Efficient Large-scale Localization by Global Instance Recognition

Fei Xue† Ignas Budvytis† Daniel Olmeda Reino‡ Roberto Cipolla†
†University of Cambridge ‡Toyota Motor Europe
{fx221, ib255, rc10001}@cam.ac.uk daniel.olmeda.reino@toyota-europe.com

Abstract

Hierarchical frameworks consisting of both coarse and fine localization are often used as the standard pipeline for large-scale visual localization. Despite their promising performance in simple environments, they still suffer from low efficiency and accuracy in large-scale scenes, especially under challenging conditions. In this paper, we propose an efficient and accurate large-scale localization framework based on the recognition of buildings, which are not only discriminative for coarse localization but also robust for fine localization. Specifically, we assign each building instance a global ID and perform pixel-wise recognition of these global instances in the localization process. For coarse localization, we employ an efficient reference search strategy to find candidates progressively from the local map observing recognized instances instead of the whole database. For fine localization, predicted labels are further used for instance-wise feature detection and matching, allowing our model to focus on fewer but more robust keypoints for establishing correspondences. The experiments in long-term large-scale localization datasets including Aachen and RobotCar-Seasons demonstrate that our method outperforms previous approaches consistently in terms of both efficiency and accuracy.

1. Introduction

Visual localization is a key technique of various applications, *e.g.*, autonomous driving and robotics. Visual localization algorithms can be roughly categorized as image-based [21, 64], scene coordinate-based [4–7], and structure-based [37, 40, 47, 56]. Image-based methods only yield approximate poses [2, 15, 34, 43] and scene coordinate-based models don't perform well in large-scale scenes [23]. Structure-based systems consisting of coarse (finding reference images in the database via image retrieval [2, 15, 58]) and fine localization (establishing correspondences between the query and reference images by keypoint matching [10, 27, 35]), are preferred in real applications.

In the pipeline of structure-based systems, both coarse and fine localization are formulated as finding the closest candidates from a given set for the query data point, *e.g.*, image to image matches in coarse localization and point to point matches in fine localization. Previous methods [10, 35, 37] employ exhaustive comparisons for all query data with the rest in the database. This, however, is computationally slow and suffers from low accuracy because of spurious wrong candidates, especially under challenging conditions, *e.g.*, changes of illumination, season, and weather. Some works [23, 47, 63] make use of semantics to improve both coarse and fine localization separately. For coarse localization, they filter unstable objects like trees [46, 63] or transfer images from night to day [1]. For fine localization, additional segmentation networks are incorporated to reject semantically inconsistent matches [18, 22, 23, 45, 47, 56]. Despite their promising improvements, modeling coarse and fine localization as two independent tasks, they ignore the fact that coarse localization should provide reference images with enough *valid*

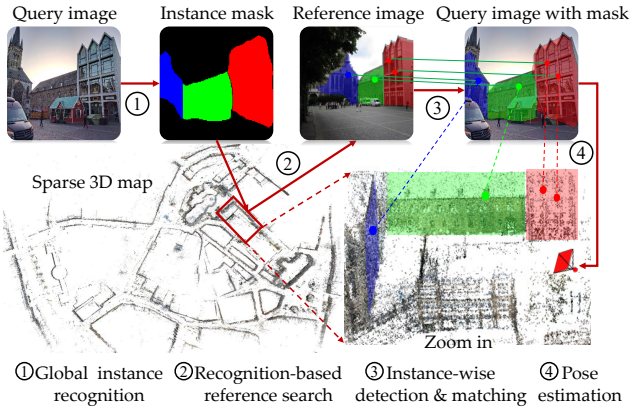


Figure 1. **Overview of our framework.** For each query image, we first perform pixel-wise global instance recognition, results of which are then used to find references from areas observing recognized instances instead of the whole database. Pixel-wise recognition masks are also used for instance-wise feature detection and matching to provide robust correspondences for fine localization. Finally, 2D-2D matches between the query and reference images are converted to 2D-3D matches for pose estimation.

areas for establishing correspondences in fine localization rather than the *most similar* images in the database. Moreover, most of which make explicit use of semantic labels [23, 47, 56] are not robust to segmentation failures.

In this paper, we aim to design an efficient and accurate large-scale localization system by modeling coarse and fine localization as a coherent process. To this end, we leverage buildings to bridge the gap between the two processes. Compared with other objects (*e.g.*, trees and cars), which are sensitive to appearance changes, buildings are able to provide robust correspondences for fine localization. Besides, buildings are also discriminative to represent a location for coarse localization. We make use of the robustness and discriminative ability of buildings in a coherent manner and propose a recognition-based localization system. Specifically, we first assign each building instance a global ID. Next, for each image, we perform pixel-wise recognition of global building instances, results of which are then utilized to find reference images from areas observing recognized instances rather than the whole map. Finally, pixel-wise recognition masks are further used for local feature detection and matching, allowing our model to extract fewer but more robust keypoints and execute instance-wise matching in a reduced space to increase the number of inliers.

Benefiting from the uniqueness of buildings and their robustness to appearance changes, in comparison to *general objects*, our model can recognize more building instances even under challenging conditions. To minimize the influence of potential recognition errors, we employ a progressive search strategy to efficiently find references for coarse localization and a robust instance-wise detection and matching approach for fine localization. We also divide the pose estimation process into two steps so that our model is able to explore more potential locations at low cost in the first step and perform a slower refinement to produce more accurate poses in the second step. Fig. 1 shows an overview of our framework. Contributions are summarized as follows:

- We propose a novel localization framework based on global building instance recognition, which models coarse and fine localization as a coherent process.
- We employ a progressive recognition-based reference search strategy to efficiently find candidates from local areas instead of the whole database.
- We leverage a robust instance-wise detection and matching technique to obtain better accuracy with fewer keypoints even under challenging conditions.

Results on long-term large-scale Aachen and RobotCar-Seasons datasets [30, 41, 42] demonstrate that our model outperforms previous approaches in terms of both efficiency and accuracy. We organize the rest of this paper as follows. In Sec. 2, we introduce related works on visual localization. In Sec. 3, our framework is described in detail. We conduct extensive experiments in Sec. 4. The limitations and conclusions are discussed in Sec. 5 and 6, respectively.

2. Related Works

In this section, we discuss works related to visual localization and instance recognition.

