the-art accuracy with low computational overhead among visual description-free methods.

1. Introduction

Visual localization aims to estimate the camera pose of a query image within a pre-built 3D environment. It is essential in computer vision applications such as Structure-from-Motion (SfM) [55, 61, 72], Augmented Reality (AR) [3, 10], and Simultaneous Localization and Mapping (SLAM) [6, 21, 40] . Classical visual localization approaches [47, 49, 50, 53, 68, 70] use visual descriptors to build the 2D-3D correspondences. Once the correspondences are established, a PnP-RANSAC solver [78] is applied to estimate the 6-DoF camera pose.

Among all camera pose estimation approaches, visual descriptor-based methods [47, 48] achieve state-of-the-art (SOTA) performance. However, utilizing visual descriptors presents several challenges, including substantial storage requirements, privacy risks, and maintenance complexity [11, 41, 69, 77]. Storing high-dimensional descriptors for each keypoint typically demands significant storage capacity. Additionally, studies [11, 41] have demonstrated that visual descriptors can be exploited to recover images, raising privacy concerns. Moreover, maintaining the 3D model becomes complicated when integrating new descriptors or points into existing point clouds [18].

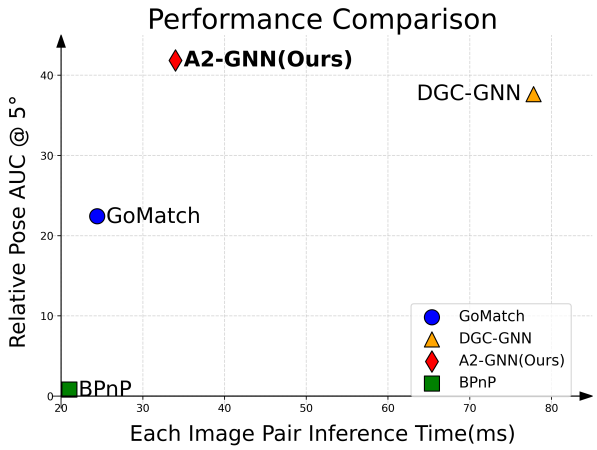

Figure 1. Matching Accuracy and Efficiency Comparisons for descriptor-free methods. Compared with GoMatch [77] and DGC-GNN [69], our A2-GNN learns effective and accurate 2D-3D matching.

To address the challenges associated with visual descriptor-based methods, several researchers have developed visual descriptor-free approaches [7, 69, 77]. These methods establish 2D-3D correspondences without relying on visual descriptors, instead leveraging geometric information from keypoints for matching. While these approaches demonstrate reasonable performance, their primary limitation lies in the inherently limited geometric information, which lacks the richness and detail provided by visual descriptors. GoMatch [77] attempts to learn the geometric-only representation by utilizing graph neural networks and incorporating outlier rejection, achieving reasonable performance. However, there is still a noticeable performance gap compared to descriptor-based methods. DGC-GNN [69]

narrows this gap by integrating color and more geometric clues, along with global-to-local clustering. Despite these improvements, DGC-GNN suffers from heavy computational demands, particularly due to the multiple clustering operations required.

To achieve accurate camera pose estimation with light computation, we reconsider the way of encoding geometric information. In previous descriptor-free approaches [69, 77], transformers [67] have been employed, utilizing self-attention and cross-attention mechanisms to enhance feature representation. The self-attention mechanism captures local geometric structures within neighborhoods, while cross-attention facilitates the exchange of geometry information between 2D keypoints and 3D point clouds. The extraction of local geometry through self-attention is particularly crucial, as learning high-quality feature representations is essential for effective information exchange in the cross-attention layer. Previous methods [69, 77] utilize max-pooling to extract local geometry from neighbors. However, this operation neglects neighbor structural information as most of neighbor information are discarded. Inspired by CLNet [76], we propose **A**ngle-**A**nnular **G**raph **N**eural **N**etwork, or **A2-GNN**, which efficiently extracts local structural geometric information for 2D-3D matching without relying on visual descriptors. Our network processes sparse points from query images and point clouds as inputs, delivering accurate correspondences with minimal computational overhead. We first construct a local graph for each point by connecting it to its neighboring points based on their distances. These neighboring points are then clustered into close, middle, and remote groups. Local structural information is extracted by processing these grouped neighbors separately. Besides, angular information between the point and its neighbors is encoded similarly to further enhance the local geometric representation. By encoding these geometric cues, our method effectively distinguishes between structurally similar yet distinct points in both 2D and 3D spaces. In addition, we also adjust outlier rejection input to position information rather than the feature representation, as explicit position information can provide consensus in epipolar geometry to remove outliers. In summary, our paper makes the following contributions:

- We introduce a novel local graph neighbor aggregation method for direct 2D-3D matching without visual descriptors. This method embeds geometric information in a hybrid manner and improves the accuracy of spare 2D-3D matching.
- We adjust outlier rejection input to position information, as pure position information can provide robust geometry constraint.
- Our method outperforms previous descriptor-free methods in matching and visual localization tasks with considerable efficiency.

2. Related work

Structure-Based Localization. Structure-based visual localization has achieved high accuracy by leveraging a prebuilt 3D map of the environment, which is crucial for establishing correspondences between 2D image pixels and 3D point clouds. Classical methods [49, 50] relied on Structure-from-Motion (SfM) models to match features between query images and 3D points. In contrast, recent approaches have incorporated learning-based techniques, significantly enhancing image retrieval [2, 23, 24, 45], feature extraction [14, 17, 46, 65], and keypoint matching [1, 38, 48] in the structure-based localization pipeline.

Image retrieval is a vital component of structure-based localization, as it identifies similar or overlapping images from a large database to narrow down the search space of the 3D points. NetVLAD [2] integrates a VLAD-like layer in CNNs and optimizes local descriptor extraction and aggregation, enhancing image retrieval accuracy. Recent works [27, 31] leverage powerful pretrained models to further improve retrieval accuracy. Feature extraction methods [14, 17, 46] extract robust visual representations from raw images, enabling more reliable keypoint detection and description. The integration of neural networks has further enhanced keypoint matching accuracy, particularly with advanced methods like SuperGlue [48], which employs graph neural networks to refine the matching process. Recent advancements [29, 38, 57] continue to push the boundaries of accuracy and efficiency in this domain, making structure-based visual localization increasingly robust and scalable for real-world applications. In addition, detector-free methods [12, 64, 71] input images directly for matching and achieve significant improvements in accuracy but encounter challenges in heavy computation.

