Sparse-to-Dense Hypercolumn Matching for Long-Term Visual Localization

Hugo Germain* [1] Guillaume Bourmaud* [2] Vincent Lepetit* [1]
[1]Laboratoire Bordelais de Recherche en Informatique, Université de Bordeaux, France
[2]Laboratoire IMS, Université de Bordeaux, France

Abstract

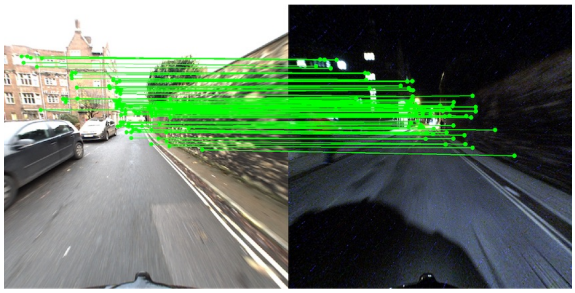
We propose a novel approach to feature point matching, suitable for robust and accurate outdoor visual localization in long-term scenarios. Given a query image, we first match it against a database of registered reference images, using recent retrieval techniques. This gives us a first estimate of the camera pose. To refine this estimate, like previous approaches, we match 2D points across the query image and the retrieved reference image. This step, however, is prone to fail as it is still very difficult to detect and match sparse feature points across images captured in potentially very different conditions. Our key contribution is to show that we need to extract sparse feature points only in the retrieved reference image: We then search for the corresponding 2D locations in the query image exhaustively. This search can be performed efficiently using convolutional operations, and robustly by using hypercolumn descriptors, i.e. *image features computed for retrieval. We refer to this method as 'Sparse-to-Dense Hypercolumn Matching'. Because we know the 3D locations of the sparse feature points in the reference images thanks to an offline reconstruction stage, it is then possible to accurately estimate the camera pose from these matches. Our experiments show that this method allows us to outperform the state-of-the-art on several challenging outdoor datasets.*

1. Introduction

Visual localization is a key component to many robotic systems, ranging from autonomous navigation [45] to augmented or mixed reality [46]. Yet, accurately predicting the 6 DoF camera pose of a visual query with respect to a reference frame can become very challenging in long-term scenarios: Despite recent progress, many outdoor location methods are still prone to fail especially at high precision thresholds and under day-to-night changes [58] as images can undergo a wide variety of visual changes between different time of day and across seasons.

[1]E-mail: {firstname.lastname}@u-bordeaux.fr

(a) Standard sparse-to-sparse matching using SuperPoint detections and two different descriptors : (Left) SuperPoint descriptors [4 inliers], (Right) HyperColumn descriptors [5 inliers]



(b) Our sparse-to-dense matching approach using Superpoint detections in the left image only and HyperColumn descriptors [87 inliers]

Figure 1: Top images: Despite recent progress, matching sparse feature points extracted from two images captured under very different conditions remains extremely challenging. Bottom image: Our key contribution is to show that it is much more robust to extract sparse feature points in only one image, and to search for their correspondents exhaustively in the other image. This exhaustive search can be performed very efficiently using convolutional operations. Using the 3D locations of the sparse feature points, we can then compute the camera pose. We show the number of inlier matches found by PnP+RANSAC.

Visual localization approaches can be classified into two categories: *Structure-based* and *image-based* methods. In structure-based methods, the camera pose is estimated from correspondences between 2D points from the query image and a reconstructed 3D point-cloud of the whole scene. This can lead to great accuracy, but often to mediocre robustness to strong visual changes. Image-based methods predict the