2.1. Visual Localization

Visual localization. Visual localization systems can be roughly categorized into three groups: image-based, scene coordinate-based and structure-based. Image-based approaches [2, 15, 58] estimate the pose of a query image by finding the most similar one in the database. As images are sparse in the database, only approximate poses can be obtained. Image-based works can be extended by regressing the pose directly with a neural network [21, 64]. However, they perform closely to image retrieval methods [43]. Instead of predicting the pose directly, scene coordinate-based methods [4–7] first predict 3D coordinates, then estimate the pose with the perspective-n-point (PnP) [24] technique. They report outstanding performance in small-scale scenes, yet struggle to give comparable results in large-scale environments [23]. Structure-based methods [9, 37, 40, 51] consist of processes of mapping and localization. In the mapping process, structure from motion (SfM) techniques [44] are used to build a sparse 3D map. In the localization process [37], image retrieval [2, 15, 34] is first used to find reference images. Next, local keypoints are utilized to build correspondences between the query and reference images, which are then fed into the PnP module for pose estimation.

With an explicit 3D map, structured-based methods are able to produce more accurate poses in large-scale scenes. However, the global reference search of comparing a query image with all in the database [2, 15, 34, 37] and exhaustive feature detection and matching [10, 29, 35] are inefficient. Besides, they suffer from low accuracy and robustness under challenging conditions, *e.g.*, day-night changes and seasonal variations. These problems can be partially solved by leveraging more powerful local features [3, 11, 29, 35, 60, 61] or matching networks [13, 14, 25, 36, 38, 49, 65, 67], while their high computational [35, 38] and memory [25, 36, 49] cost impair their efficiency in real applications.

Visual semantic localization. Another solution to aforementioned problems is introducing high-level semantics. For coarse localization, most of them learn image-level description by focusing on robust regions [18, 32, 46, 57, 63] or perform domain transformation [1, 19]. While they achieve better performance, their accuracy and efficiency are still limited by the use of global reference search. For fine localization, an additional segmentation network is usually incorporated into the localization pipeline to provide semantic labels for filtering semantically-inconsistent

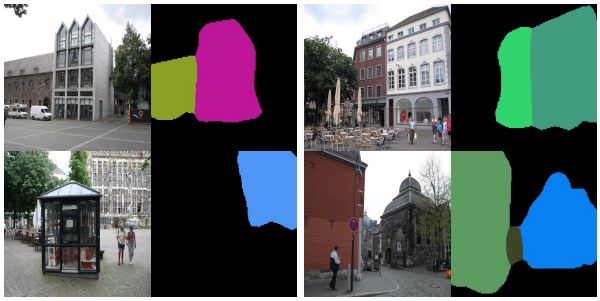


Figure 2. **Examples of automatically generated global building instances.** Different colors indicate different global instances.

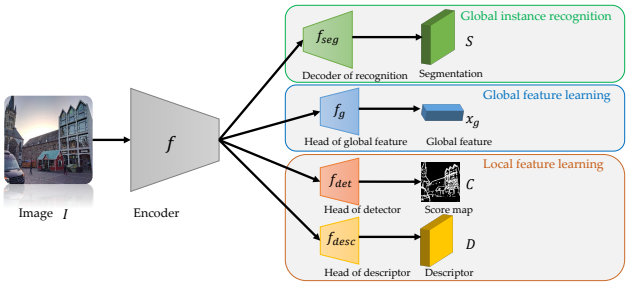


Figure 3. **Architecture of our network.** Our network consists of an encoder and three decoders for global instance recognition, global feature learning, and local feature learning.

matches [23,47,56]. Despite their promising improvements, they are vulnerable to segmentation errors especially under challenging conditions.

Unlike these methods, for coarse localization, our recognition-based framework finds references from areas observing recognized global instances, making the *global* search become *local* search, so as to obtain higher efficiency and accuracy due to reduced searching space. For fine localization, the instance-wise feature detection and matching enforce our model to focus on robust building instances, leading to better performance with fewer keypoints. During both the detection and matching processes, we take into account the potential recognition errors, allowing our model to give robust results even when recognition fails.

2.2. Instance recognition

Recently, landmark recognition, as a sub-task of place recognition, has become increasingly popular with a large number of methods [52, 53, 57] and datasets [62] proposed. Landmark recognition [53, 57, 62] differs with our global building instance recognition in two aspects. First, landmarks are defined on a whole building, while our global building instances are defined on building facades to provide more precise locations, *e.g.*, a building may have several facades indicating different locations. Second, landmark recognition is an image-level classification task, while our global instance recognition executes pixel-wise recognition to provide pixel-wise labels for fine localization.

Some works also leverage the instances of buildings [6, 7] or clusters of general objects [26] to perform hierarchical scene coordinate regression. In spite of their promising accuracy in simple and small-scale scenarios, as other scene coordinate-based methods [4, 5], they struggle to give comparable accuracy in large-scale environments and fail to deal with classification errors caused by challenging conditions.

3. Localization by Recognition

In this section, we give details of the definition of global building instances as well as the training and testing processes of our network.

3.1. Global instance definition

Global building instances are obtained by assigning each building facade a unique ID as long as it has obvious differences with neighbor ones in terms of textures, shapes, or ornaments, as shown in Fig. 2. To increase the practicability of our method, we propose an automatic global instance annotation strategy to remove the need of costly manual labeling. The details of automatic annotation process are provided in the **supplementary material**.

3.2. Network and training details

As shown in Fig. 3, our network consists of four modules responsible for feature extraction, global instance recognition, global feature learning, and local feature learning, respectively. The shared encoder f extracts high-level features X from the input image $I \in R^{3\times H\times W}$, as $X = f(I)$. H and W are the height and width of the input image and $X = \{X_1, X_2, X_3, X_4\}$ is a set of predicted multi-scale features with $2\times, 4\times, 8\times$, and $16\times$ downsampling.

Global instance recognition. We utilize the aggregation component of PSPNet [66] to generate context features, which are used as input of a segmentation head to produce pixel-wise classification. The whole process is denoted as f_{seg}. To boost the performance on hard cases, we adopt the cross entropy with online hard example mining (ohem) f_{ce}^{ohem} [48] loss between predicted $S \in R^{N\times H\times W} = f_{seg}(X)$ and ground-truth labels S_{gt} (N is the number of global instances):

$$L_{seg} = f_{ce}^{ohem}(f_{seg}(X), S_{gt}). \quad (1)$$

Global feature learning. Predicted global instance labels offer us potential areas in the map where to search references, but we still need an image-level representation to find the most suitable reference image from candidates observing the same labels. Therefore, we introduce an additional pooling layer f_g, transferring X_4 features to a global description x_g. Benefiting from the recognition task,