Descriptor-free Visual Localization. BPnPNet [7] makes significant progress in 2D pixel to 3D point cloud matching without relying on visual descriptors, introducing an end-to-end trainable matcher for cross-modal point matching. It also introduces the use of bearing vectors as a 2D keypoint representation, effectively bridging the 2D-3D gap and performing well in outlier-free scenarios. GoMatch [77] develops this concept, utilizing bearing vectors to represent both 2D and 3D keypoints for geometric-only matching. It employs SuperGlue-style [48] self- and cross-attention mechanisms to establish initial correspondences and integrates PointCN [74] to remove outliers from the initial matches. DGC-GNN [69] further improves matching accuracy by leveraging additional RGB information, multiple geometric cues, and a global-to-local feature embedding strategy. It clusters points to create a global geometric graph, enhancing feature matching across clusters by encoding Euclidean and angular relations. However, this approach comes with a trade-off: its inference time is three times longer than that of GoMatch due to the iterative point clustering process.

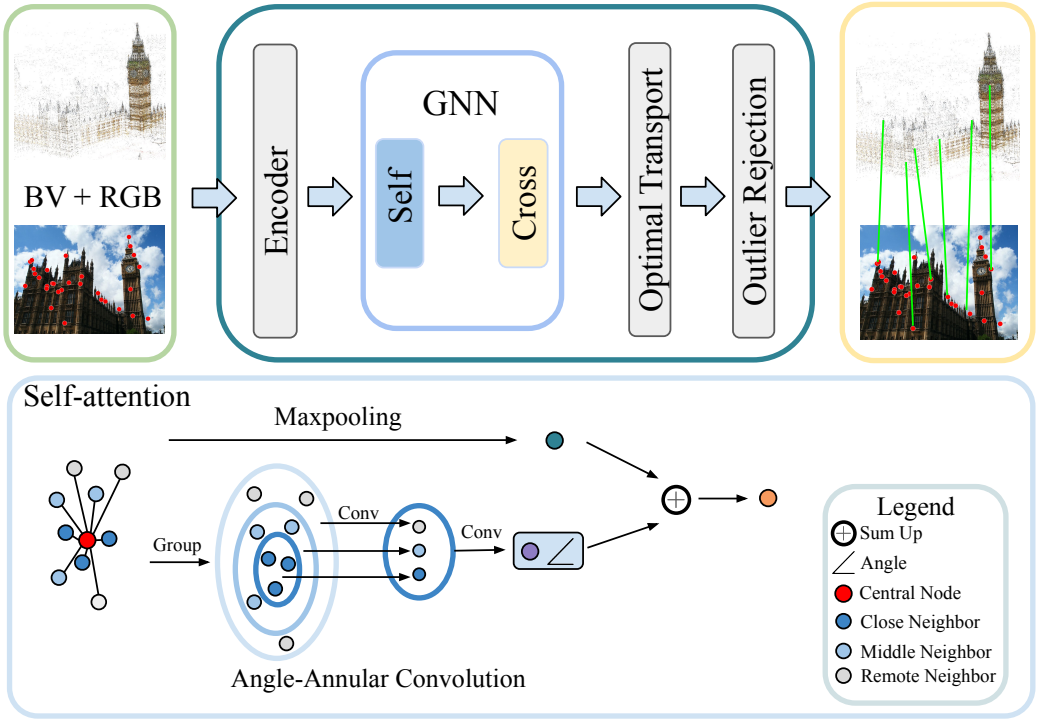

Figure 2. Architecture Overview. The bearing vector (BV) and RGB information from the query image and 3D point cloud are first processed through an encoder to generate high-dimensional features. These features are then used to construct the graph nodes. In the self-attention layer, the angle-annular convolution is employed to extract discriminative geometric information from the neighboring points. After the GNNs, these enhanced features are used to establish initial correspondences via optimal transport. Outlier rejection is then applied to eliminate erroneous correspondences, resulting in a final set of accurate correspondences.

Scene Compression. Scene compression for visual localization involves two primary approaches: map compression and descriptor compression. Map compression [8, 9, 19, 34, 36, 73] reduces map size by pruning indistinguishable points, while descriptor compression [16, 28, 30, 34, 51, 73] minimizes descriptor size while maintaining their essential properties for accurate matching. Scene coordinate regression [4, 5, 68, 70] uses neural networks to compress scene models and learns a compact representation. It predicts scene coordinates directly from 2D images without visual descriptors, but faces challenges in generalizing to new scenes. Hybrid methods [8, 34, 73] combine both map and descriptor compression to achieve more compact representations. However, while scene compression techniques can save storage space, they do not fully address privacy concerns or the complexities associated with descriptor maintenance. Our method, which avoids the use of visual descriptors altogether, can also be integrated with scene compression techniques to further reduce the model size.

Privacy-preserving visual localization. As demonstrated in [43], image details can be recovered from descriptors. This raises significant data privacy concerns, as descriptors can be leaked during transmission across devices. To address this concern, privacy-preserving visual localization methods [39, 41, 58, 62, 63] have been developed. These methods represented 2D/3D points as lines or other geometric representations to obscure spatial details. However, they suffer from accuracy degradation, high storage requirements, and maintenance complexity.

Other privacy-preserving methods [15, 42] take alternative strategies. SLD [15] identifies distinctive points as scene landmarks for camera localization. SegLoc [42] utilizes semantic segmentation to create feature representations optimized for privacy and localization performance. Nevertheless, these methods rely on extensive priors or labels and limit generalizability in other scenarios.

3. Method

3.1. Problem setting

Let $p \in \mathbb{R}^2$ denote a 2D point and $q \in \mathbb{R}^3$ denote a 3D point. Assume that a 2D image P contains M keypoints, and a 3D point cloud Q contains N keypoints. We denote the sets of keypoints in 2D and 3D cases as $\mathcal{P} = \{p_i \mid i = 1, \dots, M\}$ and $\mathcal{Q} = \{q_j \mid j = 1, \dots, N\}$, respectively. Our task is to predict a set of correspondences $\mathcal{M}_{p,q}$ between 2D

keypoints and 3D point clouds.

