# Is Geometry Enough for Matching in Visual Localization?

Qunjie Zhou*[1], Sérgio Agostinho*[2], Aljoša Ošep[1], and Laura Leal-Taixé[1]

[1]Technical University of Munich, Germany [2]Universidade de Lisboa, Portugal
[1]{qunjie.zhou,aljosa.osep,leal.taixe}@tum.de
[2]sergio.agostinho@tecnico.ulisboa.pt
https://github.com/dvl-tum/gomatch

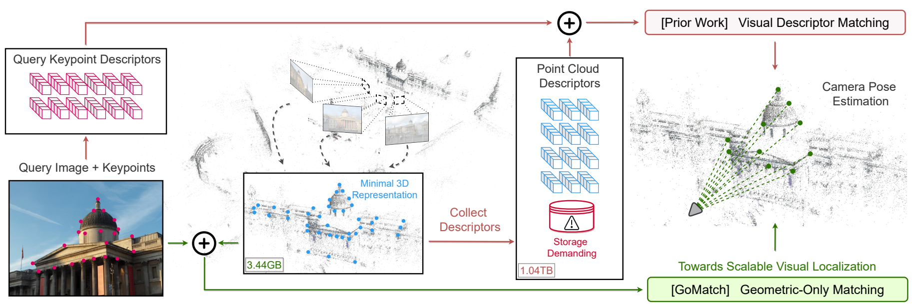


**Fig. 1.** In this work, we propose GoMatch to tackle visual localization w.r.t. a scene represented as a 3D point cloud. By relying only on geometric information for matching, GoMatch allows structure-based methods to achieve localization solely through the use of keypoints, sidestepping the need to store visual descriptors for matching. Keeping only the minimal representation of a 3D model, *i.e.*, its coordinates, leads to a more scalable pipeline towards large-scale localization that bypasses privacy concerns and is easy to maintain.

**Abstract.** In this paper, we propose to go beyond the well-established approach to vision-based localization that relies on visual descriptor matching between a query image and a 3D point cloud. While matching keypoints via visual descriptors makes localization highly accurate, it has significant storage demands, raises privacy concerns and requires update to the descriptors in the long-term. To elegantly address those practical challenges for large-scale localization, we present GoMatch, an alternative to *visual-based matching* that solely relies on geometric information for matching image keypoints to maps, represented as sets of bearing vectors. Our novel bearing vectors representation of 3D points, significantly relieves the cross-modal challenge in *geometric-based matching* that prevented prior work to tackle localization in a realistic environment. With additional careful architecture design, GoMatch improves over prior geometric-based matching work with a reduction of $(10.67m, 95.7°)$ and $(1.43m,\ 34.7°)$ in average median pose errors on Cambridge Landmarks and 7-Scenes, while requiring as little as 1.5/1.7%

* Equal contribution.

of storage capacity in comparison to the best visual-based matching methods. This confirms its potential and feasibility for real-world localization and opens the door to future efforts in advancing city-scale visual localization methods that do not require storing visual descriptors.

## 1 Introduction

In this paper we tackle scalable, data-driven visual localization. The ability to localize a query image within a 3D map based representation of the environment is vital in many applications, ranging from robotics to virtual and augmented reality. In past years, researchers have made a significant progress in vision-based localisation [25, 20, 72, 42, 46, 51, 30, 74, 65, 54]. The majority of methods [67, 25, 72, 51, 65] rely on a pre-built 3D representation of the environment, typically obtained using structure-from-motion (SfM) techniques [57, 59]. Such 3D maps store 3D points and $D$-dimensional visual feature descriptors [55]. To determine the pose of a query image, *i.e.*, its 3D position and orientation, these methods match visual descriptors, obtained from the query image, with the ones stored in the point cloud. Once image-to-point-cloud matches are established, a Perspective-n-Point (PnP) solver [36, 27] is used to estimate the camera pose. While working well in practice, this approach suffers from several drawbacks. First, we need to explicitly store per-point visual descriptors for point clouds, which hinders its applicability to large-scale environments due to the expensive storage requirement. Second, this limits the applicability to point clouds with specific descriptors, which increases the 3D map descriptor maintenance effort – maps need to be re-built or updated to be used in conjunction with newly developed descriptors [24]. Third, this approach in practice necessitates a visual descriptor exchange between the server (storing the 3D model and descriptors) and an online feature extractor. This is a point of privacy vulnerability, as human identities and personal information can be recovered from visual descriptors intercepted during the transmission [48, 23, 22, 16, 26, 29, 63, 28]. The aforementioned issues lead to the main question we pose in this paper: *can we localize an image without relying on visual descriptors?* This would significantly reduce the map storage demands and get rid of descriptor maintenance. Recently, Campbell *et al*. [10, 40] showed that it is feasible to directly match 2D image keypoints with a 3D point cloud using only geometrical cues. However, this is limited to ideal scenarios where outliers are not present. This assumption does not hold in real-world scenes and is not directly applicable to challenging visual localization. This is not surprising, as relying only on geometrical cues is a significantly more challenging compared to matching visual descriptors. In contrast to a single 2D/3D point coordinate, a visual descriptor provides a rich visual context, since it is commonly extracted from the local image patch centered around a keypoint [25, 20, 72, 42].

In this paper, we achieve significant progress in making keypoints-to-point cloud direct matching ready for real-world visual localization. To cope with noisy images, point clouds, and inevitably keypoint outliers, we present **GoMatch**, a novel neural network architecture that relies on **G**eometrical information **o**nly. GoMatch leverages self- and cross- attention mechanisms to establish initial correspondences between image keypoints and point clouds, and further improves the matching robustness by filtering match outliers using a classifier. To the best of our knowledge, GoMatch is the first approach that is applicable to visual localization *in the wild* and does not rely on storage-demanding visual descriptors. In particular, compared to its prior work on geometric matching-based localization, GoMatch leads to a reduction of $(10.67m, 95.7°)$ and $(1.43m, 34.7°)$ in average median pose errors on Cambridge Landmarks dataset [35] and 7-Scenes dataset [61], confirming its potential in real-world visual localization.

We summarize our contributions as the following: (i) we develop a novel method to match query keypoints to a point cloud relying only on geometrical information; (ii) We bridge the difference in data modalities between a 2D image keypoint to a 3D point by representing it with its bearing vectors projected into co-visible reference views and show this is remarkably more robust compared to direct cross-modal matching; (iii) Our extensive evaluation shows that our method significantly outperforms prior work, effectively enabling real-world visual localization based on geometric-only matching; (iv) Finally, we thoroughly compare our method to the well-established visual localization baselines and discuss advantages and disadvantages of each approach. With this analysis, we hope to open the door for future progress towards more general and scalable structure-based methods for visual localization, which do not critically rely on storing visual descriptors, thereby reducing storage, relieving privacy concerns and eliminating the need for descriptor maintenance.