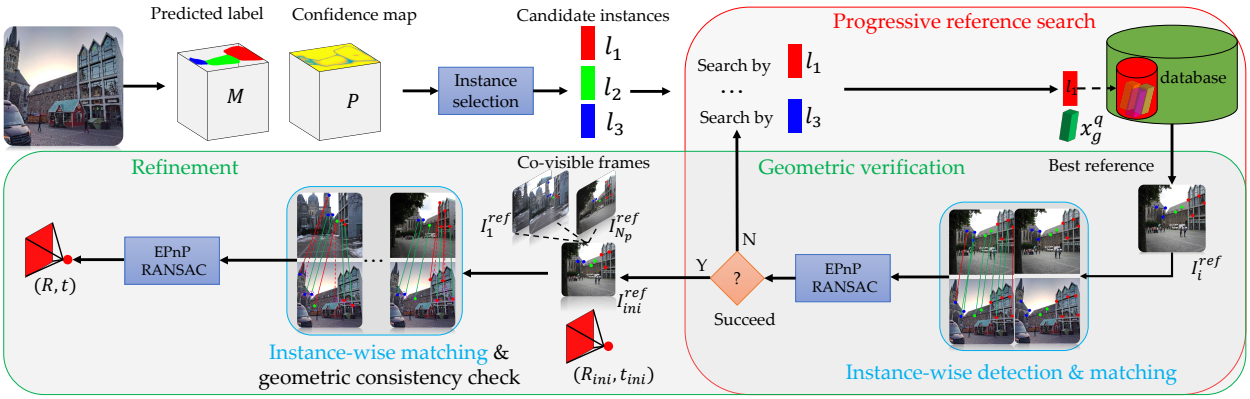


Figure 4. **Pipeline of recognition-based localization.** We first select and sort predicted instance labels according to their confidences. Then, a progressive reference search technique incorporated geometric verification is adopted to find accurate coarse locations efficiently. Finally, a further pose refinement step is executed to refine initially recovered pose by introducing more accurate reference images. Predicted instance labels are utilized for both feature detection and matching to enhance the accuracy of correspondences.

the feature X_4 with instance information embedded, can be used to generate the global feature without slow and memory-consuming layers like VLAD [2, 15, 58].

We use the output of f_g to optimize the rank of positive (with over 200 correspondences) and negative samples with the average precision (ap) loss [16]. To further enhance the discriminative ability of x_g, we take inspirations from the landmark classification task [8] and decode the instance information from x_g with an additional fully-connected layer f_{cls} to predict binary vector of existing labels $x_g^{cls} \in \{0,1\}^N$. The combined loss is defined as:

$$L_{global} = \omega_{ap} ap(x_g) + \omega_{cls} BCE(f_{cls}(x_g), y_{cls}). \quad (2)$$

BCE is the binary cross entropy. y_{cls} is the ground-truth binary indication of existing labels. ω_{ap} and ω_{cls} are two parameters balancing the ap and classification losses.

Local feature learning. High-resolution feature $X_2 \in R^{128\times\frac{H}{4}\times\frac{W}{4}}$ is first fed into several convolutional layers to get projected features $X_{local} \in R^{128\times\frac{H}{4}\times\frac{W}{4}}$, which is used as the input of both the detector head f_{det} and descriptor head f_{desc} for producing the score map $C \in R^{H\times W}$ and descriptor map $D \in R^{128\times\frac{H}{4}\times\frac{W}{4}}$, respectively.

The detector head is trained with the confidence map C_{spp} predicted by Superpoint (SPP) [10] as supervision signals. SPP is trained with synthetic geometric shapes with ground-truth corners, so it is very good at detecting corners. For the descriptor, we adopt a triplet loss to minimize and maximize the distance between positive and negative samples, respectively. The combined detection L_{det} and description loss L_{desc} is defined, as:

$$L_{local} = \omega_{det} BCE(C, C_{spp}) + \omega_{desc} \frac{1}{N_l} \sum_{x_l} tri(x_l, x_l^p, x_l^n, m). \quad (3)$$

x_l, x_l^p, and x_l^n are descriptors of the query, positive, and negative samples. N_l is the number of local features. tri is the triplet loss with margin m. ω_{det} and ω_{desc} are two parameters balancing the detection and description losses.

3.3. Recognition-based localization

In Fig. 4, we give a full description of how our recognition-based system works at test time.

Efficient progressive reference search. For each query image, we predict the global instance label for each pixel S_{ij}. Considering the recognition uncertainty, for each pixel, we keep top K predictions with the highest confidence to get recognition mask $M \in N^{K\times H\times W}$. A simple strategy could be to find the closest reference images from images observing all potential global instance labels in M. Although this is able to reduce the search time partially by filtering unrelated areas, it is still time-consuming and sensitive to recognition errors.

Instead of extracting all potential global instances from M directly, we make use of the confidence map $P \in R^{K\times H\times W}$, which tells us the probability of labels observed by the query image, and propose a progressive search strategy. For all potential instances l_i extracted from M, we first calculate their confidence by averaging values of all corresponding pixels in P. These instances are then sorted according to their confidence from high to low to form a list $\{l_1, l_2, ..., l_{N_r}\}$ (N_r is the number of recognized instances). Finally, we search references by comparing the L_2 distance between global features of the query x_g^q and candidate images observing l_i in local areas. Geometric verification is adopted between the query and the best reference image I_i^{ref} to check if l_i is a correct instance label. If this step fails, we try the next recognized instance l_{i+1} until finding the correct one and output the corresponding reference im-

age I_{ini}^{ref} and recovered pose $T_{ini} = (R_{ini}, t_{ini})$ for further refinement. More details are provided in the **Fast two-step pose estimation** section.

Robust instance-wise feature detection and matching. The pixel-wise prediction on building instances can be further leveraged as priors for both feature detection and matching. We perform a progressive detection and matching strategy to take full advantage of global instance recognition in a way which is robust to segmentation errors.

Given the predicted local feature score map C, we first select keypoints with score value larger than threshold λ to discard locally unreliable ones. Next, from selected keypoints, we retain N_{kpt} keypoints $\{p_1, p_2, ..., p_{N_{kpt}}\}$ with labels of $\{l_1^p, l_2^p, ..., l_{N_{kpt}}^p\}$. When recognized areas are unable to provide sufficient keypoints due to viewpoint changes and occlusions, we also select keypoints with the highest scores from the *background* and assign them with label 0. In the matching process, given two sets of keypoints extracted from the query and reference images, we first conduct instance-wise matching between those with the same *valid* labels independently, resulting in more accurate correspondences by reducing the distraction of other instances. Since instance-wise matching is operated for subsets with different labels individually, it can be executed in parallel for a speed up. For unmatched keypoints and those with label of 0, we further perform exhaustive matching to improve the robustness to recognition errors or occlusions.

In our instance-wise detection and matching processes, we preferentially utilize the keypoints from correctly recognized areas, enabling our system to obtain as many inliers as possible, which is especially important when only a limited number of keypoints can be used.