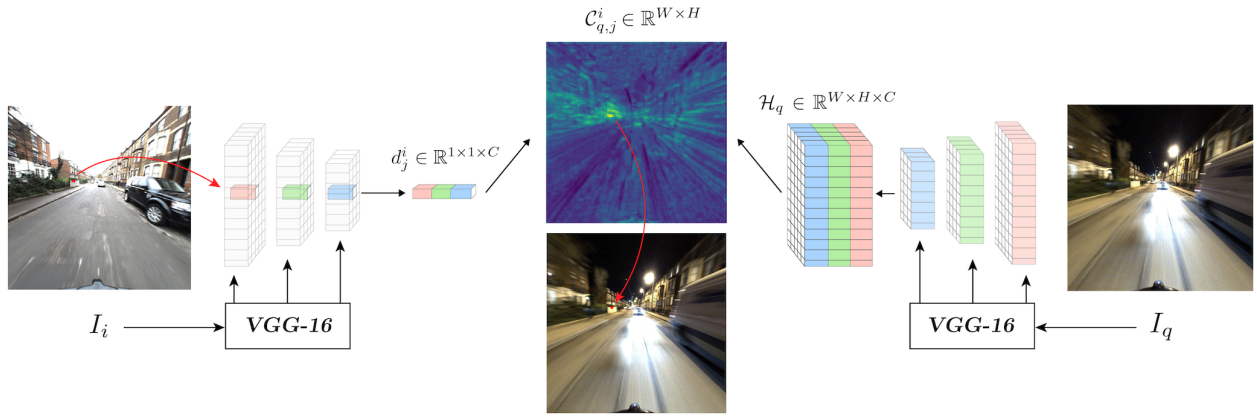


Figure 2: **Sparse-to-dense feature matching using hypercolumns.** For each detection m^i_jin the reference image I_i retrieved for query image I_q, we extract a hypercolumn descriptor d^i_j, which we cross-correlate exhaustively against the dense hypercolumn $\mathcal{H}_q$. We then define the correspondent location of m^i_j in I_q as the image location of the maximum value in the resulting correlation map $\mathcal{C}^i_{q,j} = d^i_j * \mathcal{H}_q$.

query's camera pose by retrieving the nearest geo-tagged image in a reference database [2, 5, 14, 71]. The advantage is that image retrieval methods can be very robust to strong appearance changes [2, 21, 52, 71]. The accuracy highly depends on the spatial sampling of the database, but unfortunately high sampling rate is costly both in terms of capture time and memory footprint. It is therefore natural to combine the two approaches [27, 46, 52, 53] into a 'hierarchical' pipeline by finding 2D-3D correspondences only within a subset of the 3D point cloud, obtained using image retrieval. Such methods benefit from the speed and robustness of image-based approaches, and the accuracy of structure-based methods in lenient capturing conditions.

Still, even when using very recent sparse feature detectors and descriptors [3, 17, 48, 73], local 2D-3D matching is prone to fail under strong visual changes in practice [52, 58]. As illustrated in Fig. 1, it is mostly because it is still difficult to extract the same sparse feature points in two images taken under different conditions.

We therefore propose to detect sparse feature points only in the reference images. Keeping these sparse feature points is important as they provide the 3D information required to compute the camera pose in an efficient way. To match these points against the query image, we perform an exhaustive search, which can be implemented efficiently with convolutional operations—the matching procedure takes 10ms on average in our implementation. Moreover, we notice that the image features extracted by VGG when trained together with NetVLAD to compute a robust global image descriptor provide local descriptions that are remarkably robust to capture condition changes. For our exhaustive search, we therefore rely on these features, which are sometimes called 'Hypercolumns' [24].

We call the resulting matching method 'Sparse-to-Dense Hypercolumn Matching'. We show that when used together with a powerful retrieval method, it outperforms existing pipelines on several challenging outdoor localization datasets.

The rest of the paper is structured as follows: Section 2 discusses the related work while section 3 introduces our localization pipeline. Our novel 'Sparse-to-Dense Hypercolumn Matching' approach is presented in section 4. Section 5 describes our experimental setup to thoroughly evaluate our approach in the context of long-term localization, and provides localization results. Source code will be made available.

2. Related Work

In this section, we review existing approaches tackling the problem of long-term visual localization. We distinguish *structure-based* methods, which leverage a 3D model of the scene, from *retrieval-based* methods, which do not.

2.1. Structure-Based Localization

Structure-based methods regress the full 6 DoF camera pose of query images using direct 2D-3D correspondences. Such methods [38, 39, 41, 54, 60, 66] work by first acquiring a point-cloud model of the scene through *SfM*, and computing local feature descriptors like SIFT [42], RootSIFT [3] or LIFT [73]. These descriptors are in turn used to obtain 2D-to-3D correspondences, and the predicted camera can usually be inferred from those matches using RANSAC [19, 59] combined with a Perspective-n-Point (P*n*P) solver [13, 23, 35, 37].