## 2 Related Work

**Structure-Based Localization.** Methods of this kind [53, 50, 66, 58, 5] commonly establish explicit correspondences between the query image pixels and the 3D points of the environment to compute the query image pose from the established matches using PnP solvers [36, 27]. Keypoint correspondences are made by computing and matching visual descriptors for each keypoint from a query and database images [25, 20, 72, 42, 30, 51, 65]. Another recent work [52] iteratively optimizes a camera pose by minimizing visual descriptor distances between the 3D points observed in the query and the reference images. While it does not establish matches, it relies on visual descriptors extracted from a neural network and requires 3D points. Structure-based localization methods achieve impressive localization accuracy and state-of-the-art performance [50, 20, 51] in the long-term localization benchmark [54, 67].

**Practical Challenges in Structure-Based Localization.** Despite being highly accurate, modern localization solutions encounter practical challenges when deployed onto real-life applications, spanning city-level scale. The challenges are threefold: i) Relying on visual descriptors [20, 25, 72, 42] makes the system demanding in storage[1] as shown in Table 1. To reduce storage require-

[1] *Storage* as in non-volatile preservation of data, in contrast to volatile *memory*.

**Table 1.** On the challenges of large-scale structure-based localization. Analysis is performed on the MegaDepth [39] composed of many landmarks (similar to city districts), acting as an example of a city-scale dataset. We compare visual-based matching (VM) and geometric-based matching (GM) methods by analysing their storage requirement and considering whether a method requires to maintain map descriptors as well as provides privacy protection (*c.f.* the supplementary for more details.) For structured-based localization, scene coordinates (3D) and camera metadata (Cameras) are stored to obtain 2D-3D correspondences. In contrast to VM methods that need to additionally store visual descriptors or extract descriptors on-the-fly from the raw images, we show that using GM instead of VM, significantly reduces storage requirements, safeguards user privacy and bypasses the need for descriptor maintenance [24].

| | Method | Desc. Maintenance | Privacy | Database Storage (GB, ↓) Cameras (MB) | 3D | Raw Ims | Descs | Total |
|---|---|---|---|---|---|---|---|---|
| VM | SIFT [41] | ✗ | ✗ | 15.73 | 3.44 | ✗ | 130.10 (uint8) | 133.33 |
| VM | SuperPoint [20] | ✗ | ✗ | 15.73 | 3.44 | ✗ | 1040.76 (fp32) | 1044.21 |
| VM | Extract on-the-fly | ✗ | ✗ | 15.73 | 3.44 | 157.84 | ✗ | 161.29 |
| Geometric-based Matching | | ✓ | ✓ | 15.73 | 3.44 | ✗ | ✗ | **3.45** |

ment of the 3D scene representation, compression can be done by keeping a subset of the 3D points [14, 13, 43] and quantising [13, 69, 17] the descriptors associated with the 3D points. HybridSC [13] stands out among the existing work, with its extreme compression rate and minimal accuracy loss. ii) Localization methods following a server-client model need to transmit visual descriptors between the server and client, which exposes the model to a risk of a privacy breach [48, 23, 22, 16]. To mitigate this issue, recent work [47, 26] developed descriptors that are more robust against privacy attacks with slightly lower accuracy. iii) With the ongoing advancements in local features methods [20, 25, 72, 42, 47, 26], continuously updating scene descriptors is a foreseeable demand [24] for visual-based matching methods. However, such an update requires either re-building the map with new descriptors or transforming the existing descriptors [24] to new ones. In this paper, we propose an *orthogonal* direction to address the storage, privacy and descriptor maintenance challenges in structure-based localization by relying solely on more lightweight geometric information for matching.

**End-to-End Learned Localization.** A recent trend of methods leverage data-driven techniques to learn to localize in an end-to-end manner, without relying on point clouds. This is achieved by either regressing scene coordinates, regressing the camera's absolute pose or regressing its relative pose w.r.t. to a database image. Scene coordinate regression methods [5, 6, 8, 3, 15, 38, 73] directly regress dense 3D scene coordinates from 2D images. However, they need to be re-trained for every new scene due to their lack of generalization [5, 6, 15, 7]. In certain cases, multiple instances of the same network are trained on sub-regions of the scene, due to the limited capacity of a single network [7]. Therefore, it is unclear how to scale these methods [5, 6, 8, 3, 15, 38, 73], that are traditionally evaluated only on small indoor rooms, to large-scale scenes. Absolute pose regression

(APR) methods implicitly encode the scene representation inside the network and directly regress the pose from the query image [35, 33, 34, 49, 71]. While earlier methods required training a model per scene and have been shown to overfit to the viewpoints and appearance of the training images [56], recent work in multi-scene APR [4, 60] loosened the per-scene training requirements. Compared to multi-scene APR, our method generalizes across scenes as other structure-based localization methods (*c.f.* Section 5.5) while addressing its aforementioned practical challenges. Another related approach that sidesteps maintaining a 3D model with visual descriptors, is to regress relative camera poses [21, 75, 2, 37] from a query image to its relevant database images. However, directly regressing the geometric transformations in general leads to limited generalization [56, 75].

**Direct Geometric Keypoint Matching.** Matching image keypoints directly to 3D point clouds while jointly estimating pose has been widely investigated under relatively constrained environments [19, 45, 9, 11, 12, 40, 10]. Some require pose initialization [19] or pose distribution priors [45], while others, based on globally optimal estimators, have prohibitive runtime requirements in order to produce accurate estimates [9, 11, 12]. In contrast, the recent state-of-the-art, data-driven, geometric matching approaches [40, 10] strike a good compromise between pose accuracy and time required to produce an accurate estimate. Despite not producing globally optimal solutions, BPnPNet [10] is able to estimate a reliable pose in a fraction of a second. Given a set of 2D keypoints in the query image and a set of 3D points in the scene point cloud, BPnPNet jointly estimates matches between these two sets *purely* based on geometric information. However, this approach was shown to work in idealistic scenarios assuming no outlier keypoints and, as we experimentally demonstrate, the matching performance degrades significantly once outliers are introduced. The outlier-free assumption clearly does not hold for challenging real-world localization scenarios as map building and keypoint detection are all challenging tasks, prone to errors and noise. In our work, we build upon BPnPNet and design a geometric matching module that is robust against keypoint outliers. We show in Section 5.3 that our approach significantly outperforms BPnPNet in matching keypoints with noisy outliers, effectively enabling the applicability of geometric-based matching to real-world visual localization.