Fast two-step pose estimation. 2D-2D correspondences between the query and reference images can be converted to 2D-3D matches between the query image and map, which are then fed into the EPnP+RANSAC [12, 24] module for pose estimation. Unlike HLoc [37] which performs exhaustive matching between the query and all references images and takes all matches as the input of EPnP+RANSAC for pose estimation, we divide the pose estimation into two steps: **geometric verification** and **refinement**. As mentioned in the **progressive reference search** process, geometric verification operated on the query and a single reference image helps to obtain the correct reference image I_{ini}^{ref} and initially estimated pose T_{ini} as well.

Owing to the limited number of views and correspondences, pose T_{ini} is not very accurate. To yield a more precise pose estimate, we use inlier 3D points to find top N_p images $\{I_1^{ref}, ..., I_{N_p}^{ref}\}$ with the highest number of covisible points with I_{ini}^{ref}. Then, we conduct instance-wise matching between the query and newly gained set of reference images to get more matches. T_{ini} plays the role of an advanced matcher [25, 31, 36, 38, 59] to reject 2D-3D matches with reprojection error larger than threshold η. The two-step pose estimation is designed to obtain more accurate poses with lower cost and is flexible to be applied to other frameworks such as HLoc [37].

4. Experiments

We first give details of implementation as well as baselines and metrics, and datasets used for evaluation. Next, we compare our model with previous state-of-the-art methods on the large-scale localization task in Sec. 4.1 and 4.2. We discuss the running time and ablation study in Sec. 4.3 and 4.4, respectively. More implementation details, results, and analysis can be found in the **supplementary material**.

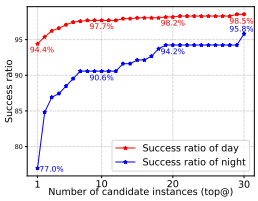
Implementation. We adopt ResNet101 [17] and PSPNet [66] as the encoder and decoder for recognition, respectively. In the training process, ω_{ap}, ω_{cls}, ω_{det}, ω_{desc} are set to 1.0, 2.0, 1.0, and 1.0. While in the localization process, K, N_r, N_p, and η are set to 10, 30, 50, and 20. We use only 4,096 kypoints (N_{kpt}) for all experiments.

Baselines and metrics. For coarse localization, we compare our system with image retrieval methods [2, 15, 34] in terms of efficiency. For fine localization, we compare it with image-based [2, 15, 34] (R), classic structure-based [9, 40, 51] (C), and semantic-based methods [26, 47, 56, 63] (S). We also compare it with state-of-the-art pipeline HLoc [37] with different local features [10, 11, 28, 33, 35, 50, 61] (H) and those with advanced or dense matching networks [13, 31, 38, 39, 49, 59, 67] (M). For fine localization, we adopt the success ratio with different error thresholds ($0.25m/2°$, $0.5m/5°$, $5m/10°$), as in [37, 41].

Datasets. We test our system on public large-scale localization datasets including Aachen [42], Aachen_v1.1 [42], and RobotCar-Seasons (RoboCS) [30]. Aachen contains 4,328 reference and 922 (824 day, 98 night) query images captured with handheld cameras around the Aachen city at different seasons with various illumination conditions. Aachen_v1.1 is extended from Aachen dataset by adding 2,369 reference and 93 night query images. RoboCS dataset was collected by a moving car running around the Oxford at different seasons, illumination and weather conditions. It has 26,121 reference and 11,934 query images recorded by three mounted cameras (left, right, rear), while only the rear camera is used. Since only day images are available in the database and extreme changes of season, weather, illumination, and dynamic objects exist in query images, these datasets are challenging for both recognition and localization. Moreover, huge variations of viewpoint in Aachen/Aachen_v1.1 dataset and motion blur/overexposure in RoboCS dataset further increase the difficulty.

4.1. Reference search

We visualize the number of reference images and success ratio of progressive reference search in Fig. 5. Bene-



Method	Avg. Search Frame day	night	Complexity
IR	6697	6697	$O(n)$
Ours	202	650	$O(N_{rec}N_{obs})$

Figure 5. **Results of reference search.** The success ratio and the number of search frames of image retrieval (IR) [2, 15, 34] and our method on on Aachen_v1.1 dataset [42] are reported.

Group	Method	Day	Night
H	SIFT [27]	72.2 / 78.4 / 81.7	19.4 / 23.0 / 27.2
	SPP [10]	87.9 / 93.6 / 96.8	70.2 / 84.8 / 93.7
	D2Net [11]	84.1 / 91.0 / 95.5	63.4 / 83.8 / 92.1
	R2D2 [35]	88.8 / 95.3 / 97.8	72.3 / 88.5 / 94.2
	ASLFeat [29]	88.0 / 95.4 / 98.2	70.7 / 84.3 / 94.2
	CAPS + SIFT [27, 61]	82.4 / 91.3 / 95.9	61.3 / 83.8 / 95.3
	LISRD + SPP [10, 33]		73.3 / 86.9 / 97.9
	LLF + R2D2 [10, 50]		71.2 / 81.2 / 94.2
M	SPP + Superglue [10, 38]	**89.8 / 96.1 / 99.4**	77.0 / **90.6** / **100.0**
	Patch2Pix [67]	86.4 / 93.0 / 97.5	72.3 / 88.5 / 97.9
	LoFTER [49]	88.7 / 95.6 / 99.0	**78.5** / **90.6** / 99.0
	Ours	89.1 / **96.1** / 99.3	77.0 / 90.1 / 99.5

Table 1. **Results on Aachen_v1.1 dataset.** The best and second best results are highlighted with **bold** and red fonts.

fiting from global instance recognition, the average number of images in a search is 80 for approximate 94% day and 77% night query images, which is 80 times smaller than global search [2, 15, 34]. For each query image, we keep 30 predicted instance labels, allowing us to achieve 99% and 96% success ratio for day and night images, respectively. Thanks to our progressive search strategy, the average number of search frames for day and night images are 202 and 650, which are 33 and 10 times fewer than global search.

Since retrieval-based methods conduct global search for each query image, their complexity is $O(n)$ (n is the number of images in the database). However, the complexity of recognition-based search strategy is only $N_{rec}N_{obs}$ (N_{rec} and N_{obs} are the number of retained instances and observations for each global instance in the database). Such kind of fast search is extremely important to real-time applications.