Keypoint Representation. Following GoMatch [77], we utilize bearing vectors as keypoint representation, as it can bring 2D keypoints and 3D point clouds into the same modality. For 2D keypoints, bearing vectors $\mathbf{b_p}$ remove the effect of camera intrinsic $\mathbf{K}$ by the following equation:

$$[\mathbf{b_p}^\top, 1]^\top = \mathbf{K}^{-1}[u, v, 1]^\top, \tag{1}$$

where (u, v) represents 2D keypoint pixel coordinates and $\mathbf{b_p} \in \mathbb{R}^2$. The bearing vector brings pixel coordinates into the corresponding camera ray by connecting the camera center and pixel coordinates. For 3D points, the bearing vector $\mathbf{b_q}$ transforms its world coordinates $\mathbf{p^w}$ to the camera ray by the following equation:

$$\mathbf{p^c} = \mathbf{R}\mathbf{p^w} + \mathbf{t} \tag{2}$$

$$[\mathbf{b_q}^\top, 1]^\top = \mathbf{p^c}/p_z^c, \tag{3}$$

where $\mathbf{p^c} \in \mathbb{R}^3$ is its the camera coordinate, p_z' is its z coordinate and $\mathbf{b^q} \in \mathbb{R}^2$. The bearing vector first transforms 3D points from the world coordinates to the camera coordinates. Similar to the image plane, we take plane $z = 1$ and connect the camera center to its camera coordinates. The camera ray is obtained by connecting the camera center and the point where the line intersects the plane.

3.2. Network Architecture

We give an overview of the proposed Angle-Annular convolution Graph Neural Network, shortened as A2-GNN in Fig. 2. The architecture includes the following modules: feature encoder, angle-annular geometric feature extraction, optimal transport, and outlier rejection. The encoder transforms positional information and RGB color from low-dimensional inputs into high-dimensional features. These features are then processed by A2-GNN to extract local geometric relations from neighboring points. Initial correspondences are established using optimal transport, and low-confidence matches are subsequently removed by the outlier rejection module.

3.2.1 Feature Encoder

Feature Encoder. We use a ResNet-style encoder [7, 25, 77] to extract both position and color features directly from 2D kepoints or 3D point clouds, denoted as $\mathcal{F}_b$ and $\mathcal{F}_c$. The feature encoder encodes the bearing vector and RGB color separately, from low dimension vectors to high-dimensional (e.g., d=128) vectors. The point features $\mathbf{f}_p \in \mathbb{R}^{N\times d}$ from query images and $\mathbf{f}_q \in \mathbb{R}^{M\times d}$ from point clouds can be computed as:

$$\mathbf{f_p} = \mathcal{F}_b(\mathbf{b_p}) + \mathcal{F}_c(\mathbf{c_p}) \tag{4a}$$

$$\mathbf{f_q} = \mathcal{F}_b(\mathbf{b_q}) + \mathcal{F}_c(\mathbf{c_q}), \tag{4b}$$

where $\mathbf{c_p} \in \mathbb{R}^3, \mathbf{c_q} \in \mathbb{R}^3$ are RGB color of points from the query images and point clouds, respectively.

3.2.2 A2-GNN

Graph Neural Networks enhance feature representation by aggregating information from neighboring nodes, which is utilized for both images and point clouds. This section introduces the Angle-Annular Convolution Aggregation method, self-attention and cross-attention mechanisms.

Geometric Local Feature. After feature extraction, we construct local graphs for each keypoint based on its neighboring points. We consider the local graphs as a self-attention mechanism to extract local geometric features, thereby enhancing the representation and capturing the contextual relationships among the keypoints. Following previous works [69, 77], edges $\mathcal{E}$ are constructed in Euclidean space by connecting each node to its k nearest neighbors. The edge feature $\mathbf{e}_{ij} \in \mathbb{R}^{k\times 2d}$ between a node $\mathbf{f}_i$ and its neighbors $\mathbf{f}_{ij}$ is defined as:

$$\mathbf{e}_{ij} = \text{cat}[\mathbf{f}_i, \mathbf{f}_i - \mathbf{f}_{ij}], j = 1, 2, \dots k, \tag{5}$$

where cat$[\cdot, \cdot]$ denotes concatenation. Then maxpooling operation is used to extract local information from neighbour nodes. The feature $\mathbf{f}_{max} \in \mathbb{R}^{1\times d}$ can be updated using following equation:

$$^{(t+1)}\mathbf{f}_{max} = \max_{(i,j)\in\mathcal{E}} h_\theta(^{(t)}\mathbf{e}_{ij}), \tag{6}$$

where h_θ represents a linear layer followed by instance normalization [66] and LeakyReLU, and max$(\cdot)$ is element-/channel-wise maxpooling.

However, the maxpooling operation disregards the inherent structural relationships between nodes in the graph, as each graph node only retains the maximum value from its neighbors. We observe that if a pair of points are correct correspondences between the keypoints from the image and the point cloud, they should exhibit similar structures or geometric patterns within their respective neighborhoods. In order to dig more inherent structural geometry information between the node and its neighbors, we introduce **Angle-Annular convolution**. First, we use annular convolution inspired by CLNet [76] to capture the relationships of neighbors in a grouped manner. As shown in Fig. 2 bottom, k neighboring nodes are determined by ranking the Euclidean distances. Those k nodes are then divided into g groups where each group contains $\frac{k}{g}$ nodes. We use convolution layers followed by one Batch Normalization layer [26] with ReLU to process the grouped graphs. This convolution utilizes distance information indirectly by dividing nodes based on the distance between node and its neighbors. The annular feature $\mathbf{f}_{ann} \in \mathbb{R}^{1\times d}$ can be formulated as:

$$^{(t+1)}\mathbf{f}_{ann} = g_2(g_1(^{(t)}\mathbf{e}_{ij})), \tag{7}$$

where $g_1(\cdot)$ and $g_2(\cdot)$ denotes the convolution layers with $1 \times \frac{k}{g}$ kernels and $1 \times g$ kernels.

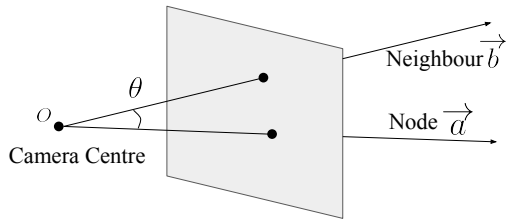

Figure 3. Illustration of angle embedding. The angle embedding between node and its neighbor is added to enhance feature representation.