## 3 Task Definition

**Structure-Based Localization Pipeline.** Structure-based methods assume as input a query image, a 3D point cloud of the scene, and database images with known poses. These methods first retrieve a set of database images that are co-visible with the query image, *i.e.*, have a visual overlap, as illustrated in Fig. 2. Next, after narrowing down the search space, they establish 2D-3D correspondences between the query image keypoints and a (retrieved) subset of the 3D point cloud. This set of correspondences can be used to estimate the query image pose using a PnP solver [32, 27]. The majority of prior work [25, 20, 72, 42, 30, 51, 50] rely on storage-consuming visual descriptors, stored together

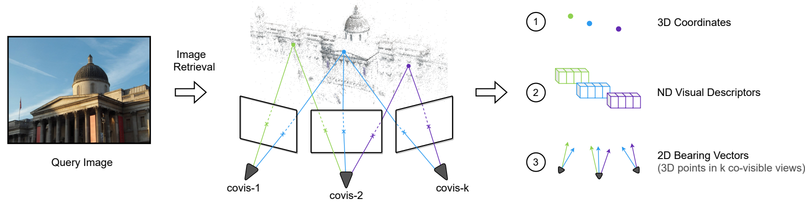


**Fig. 2.** Co-visible views & keypoint representations. Retrieving co-visible reference images (views) of a query image, narrows the matching against a full 3D point cloud to a subset of points that are more likely to be visible to the query image. Each 3D point can be represented differently by: 1) its 3D coordinate; 2) a visual descriptor that incorporates local appearance; or 3) a bearing vector that represents the direction from the reference camera origin to a 3D point in normalized coordinates. In this paper, we explore keypoint matching using representations 1) and 3).

with the point cloud, to establish 2D-3D matches. The key challenge we address is how to establish those correspondences *without* visual descriptors.

**Problem Formulation.** We assume two point sets, one with 2D keypoint coordinates in the image plane $\mathbf{p}_i \in \mathbb{R}^2$, and the second containing 3D point coordinates $\mathbf{q}_j \in \mathbb{R}^3$. We seek the matching set $\mathcal{M} := \{(i,j) | \mathbf{p}_i = \pi(\mathbf{q}_j; \mathtt{K}, \mathtt{R}, \mathbf{t})\}$, *i.e.*, the set of index pairs $i$ and $j$, for which if the $j$-th 3D keypoint is projected to the image plane, it matches the coordinates specified by the corresponding $i$-th 2D point. The camera intrinsic matrix $\mathtt{K} \in \mathbb{R}^{3\times3}$ is assumed to be known, and the operator $\pi(\cdot)$ represents the camera projection function, which transforms 3D points onto the camera's frame of reference and projects them to the image plane according to the camera's intrinsics. Our goal is to find the correct 2D-3D keypoint matches for accurate pose estimation.

**Keypoint Representation.** We represent 2D pixels using 2D coordinates $(u, v) \in \mathbb{R}^2$ in the image plane. To learn a matching function that is agnostic to different camera models, we uplift those 2D points into a bearing vector representation $\mathbf{b} \in \mathbb{R}^2$, effectively removing the effect of the camera intrinsics. Bearing vectors encode the direction (or bearing) of points in a camera's frame of reference. We compute bearing vectors from image pixels as: $[\mathbf{b}^\top\, 1]^\top \propto \mathtt{K}^{-1}[u\, v\, 1]^\top$. For a 3D point, we consider two different representations (see Fig. 2): (i) as 3D coordinates $(x, y, z) \in \mathbb{R}^3$ w.r.t. a 3D world reference/origin; and (ii) as a bearing vector w.r.t. a reference database image. The bearing vector representation allows bringing both 2D pixels and 3D points to the same data modality. Given a 3D point $\mathbf{p} \in \mathbb{R}^3$ and transformation $(\mathtt{R},\, \mathbf{t})$ from the world to the database image's frame of reference, we compute the corresponding bearing vector as:

$$\mathbf{p}' = \mathtt{R}\mathbf{p} + \mathbf{t}, \;\; [\mathbf{b}^\top\, 1]^\top = \mathbf{p}'/p'_z, \tag{1}$$

where $\mathbf{p}'$ represents $\mathbf{p}$ in the camera's frame of reference, and $\mathbf{p}'_z$ represents its $z$ coordinate. As shown in Table 1, these geometric-based point representations require significantly lower storage compared to visual descriptor based ones, *e.g.*, as low as 3% compared to the storage of modern descriptors.

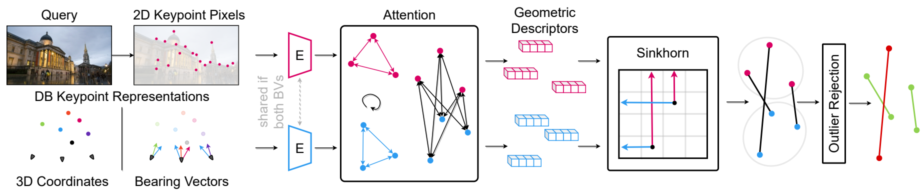


**Fig. 3.** GoMatch components overview. The query image and database keypoints first undergo a feature encoder **E** to generate per-point features. We share encoders in the query and database branch when database points are represented as bearing vectors otherwise not. These features are refined in the attention layer and then used in the Sinkhorn matching stage to establish an initial set of candidate matches, from which erroneous matches are filtered with an outlier rejection layer.

## 4 Geometric-Only Matching

**BPnPNet in a Nutshell.** BPnPNet [10] made great progress towards establishing correspondence between the query keypoints and 3D point cloud in the absence of visual descriptors. It proposes an end-to-end trainable, differentiable matcher that performs 2D to 3D cross modal matching without relying on appearance information. While this is a step in the right direction, we show in Section 5.3 that it does not scale to the real-world visual localization scenarios where outliers, *i.e.* points without a match, are pervasive. Direct 2D-3D matching of sparse keypoints is a challenging problem due to low amount of discriminative data, *i.e.* points no longer have a local visual appearance, and its cross-modal nature. In a nutshell, BPnPNet (i) encodes points to obtain per point features, (ii) establishes matches using the Sinkhorn algorithm [18, 62], which finds the optimal assignment between geometrical features, and finally, (iii) leverages a differentiable PnP solver that imposes an additional pose supervision on the network. In the following, we build on the observation that the lightweight geometric feature encoder does not possess the necessary representational power to produce features that generalize simultaneously to situations with and without outliers.

### 4.1 GoMatch: Embracing Outliers

In GoMatch we (i) propose architectural changes that enable resilience to outliers and (ii) cast the *cross-modal* nature of 2D-3D matching to an *intra-modal* setting through the use of bearing vectors. Below, we explain the details of these contributions, which are experimentally validated to be all necessary and critical to outlier-robust geometric matching in Section 5.3. We refer to Fig. 3 for a visual overview of the entire network. Furthermore, we add an outlier rejection layer to retain only quality matches from the Sinkhorn outputs. While we introduce the novel network components in the following paragraphs, we refer the reader to the supplementary material for an in-depth description of all network components.