In consistent daytime conditions, such methods achieve very competitive results [57, 60, 66, 72]. However, they

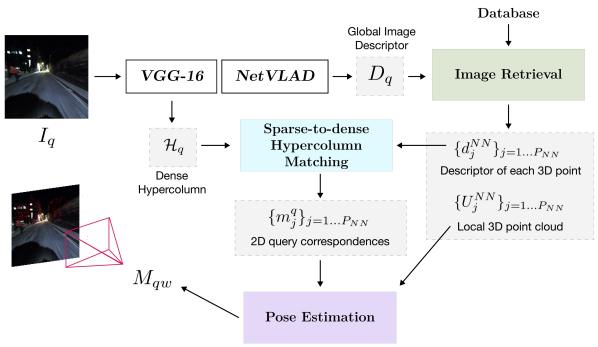


Figure 3: **Overview of our hierarchical localization pipeline.** Given a query image, we compute dense hypercolumns and a global image descriptor using NetVLAD [2]. The image features are extracted using VGG-16 specifically trained for the image retrieval task under varying capture conditions such as day and night. We find top-ranked images in a pose-annotated image database, and subsequently use the locally reconstructed point cloud as a feature point detection source. For each feature, we extract sparse hypercolumns and match each of them exhaustively with the query dense representation. This results in numerous robust correspondences suitable to perform PnP+RANSAC across changing conditions.

rely heavily on the accuracy and robustness of the local 2D-3D correspondences. Research in structure-based approaches mostly focuses on improving descriptor matching efficiency [15, 36, 38, 40, 43, 57], speed [18, 26] and robustness [39, 54, 55, 66, 67, 75]. Yet, under strong condition changes, failures in direct matching start to appear and damage the localization performance [58]. In order to improve the robustness of local feature descriptors and thus increase long-term localization performance, recent methods have used semantic reasoning [69]. Indeed, semantic maps are to some extent condition-invariant, and can enhance either the feature matching stage [6, 34, 62, 65] or the pose estimation stage [69]. While being accurate at small scale, feature-based methods bottleneck is scalability. In large-scale scenarios, both the construction of precise 3D models (and their maintenance) and local feature-matching is challenging and expensive [60].

2.2. Image-Based Localization

In image-based, or retrieval-based, localization methods, accuracy is traded-off for scalability. The scene is modeled as an image database containing ground-truth 6-DoF pose annotations. To infer the pose of a visual query, one can use compact image-level representations to retrieve the top-ranked image from the database and use their labels as pose approximation [14, 60, 74, 76]. The need for ground-truth 3D geometry is alleviated, and this method can easily generalize to large-scale environments.

To obtain robust global image descriptors, one can aggregate local features in the image into a fixed-size representation. VLAD [4] is a popular descriptor, computed by summing and concatenating many descriptors for affine-invariant regions. DenseVLAD [71] reformulates the VLAD architecture by densely sampling RootSIFT [3] descriptors in the image. Recent learning-based variants cast the task of image retrieval as a metric learning problem. NetVLAD [2] defines a differentiable VLAD layer as the final activation of a siamese network. Other activations layers [7, 22, 30, 50, 51, 70] coupled with siamese or triplet architectures, have shown to deliver competitive results for the task of image-retrieval [49]. In a very large database, unsupervised descriptor compression like PCA [28] or Product Quantization (PQ) [29] enables efficient approximate nearest-neighbor search with little loss in performance [22].

Other image-based methods include end-to-end learning approaches, which avoid using explicit feature matching altogether and leverages CNNs to learn robust representations [10, 11, 12, 31]. These methods are either hard to initialize [58, 62], struggle with large environments [58] and/or provide overall poor performance [9, 11, 32, 72].

2.3. Hierarchical Localization

For the problem of long-term localization, where strong appearance changes can occur because of the light or season differences, global descriptors have shown to provide robust pose initialization under strong visual changes [21, 52, 58]. Still, the main bottleneck of retrieval-based localization is the pose approximation step. Several schemes can be implemented to refine the coarsely estimated pose. For instance, view synthesis [68, 71] artificially generates intermediate samples, relative pose regression [9, 68] acts as a separate refinement step and multi-image methods [9, 74, 76] combine the top ranked images to improve pose accuracy.