4.2. Fine localization

Results on Aachen dataset. Table 2 shows the results on Aachen dataset [42]. We can see that retrieval-based approaches (R) only provide approximate poses and Netvlad [2] is still state-of-the-art compared with more recent AP-GEM [34] and Patch-Netvlad [15]. Classic methods [9, 40, 51] (C) offer promising accuracy on day images but poor results on night images partially because they use handcrafted features for fine localization, which are more sensitive to illumination changes than learned features (H). With the assistance of segmentation, semantic-based works

Group	Method	Day	Night
R	Netvlad [2]	0.0 / 0.4 / 25.5	0.0 / 0.0 / 21.4
	AP-GEM [34]	0.0 / 0.1 / 22.6	0.0 / 0.0 / 16.3
	Patch-Netvlad [15]	0.0 / 0.1 / 20.0	0.0 / 0.0 / 21.4
C	AS [40]	85.3 / 92.2 / 97.9	39.8 / 49.0 / 64.3
	CSL [51]	52.3 / 80.0 / 94.3	29.6 / 40.8 / 56.1
	CPF [9]	76.7 / 88.6 / 95.8	33.7 / 48.0 / 62.2
S	SSM [47]	71.8 / 91.5 / 96.8	58.2 / 76.5 / 90.8
	VLM [63]	62.4 / 71.8 / 79.9	35.7 / 44.9 / 54.1
	SMC [56]	52.3 / 80.0 / 94.3	29.6 / 40.8 / 56.1
	HSC-Net [26]	71.1 / 81.9 / 91.7	40.8 / 56.1 / 76.5
H	SIFT [27]	82.8 / 88.1 / 93.1	30.6 / 43.9 / 58.2
	SPP [10]	80.5 / 87.4 / 94.2	42.9 / 62.2 / 76.5
	D2Net [11]	84.8 / 92.6 / 97.5	84.7 / 90.8 / 96.9
	R2D2 [35]		76.5 / 90.8 / **100.0**
	CAPS + SIFT [27, 61]		77.6 / 86.7 / 99.0
	CAPS + SPP [10, 61]		82.7 / 87.8 / **100.0**
	LISRD + SPP [10, 33]		78.6 / 86.7 / 98.0
	LLF + R2D2 [10, 50]		72.4 / 90.8 / 99.0
	SOSNet + D2D [54, 55]		73.5 / 83.7 / 96.9
	ContextDesc + SIFT [27, 28]		67.3 / 79.6 / 90.8
M	ASLFeat + OANet [29, 65]		77.6 / 89.8 / **100.0**
	ENCNet [36]		76.5 / 84.7 / 98.0
	Dual-RCNet [25]		79.6 / 88.8 / **100.0**
	PDCNet [59]		80.6 / 87.8 / **100.0**
	DGCNet [31]	22.9 / 49.8 / 84.7	14.3 / 37.8 / 79.6
	Pixloc [39]	84.7 / 94.2 / **98.8**	81.6 / **93.9** / **100.0**
	AHM [13]	47.8 / 72.2 / 91.3	30.6 / 53.1 / 78.6
	S2DNet [14]	84.5 / 90.3 / 95.3	74.5 / 82.7 / 94.9
	Patch2Pix [67]	84.6 / 92.1 / 96.5	82.7 / 92.9 / 99.0
	SPP + Superglue [10, 38]	**89.6** / 95.4 / **98.8**	**86.7** / **93.9** / **100.0**
	Ours	88.3 / **95.6** / **98.8**	84.7 / **93.9** / **100.0**

Table 2. **Results on Aachen dataset.** The best and second best results are highlighted with **bold** and red fonts.

(S) yield better results for night images, but still worse than learned features (H).

HLoc [37] with Netvlad [2] for coarse localization and powerful local features for fine localization currently is the standard pipeline (H), but our model outperforms all works in group H. In group M, with advanced and dense matcher, most methods report better accuracy than others groups and SPP+Superglue achieves the state-of-the-art accuracy. However, due to instance-wise detection and matching as well as two-step pose estimation, our method gives very close results to SPP+Superglue and outperforms all other methods in groups C, S, H, and M.

As most methods in groups H and M only report results on 98 night query images in Aachen dataset, we also show their results on Aachen_v1.1 containing more challenging (191) night images. Table 1 demonstrates obvious performance drop of all methods because of increased test samples. While compared with models in group H, our system still gives much better performance especially on night images. Although some works in group M achieve close results to ours by utilizing advanced [38] or dense matching models [49], they sacrifice the time [38] or memory [49] efficiency, and require additional datasets for training.

Results on RoboCS dataset. We further test our ap-

Group	Method	Day			Night	
		overcast-summer	overcast-winter	snow	night	night-rain
R	Netvlad [2]	5.8 / 28.5 / 96.3	2.8 / 26.7 / 93.3	7.2 / 24.5 / 92.2	0.5 / 3.2 / 37.0	0.9 / 5.7 / 43.2
	AP-GEM [34]	3.9 / 16.6 / 82.3	2.8 / 18.2 / 85.4	5.5 / 15.5 / 82.2	0.5 / 0.7 / 21.7	0.0 / 1.4 / 28.2
	Patch-Netvlad [15]	5.4 / 28.5 / 91.6	4.1 / 24.4 / 92.1	7.8 / 24.9 / 89.2	0.2 / 1.1 / 12.8	1.4 / 3.9 / 20.5
C	AS [40]	32.8 / 74.1 / 97.8	37.4 / 78.7 / 94.6	50.5 / 81.8 / 95.5	1.6 / 3.9 / 10.5	2.0 / 10.9 / 18.0
	CSL [51]	34.1 / 71.1 / 93.5	39.5 / 75.9 / 92.3	53.2 / 83.6 / 92.4	0.2 / 0.9 / 5.3	0.9 / 4.3 / 9.1
	CPF [9]	36.5 / 76.5 / 97.8	43.1 / 78.2 / 93.6	54.2 / 84.9 / 95.3	2.3 / 6.6 / 15.3	4.5 / 12.3 / 18.6
S	SSM [47]	44.1 / **79.3** / 99.8	48.5 / **81.5** / 96.2	60.3 / 85.7 / 96.1	10.0 / 23.7 / 45.4	14.5 / 33.2 / 47.5
	VLM [63]	41.3 / 74.5 / 96.1	47.4 / 75.1 / 90.3	55.2 / 83.0 / 91.8	11.9 / 26.0 / 55.0	15.7 / 34.5 / 60.5
	SMC [56]	39.5 / 75.6 / 92.4	39.5 / 72.3 / 85.1	56.4 / 85.7 / 98.0	6.2 / 18.5 / 44.3	8.0 / 26.4 / 46.4
	DASGIL-FD [18]	6.0 / 31.1 / 84.7	5.1 / 27.2 / 85.9	10.6 / 29.4 / 76.9	1.6 / 4.8 / 19.9	1.8 / 4.3 / 21.6
	ToDayGAN + D2Net [1, 11]	40.0 / 79.0 / 98.9	49.0 / 77.9 / 96.2	57.1 / 84.5 / 95.7	16.4 / 43.2 / 73.3	24.1 / 50.5 / 74.1
H	SIFT [27]	39.5 / 72.4 / 91.4	52.6 / 78.7 / 95.1	61.1 / 85.1 / 95.1	7.8 / 13.9 / 22.1	9.5 / 14.5 / 17.0
	SPP [10]	43.4 / 78.2 / 99.4	54.1 / 80.3 / 96.2	62.8 / 85.5 / 96.9	16.9 / 41.6 / 71.5	22.0 / 45.0 / 68.0
	D2Net [11]	40.4 / 77.3 / 98.3	51.8 / 78.5 / 96.2	60.5 / 84.5 / 94.9	18.0 / 39.7 / 53.9	22.7 / 40.5 / 56.1
	R2D2 [35]	**45.6** / 78.8 / 99.8	**55.4** / 80.3 / 97.7	**63.4** / 86.1 / **98.6**	18.3 / 43.4 / 67.8	29.1 / 50.2 / 68.2
	DIFL + FCL [19]	5.4 / 27.2 / 72.8	4.6 / 22.8 / 78.7	9.0 / 25.2 / 71.8	0.5 / 2.7 / 8.4	4.5 / 10.2 / 23.2
M	SPP + Superglue [10, 38]	45.1 / 79.0 / **100.0**	54.6 / 80.0 / 96.2	62.0 / 84.9 / **98.6**	24.2 / **62.6** / 87.4	42.3 / 69.3 / 90.2
	Pixloc [39]	41.9 / 75.4 / 99.1	53.8 / 80.3 / 97.7	60.9 / 84.5 / 95.7	16.4 / 30.8 / 75.1	30.7 / 49.8 / 80.5
	AHM [13]	37.8 / 74.3 / 96.8	42.1 / 76.2 / 96.2	51.9 / 83.6 / 95.7	16.2 / 55.3 / **93.6**	28.4 / 68.4 / **95.5**
	Ours	45.4 / 79.0 / **100.0**	54.6 / 80.5 / **97.9**	63.0 / **86.5** / **98.6**	**24.9** / 62.3 / 86.1	**47.5** / **73.4** / 90.0