**Feature Refinement through Attention.** In BPnPNet, each keypoint node is processed in parallel with an MLP-style encoder to extract features directly for matching, and information exchange happens only in the Sinkhorn matching stage. This might lead to a learned feature representation which lacks context information within each 2D/3D modality and cross modality. Based on this assumption, we explore adding information exchange prior to matching. To enhance the context information within each modality, we apply *self-attention* to the raw encoded features where a graph neural network [31] refines features of every keypoint by exchanging the information with a fixed number of closest neighbors in coordinate space. This is followed by *cross-attention* [70], where every keypoint from one modality will interact with all keypoints from the other modality through a sequence of multi-head attention layers. By stacking several blocks of such self-/cross-attention layers, we are able to learn more representative features, which allows Sinkhorn to identify significantly better outlier matches.
**Outlier Rejection.** After Sinkhorn matching, the estimated corresponding pairs may still contain outlier matches. To filter those, we follow [44] and add a classifier that takes in the concatenated geometric features from the query and database keypoints, and predicts confidence scores for all matches. Estimated correspondences with confidence below a threshold (0.5 in practice) are rejected.
**Matching with Bearing Vectors.** Directly matching 2D keypoints to cross-modal 3D coordinates is challenging because it requires the network to learn features that have to consider not only the relationship between keypoints, but also the influence of different camera poses. Furthermore, the different distributions of 3D point clouds between datasets, *e.g.*, different scene sizes or different gravity directions, are particularly challenging for a single encoder to learn. Based on this observation, we propose to leverage the *bearing vector* representation of the database points to sidestep the difference in data modalities. In addition to nullifying the effects of the camera intrinsics, projecting 3D points as bearing vectors onto a "covisible" frame that is closer to the query frame (compared to the world reference frame), effectively mitigates the influence of the camera pose (viewpoint changes) during matching, albeit dependent on the quality of retrieval. Finally, bearing vectors provide a common modality between query and database keypoints, eliminating the need for a separate encoder. As we demonstrate in our experimental section, the change in input type has a substantial positive effect.

### 4.2 Training GoMatch

All of our models are trained to learn feature matching and outlier filtering jointly, using a matching loss and an outlier rejection loss.
**Matching Loss.** The Sinkhorn matching layer is trained to output a discrete joint probability distribution of two sets of keypoints being matched. We denote this distribution as $\tilde{\mathtt{P}} \in \mathbb{R}_{+}^{M+1 \times N+1}$, such that $\sum_{i=1}^{M+1} \sum_{j=1}^{N+1} \tilde{\mathtt{P}}_{ij} = 1$, *i.e.*, is a valid probability distribution. Here, $M$ and $N$ denote the total number of query and database keypoints considered during the matching. We include an extra row

and column to allow keypoints not to be matched. We employ a negative log loss to the joint discrete probability distribution. Consider the set of all ground truth matches $\mathcal{M}$, as well as the set of unmatched query keypoints $\mathcal{U}_q$ and database keypoints $\mathcal{U}_d$. The matching loss is of the form:

$$L_{\text{match}} = -\frac{1}{N_m}\left(\sum_{(i,j)\in\mathcal{M}} \log \tilde{\mathsf{P}}_{ij} + \sum_{i\in\mathcal{U}_q} \log \tilde{\mathsf{P}}_{i(N+1)} + \sum_{j\in\mathcal{U}_d} \log \tilde{\mathsf{P}}_{(M+1)j}\right), \quad (2)$$

where $N_m = |\mathcal{M}| + |\mathcal{U}_q| + |\mathcal{U}_d|$.

**Outlier Rejection Loss.** For the outlier rejection layer we employ a mean weighted binary cross-entropy loss:

$$L_{\text{or}} = -\frac{1}{N_c}\sum_{i=1}^{N_c} w_i \left(y_i \log p_i + (1 - y_i)\log(1 - p_i)\right), \quad (3)$$

where $N_c$ denotes the total number of correspondences supplied to the outlier rejection layer. The term $p_i$ denotes the classifier output probability for each correspondence, while $y_i$ denotes the correspondence target label, and $w_i$ is the weight balancing the negative and positive samples. Our final loss balances both terms equally, *i.e.*, $L_{\text{total}} = L_{\text{match}} + L_{\text{or}}$. We present implementation details about training and testing process in our supplementary material.

## 5 Experimental Evaluation

In this section, we thoroughly study the potential of using our proposed geometric-based matching for the task of real-world visual localization. We start our experiments by testing the robustness of BPnPNet [10] and GoMatch with keypoint outliers. Next, we verify our technical contribution of successfully diagnosing the missing components leading to robust geometric matching and enabling geometry-based visual localization. Furthermore, we position gemetric-based localization among other state-of-the-art visual localization approaches by comprehensively analysing each method in terms of localization accuracy, descriptor maintenance effort [24], privacy risk, and storage demands (Section 5.4). Finally, we present a generalization study (Section 5.5) to highlight that our proposed method generalizes across different types of datasets and keypoint detectors. We hope that our in-depth study serves as a starting point of this rarely explored new direction, and inspires new work to advance scalable visual localization through geometric-only matching in the future.

### 5.1 Datasets

We use MegaDepth [39] for training and ablations, given its large scale. It consists of images captured in-the-wild from 196 outdoor landmarks. We adopt the original test set proposed in [39], and split the remaining sequences into training and validation sets. After verifying our best models on Megadepth, we evaluate them on the popular Cambridge Landmarks [35] (Cambridge) dataset which consists of 4 outdoor scenes of different scales. It allows for convenient comparison to other localization approaches. We use the reconstructions released by [52]. In addition, we evaluate on the indoor 7-Scenes [61] dataset to further assess the generalization capability of our method. 7-Scenes is composed of dense point clouds captured by an RGB-D sensor, and thus provides an alternative environment with different keypoint distributions, in both 2D images and 3D point clouds. We perform evaluation on the official test splits released by the Cambridge and 7-Scenes datasets. We provide detailed information about training data generation using MegaDepth in the supplementary.