The image-retrieval step can also be seen as a way to obtain a query's coarse location, before running a structure-based pose refinement algorithm. By doing so, 2D-3D matching is only run on a subset of the whole point cloud, leading to competitive results at small computational costs [27, 46, 52, 53].

2.4. Learning-Based Feature Matching

Even in a hierarchical localization pipeline, refining the query camera pose using 2D-3D correspondences can prove to be difficult if the features are not invariant to visual changes and the detections are not consistent across conditions. With the advent of CNNs, learning-based methods for local feature matching have emerged. Methods such as LF-Net [48], SuperPoint [17] or DELF [47] perform both keypoint detection and feature descriptor computation us-

ing end-to-end learning. Under strong condition changes such as day-to-night, even learning-based feature descriptors fail to generalize well [52]. In this paper, we propose to reuse the pixel-wise dense features directly from the image-retrieval backbone network, and show they are more suited for long-term visual localization.

3. Method

We give in this section an overview of our pipeline. We first formalize the problem and its assumption in Section 3.1. We then provide an overall description of our method in Section 3.2.

3.1. Problem Statement

We assume that a database of registered reference images is available. More precisely, for each reference image I_i of the database, we assume that the following is available:

- A normalized global image descriptor D_i computed as explained in Section 3.2, which we will use for the retrieval step.

Moreover, additional information, which we will use for the pose refinement step, are also stored:

- the calibration matrix K_i and the absolute camera pose M_{iw} expressed in the world coordinate system;
- a set of P_i 2D feature points $\{m_j^i\}_{j=1...P_i}$ detected using SuperPoint [17];
- the descriptor d_j^i for each feature point m_j^i computed as explained in Section 4;
- the 3D coordinates U_j^i of each feature point m_j^i.

Given a query image I_q with known calibration matrix K_q, and this database, we aim to predict the camera pose M_{qw}.

3.2. Our Hierarchical Localization Pipeline

When performing localization in large-scale environments, matching a set of 2D keypoints with a large number of 3D landmarks can be difficult [60]. As suggested by [52], one way to reduce the set of 3D points to match the image keypoints against is to first perform image retrieval. The returned top-ranked images in the database provide us with a subset of the large 3D point cloud for which performing local feature matching is much more efficient. The whole pipeline is presented in Figure 3.

Image Retrieval. Like previous methods [2, 7, 22, 30, 50, 51, 70], we use a Siamese network approach to learn a discriminative image descriptor robust to changes of the capture conditions. For the architecture, we opt for the popular NetVLAD [2] pooling layer with a VGG-16 [64] backbone.

During training, we define positive and negative labels $l(I_i, I_j) \in \{0, 1\}$ for pairs of images, based on the presence or absence of co-visibility between images respectively. We use the same contrastive loss as [2]. Once trained, the network provides a global descriptor D_i for each reference image, which is stored in the database.

At test time, given a query image I_q, we compute its descriptor D_q and retrieve its k nearest neighbors by computing the Euclidean distance between D_q and each stored descriptor D_i. Such top-ranked images provide coarse camera poses which are sufficient to estimate a query's emplacement [58].

Camera Pose Refinement. In order to obtain a more accurate camera pose estimation, we make use of the local 3D point clouds fetched from the image retrieval step. For each of the k nearest neighbors, we establish 2D-3D correspondences and subsequently solve the pose using for instance a Perspective-n-Point (PnP) [13, 23, 35] solver. Given a set of matches, we refine the query pose using P3P [33] inside a RANSAC [20, 59] loop. The method we use to establish these correspondences is our main contribution, and we describe it below.

4. Sparse-to-Dense Hypercolumn Matching

If we followed the standard approach to obtain the 2D-3D correspondences needed to estimate the camera pose, we would extract sparse feature points in the query image and match them against the sparse feature points m_j^i extracted from the nearest neighbors of the query image. As mentioned in the introduction, this step is still very challenging, mostly because of the detection step that needs to identify the same image locations even under strong condition changes. In order to circumvent this challenging detection problem, we reformulate the local feature matching step to avoid performing detection in the query image, as illustrated in Fig. 2. To do so, we perform an exhaustive search in the query image for the correspondent of each sparse feature point detected in the reference images. We explain below how this search can be performed efficiently.