Annular convolution efficiently captures structural information, but its effectiveness is limited by the variability in spatial distances between nodes and their neighbors, with some neighbors being adjacent and others more distant. Inspired by [44, 69], we incorporate geometric encoding that includes both distance and angle embeddings. The distance embedding is implicitly applied in constructing node neighborhoods. As illustrated in Fig 3, angle embedding between a node and its neighbors can also be incorporated. The cosine of the angle θ, where $\theta \in (0, \pi)$, is used as the angle embedding. The cosine value can be computed as:

$$\cos(\theta) = \frac{\overrightarrow{a} \cdot \overrightarrow{b}}{\|\overrightarrow{a}\| \|\overrightarrow{b}\|}, \tag{8}$$

where $\cos(\cdot)$ is cosine function. After getting each angle embedding within its neighbors, we get angle edge $\mathcal{E}_{ang} \in \mathbb{R}^{k \times 1}$. Here, we use annular convolution again to get angle feature $\mathbf{f}_{ang} \in \mathbb{R}^{1 \times d}$:

$${}^{(t+1)}\mathbf{f}_{ang} = g_4(g_3(\cos({}^{(t)}\mathcal{E}_{ang}))), \tag{9}$$

where $g_3(\cdot)$ and $g_4(\cdot)$ are similar convolutions as $g_1(\cdot)$ and $g_2(\cdot)$ but with independent parameters. After getting the angle feature, the annular-angle features are computed by merging the angle feature and angular feature as follows: $\mathbf{f}_{aa} = \mathbf{f}_{ann} + \mathbf{f}_{ang}$. The final feature $\mathbf{f}_{self}$ can be updated twice and the final self-attention feature $\mathbf{f}_{self}$ are obtained as:

$$\begin{aligned}\mathbf{f}_{self} &= h_{\theta 1}(\text{cat}[{}^{(0)}\mathbf{f}, {}^{(1)}\mathbf{f}_{max}, {}^{(2)}\mathbf{f}_{max}]) \\ &+ h_{\theta 2}(\text{cat}[{}^{(0)}\mathbf{f}, {}^{(1)}\mathbf{f}_{aa}, {}^{(2)}\mathbf{f}_{aa}],\end{aligned} \tag{10}$$

where $h_{\theta 1}$ and $h_{\theta 2}$ denote a linear layer followed by instance normalization [66] and LeakyReLU.

Cross-attention layer. Following [77], we apply cross-attention to enhance feature representation. Each graph node from the 2D is connected with every node from the point cloud. Specifically, given a node feature $\mathbf{f}_i$ from one modality and features $\mathbf{g}_j$ from the other modality, we form the query, keys and values according to $\mathbf{q}_i = \mathbf{W_q}\mathbf{f}_i, \mathbf{k}_j = \mathbf{W}_k\mathbf{g}_j$ and $\mathbf{v}_j = \mathbf{W}_v\mathbf{g}_j$, where $\mathbf{W}_s, \mathbf{W}_k, \mathbf{W}_v \in \mathbb{R}^{d \times d}$ are learned parameters. We update feature $\mathbf{f}_i$ by following formulation:

$$\mathbf{m}_i = \sum_j \alpha_{ij}\mathbf{v}_j \tag{11}$$

$${}^{(t+1)}\mathbf{f}_i = {}^{(t)}\mathbf{f}_i + \text{MLP}(\text{cat}[\mathbf{q}_i, \mathbf{m}_i]), \tag{12}$$

where attention weights $\alpha = \text{softmax}(\mathbf{q}_j^\top \mathbf{k}_j / \sqrt{d})$.

3.2.3 Optimal Transport

After obtaining the 2D and 3D enhanced features f_p^G and f_q^G from the geometric embedding, optimal transport is employed to assign initial correspondences, denoted as $\mathcal{M}_{init}$. We first compute the cost matrix $M \in \mathbb{R}^{M \times N}$ using the L2 distance between those two feature sets. To handle unmatched points, we extend M to $M' \in \mathbb{R}^{(M+1) \times (N+1)}$ by adding an additional row and column as dustbins. The differentiable Sinkhorn algorithm [13, 60] is then applied to solve the optimal transport problem. Finally, the initial correspondences $\mathcal{M}_{init}$ are obtained by removing the dustbins from M' and performing a mutual nearest neighbor check.

3.2.4 Outlier Rejection

After the optimal transport layer, the initial correspondences $\mathcal{M}_{init}$ still contain outliers. GoMatch [77] and DGC-GNN [69] utilize an outlier classifier [74] to predict scores indicating the probability of a correspondence being an inlier. The final correspondences $\mathcal{M}_{final}$ are obtained if the probability is over the threshold t (0.5 in experiments). The method directly inputs enhanced geometric features into the classifier. However, it brings limitations as the enhanced features are primarily designed for the optimal transport layer and are mixed together with the RGB embedding, making it difficult to extract the epipolar geometry constraints necessary for accurate outlier rejection. Earlier learning-based outlier rejection methods [74–76] leveraged keypoint locations as input, which naturally encode epipolar geometry constraints. To address this issue, we propose using bearing vectors as input for the outlier rejection process, allowing the model to utilize geometric information better and improve the accuracy of inlier detection.

3.3. Training Loss

Following GoMatch [77] and DGC-GNN [69], the loss function $\mathcal{L}$ consists of a matching loss $\mathcal{L}_m$ and an outlier rejection loss $\mathcal{L}_{or}$. The matching loss $\mathcal{L}_m$ is designed to minimize the negative log-likelihood of the matching scores.

The matching loss is presented as:

$$\mathcal{L}_m = -\frac{1}{N_m}\Big(\sum_{(i,j)\in\mathcal{M}_{gt}} \log \tilde{\mathrm{P}}_{ij} + \sum_{i\in\mathcal{U}_q} \log \tilde{\mathrm{P}}_{i(N+1)} + \sum_{j\in\mathcal{U}_d} \log \tilde{\mathrm{P}}_{(M+1)j}\Big), \tag{13}$$

where $\tilde{\mathrm{P}}$ means matching score, $\mathcal{M}_{gt}$ denotes ground truth matches, $\mathcal{U}_q$ represents unmatched query keypoints and $\mathcal{U}_d$ is unmatched database 3D points. N_m refers to the total keypoints number of ground truth, unmatched query, and unmatched database.

The outlier rejection loss $\mathcal{L}_{or}$ serves to remove incorrect matches. It enhances the model's robustness against outliers and is defined as:

$$\mathcal{L}_{or} = -\frac{1}{N_c}\sum_{i=1}^{N_c} w_i \left(y_i \log p_i + (1-y_i)\log(1-p_i)\right), \tag{14}$$

where N_c means the total number of initial correspondences. The classifier's probability output for each correspondence is indicated by p_i. The target label for the correspondence is represented by y_i, and w_i refers to the balance weight for negative and positive samples.

4. Experiments

4.1. Implementation Details

Training. We train our model on the MegaDepth dataset [37]. The number of nearest neighbors to build the local graph is set to $k = 9$. For annular convolutions, we use $g = 3$, meaning each group contains 3 nodes. The model is trained using the ADAM optimizer [33] with a learning rate $lr = 0.001$. Training is conducted on a single 32GB Tesla V100 GPU with a batch size $b = 16$. The entire training process takes around 22 hours to complete 50 epochs.