### 5.2 Experimental Setup

**Keypoint Detection.** For MegaDepth and Cambridge, we use respectively SIFT [41] and SuperPoint [20], preserving the same keypoint detector used to reconstruct their 3D models. For 7-Scenes, we use both SIFT and SuperPoint to extract keypoints for both 2D images and 3D point cloud given RGB-D images.
**Retrieval Pairs.** We use ground truth to sample retrieval pairs that have at least 35% visual overlap in MegaDepth to ensure enough matches are present during training, as well as to isolate the side-effect of retrieval performance during ablations. For evaluation and comparison to state-of-the-art localization methods, we follow [52] and use their *top-10* pairs retrieved using NetVLAD [1] on Cambridge and DenseVLAD [68] on 7-Scenes.
**Matching Baselines.** We consider BPnPNet [10] as our geometric-based matching baseline. For a fair comparison, we re-train BPnPNet using our training data. Our visual-based matching baselines use SIFT [41] and SuperPoint [20] (SP) as keypoint descriptors. To match visual descriptors, we use nearest neighbor search [46] with mutual consistency by default and SuperGlue [51] (SG).
**Localization Pipeline.** Following the state-of-the-art structure-based localization, *e.g.*, HLoc [50], we first obtain up to $k = 10$ retrieval pairs between a query and database images. Then we establish per-pair 2D to 3D matches using either a geometric-based or a visual-based matching model, and then merge results from $k$ pairs based on their matching scores to estimate camera poses. For fairness, all matching baselines use identical retrieval pairs and identical settings for the PnP+RANSAC solver [32].
**Evaluation Metrics.** For MegaDepth, we follow BPnPNet [10] to report the pose error quantiles at 25/50/75% for the translation and rotation (°) errors as evaluation metrics. However, as the scale unit of MegaDepth is undetermined and varies between scenes, the translation errors are not consistent between scenes. Therefore, we propose a new metric based on pixel-level reprojection errors that preserves scene consistency. For each query, we project its inlier 3D keypoints using the predicted and the ground-truth poses. We then report the area under the cumulative curve (AUC) of the mean reprojection error up to 1/5/10px, inspired by the pose error based AUC metric used in [64, 52]. We report the commonly used median translation ($m$) and rotation (°) errors [35, 56, 13] per-scene on Cambridge and 7-Scenes.

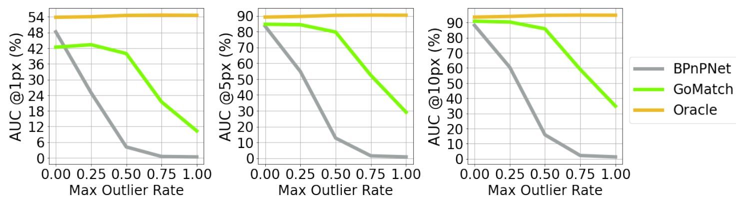


**Fig. 4.** Influence of keypoint outlier rate. In contrast to prior work BPnPnet [10], GoMatch is significantly more robust against keypoint outliers thanks to the more powerful attention-based architecture as well as our novel formulation of matching bearing vectors instead of cross-modal features.

### 5.3 Ablations

We perform ablation studies with MegaDepth's [39] test split, where all retrieval pairs have guaranteed 35% co-visibility, to focus purely on matching performance. In addition, we study the effect of using a single co-visible reference view ($k = 1$) as a minimal setting, as well as multiple views, *e.g.*, $k = 10$, following the common practice in hierarchical structure-based localization [52, 56]. To better understand the new AUC metric, we also present an **Oracle** that uses ground truth matches as its prediction. It is used to show the upper-bound performance that can be achieved using our metric and generated data.

**Sensitivity to Keypoint Outliers.** In a real-world localization setting, the detected query image keypoints will often be noisy and will not have a direct correspondence in the 3D point cloud. Keypoint matching methods thus need to be able to cope with outliers. We first study whether our baseline has this capability by manually increasing the maximum outlier rate, ranging from 0 to 1. The outlier rate is computed as the number of keypoints without a match divided by the total number of keypoints, taking the maximum between 2D and 3D. For all other experiments, we do not control keypoint the outlier rate to properly mimic realistic conditions. As shown in Fig. 4, the Oracle stays round 55/90/94% (AUC@1/5/10px). The large error at 1px is due to our match generation process (*c.f.* supplementary for a detailed discussion). BPnPNet [10] slightly outperforms GoMatch at 1px threshold, being similarly accurate to us at 5/10px thresholds in the absence of outliers. However, as the ratio of outliers increases, the performance of BPnPNet drastically drops, while GoMatch gracefully handles outliers, *i.e.*, GoMatch is always above 80% at 5/10px up to 50% of outliers. This experiment confirms that GoMatch is significantly more robust to outliers compared to BPnPNet. This outlier robustness is achieved through careful modifications to the network architecture and 3D point representation, both validated by a thorough performance analysis presented in the next sections.

**Architecture-Level Analysis.** In Table 2 (*Top*), we present the Oracle and BPnPNet [10] re-trained on our data for a direct comparison with GoMatch. This is paired with additional variants, progressively transitioning from BPnPNet to GoMatch. We found that shared encoding brings performance gains up to

**Table 2.** GoMatch ablation. *Top:* We present Oracle for reference and re-trained BPnPNet [10] as our baseline. *Middle:* We study how the 3D representation (Repr.) and architectural changes influences the performance. Using bearing vector (BVs) instead of 3D coordinates (Coords) as representation and introducing feature attention (Att) are the most crucial factors to the performance improvement. Together with further benefits from the outlier rejection (OR) component and sharing the query and database keypoint feature encoders leads us to the full GoMatch model (*Bottom*). All results rely on a singe retrieval image unless stated otherwise, *e.g.*, $k = 10$.

| Model | 3D Repr. | Share Encoder | Att | OR | Rotation (°) Quantile@25/50/75% (↓) | Translation Quantile@25/50/75% (↓) | Reproj. AUC (%) @1/5/10px (↑) |
|---|---|---|---|---|---|---|---|
| Oracle | | | | | 0.03/0.06/0.10 | 0.00/0.00/0.01 | 54.58/90.37/94.87 |
| BPnPNet | Coords | ✗ | ✗ | ✗ | 15.17/31.05/59.78 | 1.67/3.14/5.31 | 0.34/0.83/1.21 |
| BPnPNet ($k = 10$) | Coords | ✗ | ✗ | ✗ | 16.03/33.27/63.90 | 1.59/3.24/5.80 | 0.56/1.08/1.50 |
| Variants | BVs | ✗ | ✗ | ✗ | 12.19/27.68/58.22 | 1.26/2.8/5.14 | 0.37/1.48/2.18 |
| | BVs | ✓ | ✗ | ✗ | 9.16/22.62/53.20 | 0.98/2.38/4.72 | 0.85/3.09/4.36 |
| | BVs | ✓ | ✓ | ✗ | 0.55/8.08/29.34 | 0.05/0.84/3.34 | 9.13/25.71/31.65 |
| | BVs | ✗ | ✓ | ✓ | 0.38/7.46/31.75 | 0.04/0.83/3.73 | 10.22/28.17/33.69 |
| | Coords | ✗ | ✓ | ✓ | 4.09/23.56/63.21 | 0.37/2.53/5.93 | 3.81/13.54/17.46 |
| GoMatch | BVs | ✓ | ✓ | ✓ | 0.36/6.97/29.85 | 0.03/0.69/3.38 | 10.30/29.08/34.79 |
| GoMatch($k = 10$) | BVs | ✓ | ✓ | ✓ | **0.15/0.95/13.00** | **0.01/0.09/1.55** | **15.14/42.39/51.24** |

0.48/1.61/2.18 AUC percentage points. Adding feature attention on top leads to a significant improvement of 8.28/22.62/27.29 AUC percentage points. By further adding the outlier rejection increases the AUC by 1.17/3.37/3.14 percentage points. We conclude that these network components yield 9.93/27.6/32.61 percentage points of improvements in terms of AUC scores when using bearing vectors the representation.