HyperColumn Extraction. In order to perform robust matching, we rely on image features that were already used to compute the global image descriptor as shown in Fig. 3. For each query image, we extract intermediate features from the VGG-16 [64] network and aggregate them in order to obtain a dense and rich representation of the image. We extract features from the layers `conv_3_3`, `conv_4_1`, `conv_4_3`, `conv_5_1`, `conv_5_3`. We refer to these representations as "hypercolumns" [24]. Each intermediate layer is upsampled using bilinear interpolation to match the resolution $W_{\mathcal{H}} \times H_{\mathcal{H}}$ of the earliest layer, before being con-

catenated along the channel axis and normalized. We define the obtained hypercolumns for the query image I_q as $\mathcal{H}_q \in \mathbb{R}^{W_\mathcal{H} \times H_\mathcal{H} \times C}$.

For each reference image I_i, we are only interested in descriptors located at feature points. We thus only store in the database the hypercolumns at locations $\{m_j^i\}_{j=1...P_i}$. We denote $\mathcal{S}_i = \{d_j^i\}_{j=1..P_i}$ this set of sparse descriptors, where $d_j^i \in \mathbb{R}^{1\times1\times C}$.

Sparse-to-Dense Matching. To find correspondences between the set of sparse descriptors from the reference image $\mathcal{S}_i$ and the dense hypercolumns $\mathcal{H}_q$, we perform a dot product. These dot products can be efficiently implemented with a 1×1 convolution. We define the resulting cross-correlation map as $\mathcal{C}_{q,j}^i = \mathcal{H}_q * d_j^i \in \mathbb{R}^{W_\mathcal{H} \times H_\mathcal{H}}$. To retrieve the final 2D keypoints in the query image, we first fetch the global maximum of the cross-correlation map and upsample the retrieved coordinates to match the query image coordinates. Consequently, this 'Sparse-to-Dense matching' step always gives us P_i 2D-3D correspondences (See Figures 4 and 5).

Ratio Test. Some detections in the reference image may fall in image regions with repetitive textures, or in areas that are occluded in the query image. This may lead to ambiguities when looking for point correspondents. To discard matches with large ambiguity, we apply a ratio test similar to the one often used in more standard approaches, and defined as follows. For the cross-correlation map $\mathcal{C}_{q,j}^i$, let $\bar{\mathcal{C}}_{q,j}^i \in \mathbb{R}^{(W_\mathcal{H}.H_\mathcal{H})}$ be the flattened and sorted by decreasing order map. For a 2D-3D match to be retained, we apply the following rule:

$$\frac{\bar{\mathcal{C}}_{q,j}^i[0]}{\bar{\mathcal{C}}_{q,j}^i[f \times (W_\mathcal{H} \times H_\mathcal{H})]} > \alpha, f \in [0;1]\,. \qquad (1)$$

In practice, we use $\alpha = 0.9$, and adapt the factor f to the different datasets. Finding the value of $\bar{\mathcal{C}}_{q,j}^i[f\times(W_\mathcal{H}\times H_\mathcal{H})]$ actually does not require sorting the whole array, and adds negligible overload to the computational cost.

5. Experiments

In this section, we conduct experiments to evaluate our hierarchical localization approach under challenging conditions. In Section 5.1, we detail how both our evaluation datasets were setup and reconstructed. We also discuss the evaluation methods and baselines used for comparison. In Section 5.2, we show how our hierarchical method can solve camera poses accurately under challenging conditions and outperforms existing methods in such categories. Lastly, in Section 5.3, we run an ablation study, which demonstrates the improvements brought by our contribution.

Dataset	Training sequences	Condition	Training images	Reference images	Query images
RobotCar Seasons [44]	12 Dec 2014	overcast	20,965	6,954	3,978
	05 Dec 2014	overcast-rain	20,965		
	16 Dec 2014	night	19,376		
	03 Feb 2015	night	20,257		
Extended CMU-Seasons [8]	Slices 2-8	urban	9,612	7,159	75,335
	Slices 9-17	suburban	24,728		
	Slices 18-25	park	16,148		

Table 1: **Detailed statistics** regarding the training and testing sequences used for each dataset. Reference images are used to triangulate 3D keypoints offline using SuperPoint [17] detections and descriptors. Note that for RobotCar Seasons, only rear images are considered.