Table 3. **Results on RobotCar-Seasons dataset.** The best and second best results are highlighted with **bold** and red fonts.

proach on RoboCS dataset [30] under different illumination, weather, and season conditions. Table 3 demonstrates the results. Images in RoboCS were recorded by a moving car without large variations of rotation, so most methods give promising accuracy on day images in different season and weather conditions (overcast-summer, overcast-winter, snow). However, our system outperforms them. Besides, almost all of them (C, S, H) fail to produce good results on night images (night, night-rain). This problem is mitigated by semantic information, but the usage of semantic labels for simple filtering are fragile to segmentation errors on night images, so the performance of these methods (S) is very poor. On the contrary, our robust strategy of using instance labels produces state-of-the-art results, which are even better than those in group M.

Influence of the number of keypoints. Since both the time and memory efficiency are significant to real applications, we further evaluate the performance of different methods on Aachen_v1.1 dataset [42] in Fig. 6 by progressively reducing the number of keypoints used for mapping and localization from 4k to 1k. We compare our system with R2D2 [2, 35, 37] (H) and SPP+Superglue [10, 38] (M), which are the best methods without and with advanced matcher. We also list results of SPP [10] (H) as reference.

Since both SPP and R2D2 detect keypoints exhaustively from images, they fail to retain enough valid keypoints when the number of keypoints is decreased, resulting in dramatic performance drop especially for night images. By leveraging the global context of features, Superglue addresses this problem effectively. While instance-wise detection enables our model to retain more robust keypoints at low cost, which ameliorates the influence even more effectively when only 1k keypoints are used.

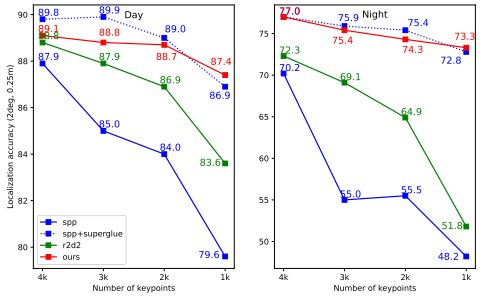


Figure 6. **Influence of the number of keypoints.** We progressively decrease the number of keypoint for both mapping and localization from 4k to 1k. Results of R2D2 [35], SPP [10], SPP+Superglue [10, 38], and our method are reported.

Qualitative results. Fig. 7 shows the predicted and automatically generated ground-truth segmentation masks of the query and reference images and their matches. Our model is able to correctly recognize global instances under different viewpoints, illuminations, and seasons. As we adopt the instance-wise detection and matching technique, most matches are from the these robust areas, boosting the localization accuracy. Due to heavy occlusions or huge viewpoint changes, our network sometimes can only recognize a small part of the instance, which is enough for finding the correct reference image but might impair the robustness of fine localization if filtering-based usage of labels is adopted. Fortunately, our robust instance-based detection and matching strategy handles this situation successfully. We provide more results and analysis about the detection and matching processes in the **supplementary material**.

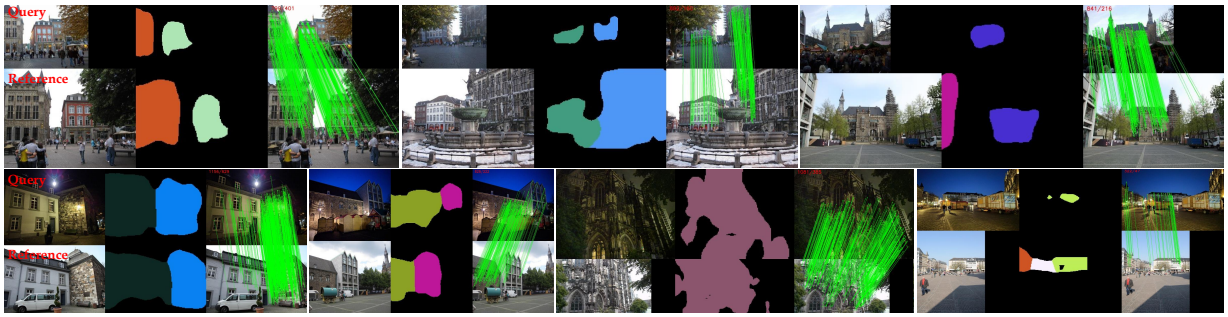


Figure 7. **Qualitative results.** We visualize recognition results, matches between the query and reference images with different viewpoints, illumination and season conditions. More qualitative results of recognition, detection, and matching are in the **supplementary material**.

Model	Input size	Running time (ms)
Netvlad [2]	1024×1024	31.9
Ours (Recognition)	256×256	9.2
R2D2 [35]	1024×1024	72.4
SPP [10]	1024×1024	12.0
Ours (Local feature)	1024×1024	30.1
Superglue [38]	2k×2k	52.0
Superglue [38]	4k×4k	146.8

Table 4. **Running time.** We report the inference time of our recognition and local feature modules in comparison with previous retrieval, local feature, and matching methods.