**Representation-Level Analysis.** Using 3D coordinates (Coords) instead of bearing vectors (BVs), even with attention and outlier rejection, hinders performance dramatically by 6.49/15.54/17.33 percentage points. If we only change the representation from Coords to BVs, without attention nor outlier rejection, the improvement is merely 0.31/1.29/1.9 percentage points. Therefore, we verify the bearing vector representation is as important as the architectural changes, and both contribute towards keypoint outlier resilience. By modifying both architecture and representation, GoMatch outperforms the re-trained BPnPNet by 9.96/28.25/33.58 AUC percentage points.

**Utilizing Multiple Co-visible Images.** As shown in Table 2, when using $k = 10$ co-visible views, both methods improved their result: BPnPNet by a small margin and GoMatch by a large margin of 4.84/13.31/16.45 AUC percentage points. We thus use $k = 10$ for all of the following experiments.

### 5.4 Comparison to Localization Baselines

Following the discussion in Section 2, we comprehensively compare GoMatch with other established baselines by looking beyond localization performance, and considering as well the storage footprint, resiliency to privacy attacks, and descriptor maintenance. As shown in Table 3, HLoc with SuperPoint and Super-

**Table 3.** Comparison to existing localization baselines. We consider end-to-end (E2E) methods and structure-based methods that either matches visual descriptors (VM) or geometries (GM). We report median translation and angular error for each landmark and combined storage requirements for operating on all landmarks. *No Desc. Maint.* is checked if a method does not require descriptor updates in the long run. *Privacy* is checked if a method is resilient to existing known privacy attacks.

| | Method | Storage (MB) | No Desc. Maint. | Privacy | King's College | Old Hospital | Shop Facade | St. Mary's Church |
|---|---|---|---|---|---|---|---|---|
| | | | | | Median Pose Error (m, °) (↓) | | | |
| E2E | PoseNet [35] | 200 | ✓ | ✓ | 1.92/5.40 | 2.31/5.38 | 1.46/8.08 | 2.65/8.48 |
| | DSAC++ [6] | 828 | ✓ | ✓ | 0.18/0.30 | 0.20/0.30 | 0.06/0.30 | 0.13/0.40 |
| | MSPN [4] | - | ✓ | ✓ | 1.73/3.65 | 2.55/4.05 | 2.92/7.49 | 2.67/6.18 |
| | MS-Transformer [60] | 71.1 | ✓ | ✓ | 0.83/1.47 | 1.81/2.39 | 0.86/3.07 | 1.62/3.99 |
| VM | HybridSC [13] | 3.13 | ✗ | ? | 0.81/0.59 | 0.75/1.01 | 0.19/0.54 | 0.50/0.49 |
| | Active Search [53] | 812.7 | ✗ | ✗ | 0.42/0.55 | 0.44/1.01 | 0.12/0.40 | 0.19/0.54 |
| | HLoc [50](w.SP [20]) | 3214.84 | ✗ | ✗ | 0.16/0.38 | 0.33/1.04 | 0.07/0.54 | 0.16/0.54 |
| | HLoc(w.SP+SG [51]) | 3214.84 | ✗ | ✗ | **0.12/0.20** | **0.15/0.30** | **0.04/0.20** | **0.07/0.21** |
| GM | BPnPNet [10] | 48.15 | ✓ | ✓ | 26.73/106.99 | 24.8/162.99 | 7.53/107.17 | 11.11/49.74 |
| | GoMatch | 48.15 | ✓ | ✓ | 0.25/0.64 | 2.83/8.14 | 0.48/4.77 | 3.35/9.94 |

Glue is the most accurate method but also has the highest storage requirements while being vulnerable to privacy attacks. Using HLoc with a newly developed descriptor method will require the map to be updated. In end-to-end methods, DSAC++ is the most accurate method while being resilient to privacy attacks as it does not need to transmit visual descriptors. However, as it requires 4 model versions trained per-scene, it requires 828 MB storage to work under 4 scenes compared to our 48.12 MB. HybridSC as the most storage-efficient method keeps only 1.5% if its original points via compression. However, it is unclear whether the privacy issue still remains for this method since it still relies on full visual descriptors to perform matching. Notice, compressing scene structure can be theoretically combined with GoMatch to lower our storage requirements, which we leave as future work to design suitable scene compression techniques for geometric-base matching. On the whole, GoMatch and MS-Transformer both properly balance those three aspects showing benefits in storage, privacy and absence of descriptor maintenance, and are competitive in accuracy. Compared to its visual-descriptor SuperPoint counterpart, GoMatch requires only 1.5% of the capacity to store same scene. GoMatch reduces the average pose errors by $(10.67m, 95.7°)$ compared to our only prior geometric-based matching work, significantly reducing the accuracy gap to state-of-the-art methods. We hope this inspires researchers to pursue this line of work.

### 5.5 Generalization

As our final experiment, we study the generalization capability of our method in terms of localization in different types of scenes, *e.g.*, indoor and outdoor, and matching keypoints obtained using different detectors. According to our results in Table 4, similar to our previous expriments, we outperform BPnPNet by a large margin achieving $(1.43m, 34.7°)$ lower average median pose errors. Except

**Table 4.** Generalization study on 7-Scenes. GoMatch generalizes between different scene types and detector types and outperforming BPnPNet and PoseNet.