5.1. Evaluation Setup

We begin our evaluation by presenting the two challenging outdoor datasets introduced by [58] which we will be using throughout this section.

Datasets. Our evaluation set consists of two outdoor datasets captured from vehicles or using hand-held mobile phone cameras. Each of the provided datasets contains a set of reference images, along with their ground truth camera poses. We are also given sparse 3D reconstructions precomputed using RootSIFT [3] features by Sattler *et al*. [58]. In practice, we do not use the provided sparse 3D reconstruction and re-triangulated our own point clouds using SuperPoint [17] detections. We perform the triangulation using COLMAP [61, 63] on the reference images of each dataset, similarly to [52].

The first dataset is the Extended CMU-Seasons dataset [58], which contains about 40% more images than the original CMU-Seasons dataset [8]. It consists of 7,159 reference images and 75,335 query images, captured using two front-facing cameras mounted on a car, in the area of Pittsburgh. The images were captured over the course of a year and the reference images depict different seasonal conditions. The *park* scene is particularly difficult as it was captured in a rural environment and faces strong vegetation changes over the year.

The second dataset is the RobotCar Seasons dataset [44], which contains 6,954 daytime images captured by a rear-facing camera mounted on a car driving in Oxford. The 3,978 query images were taken over the course of a year, including some in very challenging conditions such at nighttime [58]. Note that in this paper we do not consider the additional reference images taken by the two side-facing cameras. We report details about the exact sequences used for training for each dataset in Table 1.

Baselines. We compare our approach both against structure-based and retrieval-based state-of-the-art meth-

	Method	*RobotCar Seasons*						*Extended CMU-Seasons*								
		Day-All			*Night-All*			*Urban*			*Suburban*			*Park*		
		Threshold Accuracy			Threshold Accuracy			Threshold Accuracy			Threshold Accuracy			Threshold Accuracy		
		0.25m 2°	0.5m 5°	5m 10°	0.25m 2°	0.5m 5°	5m 10°	0.25m 2°	0.5m 5°	5m 10°	0.25m 2°	0.5m 5°	5m 10°	0.25m 2°	0.5m 5°	5m 10°
Structure-based	CSL [66]	45.3	73.5	90.1	0.6	2.6	7.2	71.2	74.6	78.7	57.8	61.7	67.5	34.5	37.0	42.2
	AS [56]	35.6	67.9	90.4	0.9	2.1	4.3	-	-	-	-	-	-	-	-	-
	SMC [69]	**50.3**	**79.3**	**95.2**	**7.1**	**22.4**	45.3	**88.8**	**93.6**	**96.3**	**78.0**	**83.8**	89.2	**63.6**	**70.3**	77.3
Retrieval-based	FAB-MAP [16]	2.7	11.8	37.3	0.0	0.0	0.0	-	-	-	-	-	-	-	-	-
	NetVLAD [2]	6.4	26.3	90.9	0.3	2.3	15.9	12.2	31.5	89.8	3.7	13.9	74.7	2.6	10.4	55.9
	DenseVLAD [71]	7.6	31.2	91.2	1.0	4.4	22.7	14.7	36.3	83.9	5.3	18.7	73.9	5.2	19.1	62.0
	ToDayGAN [1]	7.6	31.2	91.2	2.2	10.8	**50.5**	-	-	-	-	-	-	-	-	-
Hierarchical	NV+SP [52]	**53.0**	**79.3**	95.0	5.9	17.1	29.4	**89.5**	**94.2**	**97.9**	**76.5**	**82.7**	**92.7**	**57.4**	64.4	**80.4**
	NV-r + S-D + H (Ours)	45.7	**78.0**	**95.1**	**22.3**	**61.8**	**94.5**	65.7	82.7	91.0	66.5	82.6	**92.9**	54.3	**71.6**	**84.1**

Table 2: **Localization results**. We report localization recalls in percent, for three translation and orientation thresholds (*high*, *medium*, and *coarse*) as in [58]. We highlight the **best** in red and **second-best** in blue performances for each threshold. Note that NetVLAD, ToDayGAN, and NV+SP all use pre-trained NetVLAD weights from Pittsburgh30k [2], while we retrained ours on other RobotCar sequences. We also include SMC, which uses additional semantic data and assumptions. For Extended CMU-Seasons, some methods did not provide results for the benchmark.

ods. Localization results for these methods were provided by the authors of the benchmark [58].