Inst. detection & matching	Refinement	Day	Night
		86.3 / 95.0 / 98.7	71.7 / 86.4 / 99.0
✓		87.1 / 95.1 / 98.7	73.3 / 85.9 / 99.5
✓	✓	89.1 / 96.1 / 99.3	77.0 / 90.1 / 99.5

Table 5. **Ablation study.** We test the effectiveness of our instance-wise detection and matching, and the two-step pose estimation on the Aachen_v1.1 dataset.

4.3. Time analysis

In this section, we give an analysis of running time. Considering the implementation variations of different methods (C++/Python) and the complexity of localization process, we test the time of modules which take up the majority of time. Table 4 shows the average time of 1,000 measurements on a RTX3090 GPU. We utilize low (256×256) and high (1024×1024) resolutions for recognition and local feature extraction respectively to balance the efficiency and accuracy. Our recognition module runs much faster than Netvlad (image size of 1024 is used by methods in groups H and M). Since R2D2 extracts features from images with full size, it is much slower than ours ($4\times$ downsampling) and SPP ($8\times$ downsampling). Superglue [38] is powerful but slow even with only 2k keypoints as input. Although R2D2 and SPP+Superglue achieve outstanding accuracy, they have to sacrifice the efficiency, whereas, our system maintains high accuracy and efficiency at the same time.

4.4. Ablation study

Finally, we verify the efficacy of each component in our model by progressively adding the instance-wise detection and matching as well as the second step for pose refinement. Table 5 shows the results on Aachen_v1.1 dataset. When introducing the instance-wise detection and matching, we can see improvement especially on night images. Since we adopt explicit assignment of instance labels, the usage of semantics is inevitably influenced by recognition results. However, it is more important to efficiency especially when the number of keypoints is limited. A huge improvement can be seen by incorporating the refinement step because of more correct references and fewer outliers (rejected by the initially estimated pose).

5. Limitations

Our framework is based on building instance recognition, so its performance is influenced by the distribution of buildings in the map. For areas without buildings, we have to perform global search to find reference images. Besides, directly recognizing a large number of global instances in large-scale scenes is memory-consuming, but hierarchical recognition [20] has the potential to solve this problem.

6. Conclusion

In this paper, we propose an efficient and accurate localization framework based on global building instance recognition. Specifically, a recognition-based progressive reference search strategy is proposed to find candidates from local areas rather than the whole database. Besides, robust instance-wise feature detection and matching techniques are introduced to enhance the localization robustness under challenging conditions with recognition errors considered. Moreover, a two-step pose estimation is adopted to make our recognition-based framework work faster and yield more accurate results at low cost. Experiments demonstrate that our model outperforms previous methods in terms of both accuracy and efficiency.

References

[1] Asha Anoosheh, Torsten Sattler, Radu Timofte, Marc Pollefeys, and Luc Van Gool. Night-to-day image translation for retrieval-based localization. In *ICRA*, 2019. 1, 2, 7

[2] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *CVPR*, 2016. 1, 2, 4, 5, 6, 7, 8

[3] Aritra Bhowmik, Stefan Gumhold, Carsten Rother, and Eric Brachmann. Reinforced feature points: Optimizing feature detection and description for a high-level task. In *CVPR*, 2020. 2

[4] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. DSAC-differentiable RANSAC for camera localization. In *CVPR*, 2017. 1, 2, 3

[5] Eric Brachmann and Carsten Rother. Learning less is more-6d camera localization via 3d surface regression. In *CVPR*, 2018. 1, 2, 3

[6] Ignas Budvytis, Patrick Sauer, and Roberto Cipolla. Semantic localisation via globally unique instance segmentation. In *BMVC*, 2018. 1, 2, 3

[7] Ignas Budvytis, Marvin Teichmann, Tomas Vojir, and Roberto Cipolla. Large scale joint semantic re-localisation and scene understanding via globally unique instance coordinate regression. In *BMVC*, 2019. 1, 2, 3

[8] Bingyi Cao, Andre Araujo, and Jack Sim. Unifying deep local and global features for image search. In *ECCV*, 2020. 4

[9] Wentao Cheng, Weisi Lin, Kan Chen, and Xinfeng Zhang. Cascaded parallel filtering for memory-efficient image-based localization. In *ICCV*, 2019. 2, 5, 6, 7

[10] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPRW*, 2018. 1, 2, 4, 5, 6, 7, 8

[11] 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 *CVPR*, 2019. 2, 5, 6, 7

[12] 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*, 1981. 5

[13] Hugo Germain, Guillaume Bourmaud, and Vincent Lepetit. Sparse-to-dense hypercolumn matching for long-term visual localization. In *3DV*, 2019. 2, 5, 6, 7

[14] Hugo Germain, Guillaume Bourmaud, and Vincent Lepetit. S2DNet: Learning accurate correspondences for sparse-to-dense feature matching. In *ECCV*, 2020. 2, 6

[15] Stephen Hausler, Sourav Garg, Ming Xu, Michael Milford, and Tobias Fischer. Patch-NetVLAD: Multi-Scale Fusion of Locally-Global Descriptors for Place Recognition. In *CVPR*, 2021. 1, 2, 4, 5, 6, 7

[16] Kun He, Yan Lu, and Stan Sclaroff. Local descriptors optimized for average precision. In *CVPR*, 2018. 4

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

[18] Hanjiang Hu, Zhijian Qiao, Ming Cheng, Zhe Liu, and Hesheng Wang. DASGIL: Domain adaptation for semantic and geometric-aware image-based localization. *TIP*, 2020. 1, 2, 7

[19] Hanjiang Hu, Hesheng Wang, Zhe Liu, Chenguang Yang, Weidong Chen, and Le Xie. Retrieval-based localization based on domain-invariant feature learning under changing environments. In *IROS*, 2019. 2, 7

[20] Zhaoyang Huang, Han Zhou, Yijin Li, Bangbang Yang, Yan Xu, Xiaowei Zhou, Hujun Bao, Guofeng Zhang, and Hongsheng Li. VS-Net: Voting with segmentation for visual localization. In *CVPR*, 2021. 8

[21] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *ICCV*, 2015. 1, 2

[22] Nikolay Kobyshev, Hayko Riemenschneider, and Luc Van Gool. Matching features correctly through semantic understanding. In *3DV*, 2014. 1

[23] Mans Larsson, Erik Stenborg, Carl Toft, Lars Hammarstrand, Torsten Sattler, and Fredrik Kahl. Fine-grained segmentation networks: Self-supervised segmentation for improved long-term visual localization. In *ICCV*, 2019. 1, 2, 3

[24] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o (n) solution to the pnp problem. *IJCV*, 2009. 2, 5

[25] Xinghui Li, Kai Han, Shuda Li, and Victor Prisacariu. Dual-resolution correspondence networks. In *NeurIPS*, 2020. 2, 5, 6