| | Method | Storage (MB) | No Desc. Maint. | Priv -acy | Chess | Fire | Heads | Office | Pumpkin | Kitchen | Stairs |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | Median Pose Error (m, °) (↓) | | | | | | |
| E2E | PoseNet [35] | 350 | ✓ | ✓ | 0.32/8.12 | 0.47/14.4 | 0.29/12.0 | 0.48/7.68 | 0.47/8.42 | 0.59/8.64 | 0.47/13.8 |
| E2E | DSAC++ [6] | 1449 | ✓ | ✓ | **0.02/0.50** | **0.02/0.90** | **0.01**/0.80 | **0.03/0.70** | **0.04/1.10** | **0.04/1.10** | 0.09/2.60 |
| E2E | MSPN [4] | - | ✓ | ✓ | 0.09/4.76 | 0.29/10.5 | 0.16/13.1 | 0.16/6.8 | 0.19/5.5 | 0.21/6.61 | 0.31/11.63 |
| E2E | MS-Transformer [60] | 71.1 | ✓ | ✓ | 0.11/4.66 | 0.24/9.6 | 0.14/12.19 | 0.17/5.66 | 0.18/4.44 | 0.17/5.94 | 0.26/8.45 |
| VM | Active Search [53] | - | ✗ | ✗ | 0.04/1.96 | 0.03/1.53 | 0.02/1.45 | 0.09/3.61 | 0.08/3.10 | 0.07/3.37 | **0.03**/2.22 |
| VM | HLoc [50](w.SIFT [41]) | 2923 | ✗ | ✗ | 0.03/1.13 | 0.03/1.08 | 0.02/2.19 | 0.05/1.42 | 0.07/1.80 | 0.06/1.84 | 0.18/4.41 |
| VM | HLoc(w.SP [20]) | 22977 | ✗ | ✗ | 0.03/1.28 | 0.03/1.3 | 0.02/1.99 | 0.04/1.31 | 0.06/1.63 | 0.06/1.73 | 0.07/1.91 |
| VM | HLoc(w.SP+SG [51]) | 22977 | ✗ | ✗ | **0.02**/0.85 | **0.02**/0.94 | **0.01/0.75** | **0.03**/0.92 | 0.05/1.30 | **0.04**/1.40 | 0.05/**1.47** |
| GM | BPnPNet [10](SIFT [41]) | 302 | ✓ | ✓ | 1.29/43.82 | 1.48/51.82 | 0.93/55.13 | 2.61/59.06 | 2.15/39.85 | 2.15/43.00 | 2.98/60.27 |
| GM | BPnPNet (SP [20]) | 397 | ✓ | ✓ | 1.25/43.9 | 1.42/45.09 | 0.8/50.05 | 2.33/14.54 | 1.71/31.81 | 1.68/33.91 | 2.1/55.78 |
| GM | GoMatch (SIFT) | 302 | ✓ | ✓ | 0.04/1.65 | 0.13/3.86 | 0.09/5.17 | 0.11/2.48 | 0.16/3.32 | 0.13/2.84 | 0.89/21.12 |
| GM | GoMatch (SP) | 397 | ✓ | ✓ | 0.04/1.56 | 0.12/3.71 | 0.05/3.43 | 0.07/1.76 | 0.28/5.65 | 0.14/3.03 | 0.58/13.12 |

for GoMatch with SIFT keypoints which produces a relatively large 21.12° median rotation error in Stairs, we are only slightly worse than our visual-based matching baselines with SIFT and SuperPoint. Yet, we require only 10/1.7% of the storage that is required by SIFT/SuperPoint to store maps. We also largely outperform PoseNet [35] in all metrics for all scenes except for the relatively lower translation error in Stairs scene, *i.e.*, (0.47m vs 0.58m). Furthermore, we achieve better pose than MS-Transformer in the majority of scenes, at the expense of a higher storage requirement. The results clearly verify that GoMatch trained on outdoor scenes (MegaDepth) generalizes smoothly to indoor scenes (7-Scenes), being agnostic to scene types. Similarly, we also confirm that GoMatch trained with SIFT keypoints generalizes well to SuperPoint keypoints, being agnostic to detector types.

## 6 Conclusion

We present GoMatch, a novel sparse keypoint matching method for visual localization that relies only on geometrical information and that carefully balances common practical challenges of large-scale localization, namely: localization performance, storage demands, privacy and descriptor maintenance (or lack thereof). From all these, the last three are often overlooked. Through a rigorous architecture design process, GoMatch dramatically surpasses its prior work in handling outliers, enabling it for real-world localization. Compared to localization pipelines using visual descriptor-based matching, GoMatch allows localization with a minimal 3D scene representation, requiring as little as 1.5/1.7% to store the same scene. Geometric-based matching brings localization pipelines to a new level of scalability that opens the door for localizing in much larger environments. We see our work as a starting point for this new direction and we look forward to inspire other researchers to pursue more accurate and reliable geometric-based visual localization in the future.

**Acknowledgments.** This research was partially funded by the Humboldt Foundation through the Sofja Kovalevskaya Award.

## References


1. Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
2. Vassileios Balntas, Shuda Li, and Victor Prisacariu. Relocnet: Continuous metric learning relocalisation using neural nets. In *European Conference on Computer Vision (ECCV)*, September 2018.
3. Aritra Bhowmik, Stefan Gumhold, Carsten Rother, and Eric Brachmann. Reinforced feature points: Optimizing feature detection and description for a high-level task. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4948–4957, 2020.
4. Hunter Blanton, Connor Greenwell, Scott Workman, and Nathan Jacobs. Extending absolute pose regression to multiple scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.
5. Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. DSAC - Differentiable RANSAC for Camera Localization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
6. Eric Brachmann and Carsten Rother. Learning less is more - 6d camera localization via 3d surface regression. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
7. Eric Brachmann and Carsten Rother. Expert sample consensus applied to camera re-localization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7525–7534, 2019.
8. Eric Brachmann and Carsten Rother. Neural-guided ransac: Learning where to sample model hypotheses. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4322–4331, 2019.
9. Mark Brown, David Windridge, and Jean-Yves Guillemaut. Globally optimal 2d-3d registration from points or lines without correspondences. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.
10. Dylan Campbell, Liu Liu, and Stephen Gould. Solving the blind perspective-n-point problem end-to-end with robust differentiable geometric optimization. In *European Conference on Computer Vision (ECCV)*, pages 244–261. Springer, 2020.
11. Dylan Campbell, Lars Petersson, Laurent Kneip, and Hongdong Li. Globally-optimal inlier set maximisation for simultaneous camera pose and feature correspondence. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.
12. Dylan Campbell, Lars Petersson, Laurent Kneip, Hongdong Li, and Stephen Gould. The alignment of the spheres: Globally-optimal spherical mixture alignment for camera pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
13. 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 (CVPR)*, June 2019.
14. 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 (CVPR)*, June 2014.
15. Tommaso Cavallari, Luca Bertinetto, Jishnu Mukhoti, Philip Torr, and Stuart Golodetz. Let's take this online: Adapting scene coordinate regression network