Datasets. Following the setting in GoMatch [77] and DGC-GNN [69], we train and evaluate our model on the MegaDepth dataset and conduct visual localization evaluations on Cambridge Landmark [32] and 7Scenes [59] datasets. MegaDepth is a large-scale outdoor dataset with 196 scenes captured around the world. We train our outdoor model on 99 scenes, validate on 16 scenes and test on 53 scenes. The ground truth sparse 3D point clouds are reconstructed by the COLMAP [54]. Cambridge Landmarks is a middle-scale outdoor dataset and 7Scenes is a small indoor dataset. The 3D reconstruction of Cambridge Landmarks is obtained by SfM, we evaluate the localization accuracy on four out of six scenes. The 7Scenes dataset, captured with an RGB-D camera, provides sequences where the ground truth camera poses are determined using a SLAM system, we evaluate our model on all 7Scenes as in [69, 77].

Ground Truth Correspondences. During training, we obtain the ground truth correspondences by reprojecting the 3D point clouds from the top-k retrieved images onto the query image plane. A correspondence is considered ground truth if the reprojection error is less than 0.001 in normalized image coordinates.

Evaluation Metrics. Similar to [69, 77], we report the AUC score for mean reprojection error at 1 / 5 / 10 pixel and translation and rotation errors quantiles at 25 / 50 / 75% as evaluation metrics on MegaDepth dataset. For Cambridge and 7Scenes, we report the commonly used median translation and rotation errors per scene. The camera pose estimation is determined by using the PnP-RANSAC [20, 22] for the final established correspondences. Oracle results are obtained by utilizing ground truth matches as predictions.

4.2. Results

Matching Results. Table 1 illustrates the matching results on MegaDepth dataset using top-1 and top-10 retrieved images. The proposed A2-GNN outperforms both GoMatch and DGC-GNN. With $k = 1$ retrieved image, A2-GNN shows significant improvements in reprojection AUC, outperforming GoMatch by 7.05 / 19.41 / 20.01% and DGC-GNN by 2.52 / 4.2 / 3.98%. In terms of running time, A2-GNN is slightly slower than GoMatch due to the additional encoding of RGB and geometric information, but it is x2.2 times faster than DGC-GNN by avoiding multiple times point clustering operations. When using $k = 10$ retrieved images, the same conclusion holds: A2-GNN consistently outperforms the other methods, demonstrating its robust learning capabilities.

Visual Localization. As presented in Table 2, A2-GNN outperforms GoMatch and DGC-GNN, achieving state-of-the-art results in the visual descriptor-free group. Specifically, the average pose error for the Cambridge Landmarks dataset is 40.5 cm / 1.48° for A2-GNN, compared to 54 cm / 2.23° for DGC-GNN and 173 cm / 5.87° for GoMatch. For the 7Scenes dataset, the average pose error is 14 cm / 4.04° for A2-GNN, 16 cm / 4.81° for DGC-GNN, and 22 cm / 5.77° for GoMatch. Additionally, A2-GNN retains the advantages of DF methods, including low storage requirements, privacy preservation, and low maintenance costs.

Generalizability. Similar to previous work [69, 77], we evaluate the generalizability of our model across different datasets and keypoint detectors for visual localization tasks. Our model is trained on the MegaDepth [37] using the SIFT detector and evaluated on the indoor 7Scenes dataset [59] with the SIFT keypoint detector and the outdoor Cambridge dataset [32] using the SuperPoint [14] detector. These experiments are summarized in Table 2, providing a comprehensive overview of our A2-GNN performance under varying training and evaluation conditions. Remarkably, even when using the SuperPoint detector on the Cambridge

Methods		Reproj. AUC (%) @1 / 5 / 10px (↑)	Rotation (°) Quantile @25 / 50 / 75% (↓)	Translation Quantile @25 / 50 / 75% (↓)	Time (ms) (↓)
k=1	Oracle	34.59 / 85.02 / 92.02	0.04 / 0.06 / 0.12	0.00 / 0.01 / 0.01	-
	GoMatch [77]	5.67 / 22.43 / 28.01	0.60 / 10.08 / 34.63	0.06 / 1.06 / 3.73	**24.4**
	DGC-GNN [69]	10.20 / 37.64 / 44.04	0.15 / 1.53 / 27.93	**0.01** / 0.15 / 3.00	77.8
	A2-GNN	**12.72 / 41.84 / 48.02**	**0.12 / 0.79 / 26.37**	**0.01 / 0.08 / 2.80**	34.0
k=10	GoMatch [77]	8.90 / 35.67 / 44.99	0.18 / 1.29 / 16.65	0.02 / 0.12 / 1.92	**263.1**
	DGC-GNN [69]	15.30 / 51.70 / 60.01	0.07 / 0.26 / 5.41	**0.01 / 0.02** / 0.57	701.9
	A2-GNN	**17.29 / 54.41 / 62.24**	**0.06 / 0.19 / 4.6**	**0.01 / 0.02 / 0.48**	372.0

Table 1. Matching results on MegaDepth [37]. The results include the reprojection AUC, rotation and translation errors, and inference time. Parameter k is the number of retrieved images. The best results are bold.

Methods		No Desc. Maint.	Privacy	Cambridge-Landmarks [32] (cm, °) King's	Hospital	Shop	St. Mary's	MB used	7Scenes [59] (cm, °) Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs	MB used
E2E	MS-Trans. [56]	✓	✓	83 / 1.47	181 / 2.39	86 / 3.07	162 / 3.99	71	11 / 4.66	24 / 9.60	14 / 12.19	17 / 5.66	18 / 4.44	17 / 5.94	26 / 8.45	71
	DSAC* [4]	✓	✓	**15 / 0.30**	21 / 0.40	**5 / 0.30**	13 / 0.40	112	**2** / 1.10	**2** / 1.24	**1** / 1.82	**3** / 1.15	**4** / 1.34	**4** / 1.68	**3** / 1.16	196
	HSCNet [35]	✓	✓	18 / **0.30**	**19 / 0.30**	6 / **0.30**	**9 / 0.30**	592	**2 / 0.70**	**2 / 0.90**	**1 / 0.90**	**3 / 0.80**	**4 / 1.00**	**4 / 1.20**	**3 / 0.80**	1036
DB	HybridSC [8]	✗	–	81 / 0.59	75 / 1.01	19 / 0.54	50 / 0.49	3	-	-	-	-	-	-	-	-
	AS [52]	✗	✗	13 / 0.22	20 / 0.36	**4** / 0.21	8 / 0.25	813	3 / 0.87	**2** / 1.01	**1** / 0.82	4 / 1.15	7 / 1.69	5 / 1.72	**4 / 1.01**	-
	SP [14]+SG [48]	✗	✗	**12** / 0.20	15 / 0.30	**4 / 0.20**	**7 / 0.21**	3215	**2** / 0.85	**2** / 0.94	**1 / 0.75**	**3** / 0.92	5 / 1.30	**4** / 1.40	5 / 1.47	22977
DF	GoMatch [77]	✓	✓	25 / 0.64	283 / 8.14	48 / 4.77	335 / 9.94	48	4 / 1.65	13 / 3.86	9 / 5.17	11 / 2.48	16 / 3.32	13 / 2.84	89 / 21.12	302
	DGC-GNN [69]	✓	✓	18 / 0.47	75 / 2.83	15 / 1.57	106 / 4.03	69	**3** / 1.41	**5** / 1.81	**4** / 3.13	7 / 1.66	8 / 2.03	8 / 2.14	83 / 21.53	355
	A2-GNN	✓	✓	**15 / 0.39**	**59 / 1.74**	**12 / 1.16**	**76 / 2.65**	69	**3 / 1.37**	**5 / 1.78**	**4 / 2.70**	**6 / 1.56**	**7 / 1.86**	**7 / 2.00**	**72 / 17.05**	355