[26] Xiaotian Li, Shuzhe Wang, Yi Zhao, Jakob Verbeek, and Juho Kannala. Hierarchical scene coordinate classification and regression for visual localization. In *CVPR*, 2020. 3, 5, 6

[27] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 1, 6, 7

[28] Zixin Luo, Tianwei Shen, Lei Zhou, Jiahui Zhang, Yao Yao, Shiwei Li, Tian Fang, and Long Quan. Contextdesc: Local descriptor augmentation with cross-modality context. In *CVPR*, 2019. 5, 6

[29] 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 *CVPR*, 2020. 2, 6

[30] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The Oxford RobotCar dataset. *IJRR*, 2017. 2, 5, 7

[31] Iaroslav Melekhov, Aleksei Tiulpin, Torsten Sattler, Marc Pollefeys, Esa Rahtu, and Juho Kannala. DGC-Net: Dense geometric correspondence network. In *WACV*, 2019. 5, 6

[32] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *ICCV*, 2017. 2

[33] Rémi Pautrat, Viktor Larsson, Martin R. Oswald, and Marc Pollefeys. Online Invariance Selection for Local Feature Descriptors. In *ECCV*, 2020. 5, 6

[34] Jerome Revaud, Jon Almazán, Rafael S Rezende, and Cesar Roberto de Souza. Learning with average precision: Training image retrieval with a listwise loss. In *ICCV*, 2019. 1, 2, 5, 6, 7

[35] Jerome Revaud, Philippe Weinzaepfel, César Roberto de Souza, and Martin Humenberger. R2D2: Repeatable and reliable detector and descriptor. In *NeurIPS*, 2019. 1, 2, 5, 6, 7, 8

[36] Ignacio Rocco, Relja Arandjelović, and Josef Sivic. Efficient neighbourhood consensus networks via submanifold sparse convolutions. In *ECCV*, 2020. 2, 5, 6

[37] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019. 1, 2, 5, 6, 7

[38] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 2, 5, 6, 7, 8

[39] 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 *CVPR*, 2021. 5, 6, 7

[40] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & effective prioritized for large-scale image-based localization. *TPAMI*, 2016. 1, 2, 5, 6, 7

[41] 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 *CVPR*, 2018. 2, 5

[42] Torsten Sattler, Tobias Weyand, Bastian Leibe, and Leif Kobbelt. Image retrieval for image-based localization revisited. In *BMVC*, 2012. 2, 5, 6, 7

[43] Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixe. Understanding the limitations of cnn-based absolute camera pose regression. In *CVPR*, 2019. 1, 2

[44] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-Motion Revisited. In *CVPR*, 2016. 2

[45] Johannes L Schönberger, Marc Pollefeys, Andreas Geiger, and Torsten Sattler. Semantic visual localization. In *CVPR*, 2018. 1

[46] Zachary Seymour, Karan Sikka, Han-Pang Chiu, Supun Samarasekera, and Rakesh Kumar. Semantically-aware attentive neural embeddings for long-term 2d visual localization. In *BMVC*, 2019. 1, 2

[47] Tianxin Shi, Shuhan Shen, Xiang Gao, and Lingjie Zhu. Visual localization using sparse semantic 3D map. In *ICIP*, 2019. 1, 2, 3, 5, 6, 7

[48] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *CVPR*, 2016. 3

[49] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. In *CVPR*, 2021. 2, 5, 6

[50] Suwichaya Suwanwimolkul, Satoshi Komorita, and Kazuyuki Tasaka. Learning of low-level feature keypoints for accurate and robust detection. In *WACV*, 2021. 5, 6

[51] Linus Svärm, Olof Enqvist, Fredrik Kahl, and Magnus Oskarsson. City-scale localization for cameras with known vertical direction. *TPAMI*, 2016. 2, 5, 6, 7

[52] Fuwen Tan, Jiangbo Yuan, and Vicente Ordonez. Instance-level Image Retrieval using Reranking Transformers. In *ICCV*, 2021. 3

[53] Marvin Teichmann, Andre Araujo, Menglong Zhu, and Jack Sim. Detect-to-retrieve: Efficient regional aggregation for image search. In *CVPR*, 2019. 3

[54] Yurun Tian, Vassileios Balntas, Tony Ng, Axel Barroso-Laguna, Yiannis Demiris, and Krystian Mikolajczyk. D2D: Keypoint extraction with describe to detect approach. In *ACCV*, 2020. 6

[55] Yurun Tian, Xin Yu, Bin Fan, Fuchao Wu, Huub Heijnen, and Vassileios Balntas. SOSNet: Second order similarity regularization for local descriptor learning. In *CVPR*, 2019. 6

[56] Carl Toft, Erik Stenborg, Lars Hammarstrand, Lucas Brynte, Marc Pollefeys, Torsten Sattler, and Fredrik Kahl. Semantic match consistency for long-term visual localization. In *ECCV*, 2018. 1, 2, 3, 5, 6, 7

[57] Giorgos Tolias, Tomas Jenicek, and Ondřej Chum. Learning and aggregating deep local descriptors for instance-level recognition. In *ECCV*, 2020. 2, 3

[58] Akihiko Torii, Relja Arandjelovic, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 place recognition by view synthesis. In *CVPR*, 2015. 1, 2, 4

[59] Prune Truong, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning accurate dense correspondences and when to trust them. In *CVPR*, 2021. 5, 6

[60] Michał J Tyszkiewicz, Pascal Fua, and Eduard Trulls. DISK: Learning local features with policy gradient. In *NeurIPS*, 2020. 2

[61] Qianqian Wang, Xiaowei Zhou, Bharath Hariharan, and Noah Snavely. Learning feature descriptors using camera pose supervision. In *ECCV*, 2020. 2, 5, 6

[62] Tobias Weyand, Andre Araujo, Bingyi Cao, and Jack Sim. Google landmarks dataset v2-a large-scale benchmark for instance-level recognition and retrieval. In *CVPR*, 2020. 3

[63] Zhe Xin, Yinghao Cai, Tao Lu, Xiaoxia Xing, Shaojun Cai, Jixiang Zhang, Yiping Yang, and Yanqing Wang. Localizing discriminative visual landmarks for place recognition. In *ICRA*, 2019. 1, 2, 5, 6, 7

[64] Fei Xue, Xin Wu, Shaojun Cai, and Junqiu Wang. Learning multi-view camera relocalization with graph neural networks. In *CVPR*, 2020. 1, 2

[65] 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 *CVPR*, 2019. 2, 6

[66] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 3, 5

[67] Qunjie Zhou, Torsten Sattler, and Laura Leal-Taixe. Patch2pix: Epipolar-guided pixel-level correspondences. In *CVPR*, 2021. 2, 5, 6