predictions for online rgb-d camera relocalisation. In *2019 International Conference on 3D Vision (3DV)*, pages 564–573, 2019.
16. Kunal Chelani, Fredrik Kahl, and Torsten Sattler. How privacy-preserving are line clouds? recovering scene details from 3d lines. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15668–15678, 2021.
17. Wentao Cheng, Weisi Lin, Kan Chen, and Xinfeng Zhang. Cascaded parallel filtering for memory-efficient image-based localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.
18. Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.
19. Philip David, Daniel Dementhon, Ramani Duraiswami, and Hanan Samet. Softposit: Simultaneous pose and correspondence determination. *International Journal of Computer Vision*, 59(3):259–284, 2004.
20. Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPR Workshops*, pages 224–236, 2018.
21. Mingyu Ding, Zhe Wang, Jiankai Sun, Jianping Shi, and Ping Luo. Camnet: Coarse-to-fine retrieval for camera re-localization. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2871–2880, 2019.
22. Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.
23. Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4829–4837, 2016.
24. Mihai Dusmanu, Ondrej Miksik, Johannes L Schonberger, and Marc Pollefeys. Cross-descriptor visual localization and mapping. In *IEEE International Conference on Computer Vision (ICCV)*, pages 6058–6067, 2021.
25. Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-Net: A Trainable CNN for Joint Detection and Description of Local Features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
26. Mihai Dusmanu, Johannes L Schönberger, Sudipta N Sinha, and Marc Pollefeys. Privacy-preserving image features via adversarial affine subspace embeddings. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
27. 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.
28. Marcel Geppert, Viktor Larsson, Pablo Speciale, Johannes L Schönberger, and Marc Pollefeys. Privacy preserving structure-from-motion. In *European Conference on Computer Vision (ECCV)*, pages 333–350. Springer, 2020.
29. Marcel Geppert, Viktor Larsson, Pablo Speciale, Johannes L Schonberger, and Marc Pollefeys. Privacy preserving localization and mapping from uncalibrated cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1809–1819, 2021.
30. Hugo Germain, Guillaume Bourmaud, and Vincent Lepetit. S2dnet: Learning accurate correspondences for sparse-to-dense feature matching. *European Conference on Computer Vision (ECCV)*, 2020.

31. Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. Predator: Registration of 3d point clouds with low overlap. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4267–4276, June 2021.
32. Tong Ke and Stergios I. Roumeliotis. An efficient algebraic solution to the perspective-three-point problem. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
33. Alex Kendall and Roberto Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.
34. Alex Kendall and Roberto Cipolla. Geometric Loss Functions for Camera Pose Regression With Deep Learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
35. Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.
36. Laurent Kneip, Davide Scaramuzza, and Roland Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.
37. Zakaria Laskar, Iaroslav Melekhov, Surya Kalia, and Juho Kannala. Camera Relocalization by Computing Pairwise Relative Poses Using Convolutional Neural Network. In *IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.
38. Xiaotian Li, Shuzhe Wang, Yi Zhao, Jakob Verbeek, and Juho Kannala. Hierarchical scene coordinate classification and regression for visual localization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11983–11992, 2020.
39. Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
40. Liu Liu, Dylan Campbell, Hongdong Li, Dingfu Zhou, Xibin Song, and Ruigang Yang. Learning 2d-3d correspondences to solve the blind perspective-n-point problem, 2020.
41. David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, Nov 2004.
42. Zixin Luo, Lei Zhou, Xuyang Bai, Hongkai Chen, Jiahui Zhang, Yao Yao, Shiwei Li, Tian Fang, and Long Quan. Aslfeat: Learning local features of accurate shape and localization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6589–6598, 2020.
43. Marcela Mera-Trujillo, Benjamin Smith, and Victor Fragoso. Efficient scene compression for visual-based localization. In *2020 International Conference on 3D Vision (3DV)*, pages 1–10, 2020.
44. Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to find good correspondences. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2666–2674, 2018.
45. Francesc Moreno-Noguer, Vincent Lepetit, and Pascal Fua. Pose priors for simultaneously solving alignment and correspondence. In *European Conference on Computer Vision (ECCV)*, pages 405–418. Springer, 2008.
46. Marius Muja and David G. Lowe. Scalable nearest neighbor algorithms for high dimensional data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2227–2240, 2014.

47. Tony Ng, Hyo Jin Kim, Vincent T Lee, Daniel DeTone, Tsun-Yi Yang, Tianwei Shen, Eddy Ilg, Vassileios Balntas, Krystian Mikolajczyk, and Chris Sweeney. Ninjadesc: Content-concealing visual descriptors via adversarial learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12797–12807, 2022.
48. Francesco Pittaluga, Sanjeev J Koppal, Sing Bing Kang, and Sudipta N Sinha. Revealing scenes by inverting structure from motion reconstructions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 145–154, 2019.
49. Noha Radwan, Abhinav Valada, and Wolfram Burgard. Vlocnet++: Deep multitask learning for semantic visual localization and odometry. *IEEE Robotics and Automation Letters*, 3(4):4407–4414, 2018.
50. Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
51. Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4938–4947, 2020.
52. 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3247–3257, 2021.
53. 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, 2017.
54. Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, et al. Benchmarking 6dof outdoor visual localization in changing conditions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8601–8610, 2018.
55. 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
56. Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixe. Understanding the limitations of cnn-based absolute camera pose regression. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
57. Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
58. Johannes L Schönberger, Marc Pollefeys, Andreas Geiger, and Torsten Sattler. Semantic Visual Localization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
59. Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, pages 501–518. Springer, 2016.
60. 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 (ICCV)*, pages 2733–2742, October 2021.

61. 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2930–2937, 2013.
62. Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967.
63. Pablo Speciale, Johannes L Schonberger, Sing Bing Kang, Sudipta N Sinha, and Marc Pollefeys. Privacy preserving image-based localization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5493–5503, 2019.
64. Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8922–8931, 2021.
65. Weiwei Sun, Wei Jiang, Eduard Trulls, Andrea Tagliasacchi, and Kwang Moo Yi. Acne: Attentive context normalization for robust permutation-equivariant learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
66. Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. InLoc: Indoor Visual Localization with Dense Matching and View Synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
67. Carl Toft, Will Maddern, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, Fredrik Kahl, and Torsten Sattler. Long-term visual localization revisited. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(4):2074–2088, 2022.
68. Akihiko Torii, Relja Arandjelovic, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 place recognition by view synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1808–1817, 2015.
69. Ngoc-Trung Tran, Dang-Khoa Le Tan, Anh-Dzung Doan, Thanh-Toan Do, Tuan-Anh Bui, Mengxuan Tan, and Ngai-Man Cheung. On-device scalable image-based localization via prioritized cascade search and fast one-many ransac. *IEEE Transactions on Image Processing*, 28(4):1675–1690, 2019.
70. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.
71. 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 International Conference on Computer Vision (ICCV)*, 2017.
72. Qianqian Wang, Xiaowei Zhou, Bharath Hariharan, and Noah Snavely. Learning feature descriptors using camera pose supervision. In *European Conference on Computer Vision (ECCV)*, 2020.
73. Luwei Yang, Ziqian Bai, Chengzhou Tang, Honghua Li, Yasutaka Furukawa, and Ping Tan. Sanet: Scene agnostic network for camera localization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 42–51, 2019.
74. 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 (ICCV)*, October 2019.

75. Qunjie Zhou, Torsten Sattler, Marc Pollefeys, and Laura Leal-Taixe. To learn or not to learn: Visual localization from essential matrices. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3319–3326. IEEE, 2020.