Table 2. The comparison to existing localization baselines. E2E, DB and DF indicate end-to-end methods, descriptor-based and descriptor-free methods, respectively. Median translation and rotation errors for each scene are reported, as well as storage demand. The best results in each group are bold.

Methods	OR Input	Self-Attention Maxpooling	Annular	Angle	Color	Reproj. AUC (%) @1 / 5 / 10px (↑)	Rotation (°) Quantile@25 / 50 / 75% (↓)	Translation Quantile@25 / 50 / 75% (↓)
GoMatch [77]	Feat.	✓				8.90 / 35.67 / 44.99	0.18 / 1.29 / 16.65	0.02 / 0.12 / 1.92
Variants	BV	✓				10.57 / 39.22 / 47.98	0.13 / 0.96 / 17.77	0.01 / 0.09 / 1.96
	BV		✓			12.02 / 42.42 / 50.67	0.10 / 0.67 / 15.99	0.01 / 0.06 / 1.76
	BV	✓	✓			14.04 / 47.49 / 55.82	0.08 / 0.40 / 11.40	0.01 / 0.04 / 1.20
	BV	✓	✓	✓		14.79 / 49.11 / 57.41	0.08 / 0.33 / 9.14	0.01 / 0.03 / 1.06
	Feat.	✓	✓	✓	✓	15.97 / 52.01 / 59.88	0.06 / 0.23 / 7.34	0.01 / 0.02 / 0.79
A2-GNN	BV	✓	✓	✓	✓	**17.29 / 54.41 / 62.24**	**0.06 / 0.19 / 4.60**	**0.01 / 0.02 / 0.48**

Table 3. Architecture Ablations. The reprojection AUC, rotation and translation errors are reported. OR Input indicates the input types of outlier rejection. Feat. means enhanced feature from GNN as input, and BV refers to bearing vector. The best results are bold.

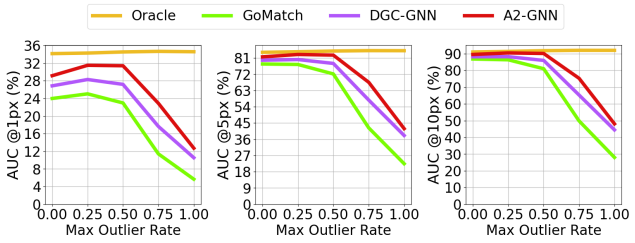

Figure 4. Outlier Sensitivity. Comparison of the AUC for GoMatch [77], DGC-GNN [69], and the proposed A2-GNN under different outlier ratios at 1, 5, and 10 pixels thresholds. Oracle is the upper bound by using ground truth matches.

Method	Reproj. AUC (%)	Rotation (°)	Translation (m)
	@1 / 5 / 10px (↑)	Quantile @25 / 50 / 75% (↓)	
GoMatch (no OR)	4.47 / 17.95 / 23.42	1.29 / 11.85 / 33.60	0.11 / 1.18 / 3.58
DGC-GNN (no OR)	8.56 / 30.79 / 37.02	0.22 / 4.85 / 30.07	0.02 / 0.47 / 3.10
A2-GNN (no OR)	**11.37 / 37.04 / 43.15**	**0.13 / 2.32 / 27.00**	**0.01 / 0.22 / 2.87**

Table 4. Ablation results without outlier rejection on MegaDepth on top-1 image retrieval.

dataset, A2-GNN demonstrates strong performance, highlighting its robust generalizability.

Ablation Studies. We validate our A2-GNN module design by conducting ablation studies on the MegaDepth dataset [37] with the top-10 retrieved images ($k = 10$). The ablation results are presented in Table 3. First, we analyze the impact of the outlier rejection inputs by changing the input from enhanced features to bearing vectors. This change improves the reprojection AUC by 1.67 / 3.55 / 2.99 %, confirming our assumption that explicit positional information can enhance consensus in epipolar geometry, thereby improving outlier rejection accuracy. Next, we perform ablations on our geometric feature embedding. Incorporating both max-pooling and annular convolution significantly boosts reprojection AUC by 3.47 / 8.27 / 7.84 %. We can conclude that max-pooling operation holds the permutation invariance, a property that is vital for robust feature learning. Additionally, angle embedding further enhances geometric embedding and shows improvement by 0.75 / 1.62 / 1.59%. Similar to [69], we also observe notable gains when color information is added to the point feature.

The evaluation results without outlier rejection are shown in Table 4. We use image retrieval with $k = 1$ and set the outlier rejection threshold to 0. Our A2-GNN demonstrates the ability to generate high-quality 2D-3D correspondences without the need for outlier rejection, outperforming both GoMatch and DGC-GNN by large margins.

Sensitivity to Keypoint Outliers. To evaluate our method's performance in handling keypoint outliers, we follow the procedure outlined in [69, 77] and vary the outlier ratios from 0 to 1 during inference. We present the AUC across various pixel thresholds and outlier ratios in Fig. 4. When the outlier ratio is 0, all input keypoints are derived from the ground truth, meaning no outliers are present. Conversely, at a ratio of 1, all keypoints are directly taken from the 2D query and the 3D points from the top-k retrieved images.

As shown in Fig. 4, our A2-GNN model consistently outperforms GoMatch and DGC-GNN across all outlier rate scenarios. When the outlier rate is below 50%, the AUC of our method closely approaches the Oracle upper bound However, when the outlier rate exceeds 50%, performance declines rapidly. This is due to the increased difficulty in identifying correct matches in high-outlier scenarios. Additionally, our model was trained under conditions with a 50% outlier rate and thus does not perform as well when inputs are chosen randomly.

5. Conclusion

This paper introduces A2-GNN, a novel local graph neighbor aggregation method for direct 2D-3D matching without relying on visual descriptors. Our angle-annular convolution method effectively captures more robust local geometric information. Compared to the state-of-the-art descriptor-free matcher DGC-GNN [69], A2-GNN demonstrates superior accuracy and efficiency, leading to significantly improved localization performance. These advancements help bridge the gap between descriptor-based and descriptor-free methods while addressing the limitations of descriptor-based approaches, such as high storage requirements, maintenance complexity, and privacy concerns. Overall, A2-GNN represents a substantial advancement in the field of direct 2D-3D matching.

Limitations. While the proposed A2-GNN method demonstrates significant efficiency and accuracy improvements over existing descriptor-free approaches, its performance still lags behind that of traditional descriptor-based algorithms. This shortfall is primarily due to noisy keypoint inputs from query images and the limited information available. Specifically, when image keypoints contain more than 50% outliers, the performance drops dramatically. Although A2-GNN advances the state of descriptor-free methods, further research is needed to bridge this gap.

Acknowledgements. This work was supported by the Finnish Doctoral Program Network in Artificial Intelligence, AI-DOC (decision number VN/3137/2024-OKM-6). We acknowledge the computational resources provided by the CSC-IT Center for Science, Finland.

References

[1] Relja Arandjelović and Andrew Zisserman. Three things everyone should know to improve object retrieval. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2911–2918. IEEE, 2012. 2

[2] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: CNN architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5297–5307, 2016. 2

[3] Mark Billinghurst, Adrian Clark, Gun Lee, et al. A survey of augmented reality. *Foundations and Trends® in Human–Computer Interaction*, 8(2-3):73–272, 2015. 1

[4] Eric Brachmann and Carsten Rother. Visual camera relocalization from rgb and rgb-d images using dsac. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):5847–5865, 2021. 3, 7

[5] Eric Brachmann, Tommaso Cavallari, and Victor Adrian Prisacariu. Accelerated coordinate encoding: Learning to relocalize in minutes using rgb and poses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5044–5053, 2023. 3

[6] 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 Transactions on robotics*, 32(6):1309–1332, 2016. 1

[7] Dylan Campbell, Liu Liu, and Stephen Gould. Solving the blind perspective-n-point problem end-to-end with robust differentiable geometric optimization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 244–261. Springer, 2020. 1, 2, 4

[8] Federico Camposeco, Andrea Cohen, Marc Pollefeys, and Torsten Sattler. Hybrid scene compression for visual localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7653–7662, 2019. 3, 7

[9] Song Cao and Noah Snavely. Minimal scene descriptions from structure from motion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 461–468, 2014. 3

[10] Julie Carmigniani and Borko Furht. Augmented reality: an overview. *Handbook of augmented reality*, pages 3–46, 2011. 1

[11] Kunal Chelani, Fredrik Kahl, and Torsten Sattler. How privacy-preserving are line clouds? recovering scene details from 3d lines. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15668–15678, 2021. 1

[12] Hongkai Chen, Zixin Luo, Lei Zhou, Yurun Tian, Mingmin Zhen, Tian Fang, David Mckinnon, Yanghai Tsin, and Long Quan. Aspanformer: Detector-free image matching with adaptive span transformer. In *European Conference on Computer Vision*, pages 20–36. Springer, 2022. 2

[13] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013. 5

[14] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018. 2, 6, 7

[15] Tien Do, Ondrej Miksik, Joseph DeGol, Hyun Soo Park, and Sudipta N Sinha. Learning to detect scene landmarks for camera localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11132–11142, 2022. 3

[16] Hao Dong, Xieyuanli Chen, Mihai Dusmanu, Viktor Larsson, Marc Pollefeys, and Cyrill Stachniss. Learning-based dimensionality reduction for computing compact and effective local feature descriptors. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6189–6195. IEEE, 2023. 3

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

[18] Mihai Dusmanu, Ondrej Miksik, Johannes L Schönberger, and Marc Pollefeys. Cross-descriptor visual localization and mapping. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6058–6067, 2021. 1

[19] Marcin Dymczyk, Simon Lynen, Titus Cieslewski, Michael Bosse, Roland Siegwart, and Paul Furgale. The gist of maps-summarizing experience for lifelong localization. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 2767–2773. IEEE, 2015. 3

[20] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 6

[21] Jorge Fuentes-Pacheco, José Ruiz-Ascencio, and Juan Manuel Rendón-Mancha. Visual simultaneous localization and mapping: a survey. *Artificial intelligence review*, 43:55–81, 2015. 1

[22] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE transactions on pattern analysis and machine intelligence*, 25(8):930–943, 2003. 6

[23] Yixiao Ge, Haibo Wang, Feng Zhu, Rui Zhao, and Hongsheng Li. Self-supervising fine-grained region similarities for large-scale image localization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 369–386. Springer, 2020. 2

[24] Albert Gordo, Jon Almazan, Jerome Revaud, and Diane Larlus. End-to-end learning of deep visual representations for image retrieval. *International Journal of Computer Vision*, 124(2):237–254, 2017. 2

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceed-*

ings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016. 4

[26] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015. 4

[27] Sergio Izquierdo and Javier Civera. Optimal transport aggregation for visual place recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

[28] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence*, 33(1): 117–128, 2010. 3

[29] Hanwen Jiang, Arjun Karpur, Bingyi Cao, Qixing Huang, and André Araujo. Omniglue: Generalizable feature matching with foundation model guidance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19865–19875, 2024. 2

[30] Yan Ke and Rahul Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, pages II–II. IEEE, 2004. 3

[31] Nikhil Keetha, Avneesh Mishra, Jay Karhade, Krishna Murthy Jatavallabhula, Sebastian Scherer, Madhava Krishna, and Sourav Garg. Anyloc: Towards universal visual place recognition. *IEEE Robotics and Automation Letters*, 2023. 2

[32] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015. 6, 7

[33] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[34] Zakaria Laskar, Iaroslav Melekhov, Assia Benbihi, Shuzhe Wang, and Juho Kannala. Differentiable product quantization for memory efficient camera relocalization. *arXiv preprint arXiv:2407.15540*, 2024. 3

[35] Xiaotian Li, Shuzhe Wang, Yi Zhao, Jakob Verbeek, and Juho Kannala. Hierarchical scene coordinate classification and regression for visual localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11983–11992, 2020. 7

[36] Yunpeng Li, Noah Snavely, and Daniel P Huttenlocher. Location recognition using prioritized feature matching. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part II 11*, pages 791–804. Springer, 2010. 3

[37] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2041–2050, 2018. 6, 7, 8

[38] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. Lightglue: Local feature matching at light speed. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17627–17638, 2023. 2

[39] Heejoon Moon, Chunghwan Lee, and Je Hyeong Hong. Efficient privacy-preserving visual localization using 3d ray clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9773–9783, 2024. 3

[40] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 33(5):1255–1262, 2017. 1

[41] Linfei Pan, Johannes L Schönberger, Viktor Larsson, and Marc Pollefeys. Privacy preserving localization via coordinate permutations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18174–18183, 2023. 1, 3

[42] Maxime Pietrantoni, Martin Humenberger, Torsten Sattler, and Gabriela Csurka. Segloc: Learning segmentation-based representations for privacy-preserving visual localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15380–15391, 2023. 3

[43] Francesco Pittaluga, Sanjeev J Koppal, Sing Bing Kang, and Sudipta N Sinha. Revealing scenes by inverting structure from motion reconstructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 145–154, 2019. 3

[44] Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, and Kai Xu. Geometric transformer for fast and robust point cloud registration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11143–11152, 2022. 5

[45] Jerome Revaud, Jon Almazán, Rafael S Rezende, and Cesar Roberto de Souza. Learning with average precision: Training image retrieval with a listwise loss. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5107–5116, 2019. 2

[46] Jerome Revaud, Cesar De Souza, Martin Humenberger, and Philippe Weinzaepfel. R2d2: Reliable and repeatable detector and descriptor. *Advances in neural information processing systems*, 32, 2019. 2

[47] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12716–12725, 2019. 1

[48] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020. 1, 2, 7

[49] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Fast image-based localization using direct 2d-to-3d matching. In *2011 International Conference on Computer Vision*, pages 667–674. IEEE, 2011. 1, 2

[50] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Improving image-based localization by active correspondence search. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13,*

2012, Proceedings, Part I 12, pages 752–765. Springer, 2012. 1, 2

[51] Torsten Sattler, Michal Havlena, Filip Radenovic, Konrad Schindler, and Marc Pollefeys. Hyperpoints and fine vocabularies for large-scale location recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2102–2110, 2015. 3

[52] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1744–1756, 2016. 7

[53] Torsten Sattler, Akihiko Torii, Josef Sivic, Marc Pollefeys, Hajime Taira, Masatoshi Okutomi, and Tomas Pajdla. Are large-scale 3d models really necessary for accurate visual localization? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1637–1646, 2017. 1

[54] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 6

[55] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 1

[56] Yoli Shavit, Ron Ferens, and Yosi Keller. Learning multi-scene absolute pose regression with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2733–2742, 2021. 7

[57] Yan Shi, Jun-Xiong Cai, Yoli Shavit, Tai-Jiang Mu, Wensen Feng, and Kai Zhang. Clustergnn: Cluster-based coarse-to-fine graph neural network for efficient feature matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12517–12526, 2022. 2

[58] Mikiya Shibuya, Shinya Sumikura, and Ken Sakurada. Privacy preserving visual slam. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*, pages 102–118. Springer, 2020. 3

[59] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2930–2937, 2013. 6, 7

[60] Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967. 5

[61] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM siggraph 2006 papers*, pages 835–846, 2006. 1

[62] Pablo Speciale, Johannes L Schonberger, Sing Bing Kang, Sudipta N Sinha, and Marc Pollefeys. Privacy preserving image-based localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5493–5503, 2019. 3

[63] Pablo Speciale, Johannes L Schonberger, Sudipta N Sinha, and Marc Pollefeys. Privacy preserving image queries for camera localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1486–1496, 2019. 3

[64] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8922–8931, 2021. 2

[65] Michał Tyszkiewicz, Pascal Fua, and Eduard Trulls. Disk: Learning local features with policy gradient. *Advances in Neural Information Processing Systems*, 33:14254–14265, 2020. 2

[66] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 4, 5

[67] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[68] Shuzhe Wang, Zakaria Laskar, Iaroslav Melekhov, Xiaotian Li, and Juho Kannala. Continual learning for image-based camera localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3252–3262, 2021. 1, 3

[69] Shuzhe Wang, Juho Kannala, and Daniel Barath. Dgc-gnn: Leveraging geometry and color cues for visual descriptor-free 2d-3d matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20881–20891, 2024. 1, 2, 4, 5, 6, 7, 8

[70] Shuzhe Wang, Zakaria Laskar, Iaroslav Melekhov, Xiaotian Li, Yi Zhao, Giorgos Tolias, and Juho Kannala. Hscnet++: Hierarchical scene coordinate classification and regression for visual localization with transformer. *International Journal of Computer Vision*, pages 1–21, 2024. 1, 3

[71] Yifan Wang, Xingyi He, Sida Peng, Dongli Tan, and Xiaowei Zhou. Efficient loftr: Semi-dense local feature matching with sparse-like speed. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21666–21675, 2024. 2

[72] Changchang Wu. Towards linear-time incremental structure from motion. In *2013 International Conference on 3D Vision-3DV 2013*, pages 127–134. IEEE, 2013. 1

[73] Luwei Yang, Rakesh Shrestha, Wenbo Li, Shuaicheng Liu, Guofeng Zhang, Zhaopeng Cui, and Ping Tan. Scenesqueezer: Learning to compress scene for camera relocalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8259–8268, 2022. 3

[74] Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to find good correspondences. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2666–2674, 2018. 2, 5

[75] Jiahui Zhang, Dawei Sun, Zixin Luo, Anbang Yao, Lei Zhou, Tianwei Shen, Yurong Chen, Long Quan, and Hongen Liao. Learning two-view correspondences and geometry using order-aware network. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5845–5854, 2019.

[76] Chen Zhao, Yixiao Ge, Feng Zhu, Rui Zhao, Hongsheng Li, and Mathieu Salzmann. Progressive correspondence pruning by consensus learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6464–6473, 2021. 2, 4, 5

[77] Qunjie Zhou, Sérgio Agostinho, Aljoša Ošep, and Laura Leal-Taixé. Is geometry enough for matching in visual localization? In *European Conference on Computer Vision*, pages 407–425. Springer, 2022. 1, 2, 4, 5, 6, 7, 8

[78] Marco Zuliani. Ransac for dummies. *Vision Research Lab, University of California, Santa Barbara*, 2009. 1