For structure-based methods, we compare our approach to Active Search (AS) [56] and City-Scale Localization (CSL) [66]. Both methods are direct 2D-3D matching techniques optimized for matching efficiency and robustness respectively, and have shown to deliver great accuracy in daytime conditions at a high precision threshold [58]. We also display results for Semantic Match Consistency (SMC) [69], which leverages semantic maps to filter outliers in the matching stage, and makes additional assumptions regarding the camera height and gravity vector.

We also compare our approach to retrieval-based methods, such as NetVLAD (pre-trained on Pittsburgh30k [2] with a VGG-16 [64] backbone), and to DenseVLAD [71]. For these methods, we simply approximate the query image camera pose by the pose of its retrieved top-ranked database image. Details about their configuration and implementation details can be found in the original benchmark [58]. Additionally for RobotCar Seasons, we report the results obtained by performing night-to-day image translation using a GAN architecture (ToDayGAN) [1], prior to running DenseVLAD. Lastly, we show the results obtained by Sarlin *et al*. [52], which is a hierachical approach using a pretrained NetVLAD backbone followed by SuperPoint [17] feature detection and local descriptors for 2D-3D matching (NV+SP). This method also uses co-visibility clusters to merge 3D points from neighbouring database images.

Metrics. We evaluate our approach using the same localization metric as [58]. Three precision thresholds are defined, accounting for both positional and rotational error. We refer to these thresholds as *high* (0.25m and 2°), *medium* (0.5m and 5°) and *coarse* (5m and 10°) precision. For each threshold, we report the localization recall in percent.

5.2. Large-Scale Localization

Having established our evaluation process, we now report the performance of our approach.

Training sets. For the NetVLAD retrieval backbone, we use different weights for both datasets. For RobotCar Seasons [44], we retrained NetVLAD on tuples extracted from other RobotCar sequences, featuring for daytime and nighttime images (see Table 1). Positive and negative tuples were assembled using the provided GPS and INS data. Note that these sequences do not overlap with the test set. For Extended CMU-Seasons [8], we built training samples using all the provided annotated training data from the *urban*, *suburban* and *park* slices. When training NetVLAD, we use hard-negative mining at every epoch, to obtain for each query the hardest subset of all possible negatives in the database.

Methods. As presented in Section 3, we run our hierarchical localization pipeline by first ranking each query with respect to the reference images. We use the normalized global image descriptors produced by NetVLAD (NV), and obtain the rankings using a simple dot product. To account for potential image retrieval errors, for every query we run the exhaustive matching step on each of the top-N nearest neighbors. The final predicted pose is picked as the one having the highest number of inliers in the RANSAC loop of the PnP. For RobotCar Seasons, we use $N = 15$ and for Ex-

Method	*Day-All* Threshold Accuracy			*Night-All* Threshold Accuracy		
	0.25m 2°	0.5m 5°	5m 10°	0.25m 2°	0.5m 5°	5m 10°
NV (pre-trained)	6.4	26.3	90.9	0.3	2.3	15.9
NV-r (re-trained)	4.1	17.8	86.9	2.4	11.4	84.6
NV-r + S-S + SP	**52.9**	**78.5**	93.8	10.9	32.7	87.4
NV-r + S-S + H	49.0	77.9	93.6	14.8	44.5	89.7
NV-r + S-D + SP	50.3	77.5	92.9	14.4	43.2	87.8
NV-r + S-D + H	45.7	78.0	**95.1**	**22.3**	**61.8**	**94.5**

Table 3: **Ablation Study** on the RobotCar Seasons dataset. We first show the improvements coming from using a re-trained NetVLAD (NV) [2] backbone. Then, we report localization performance using standard 'Sparse-to-Sparse' (S-S) matching using SuperPoint detections and two different descriptors: SuperPoint descriptors (S-S + SP) and Hypercolumn descriptors (S-S + H), as well as the results of our 'Sparse-to-dense' (S-D) matching using SuperPoint descriptors (S-D + SP) and Hypercolumn descriptors (S-D + H). We report localization recall in percent, for three translation and orientation thresholds.

	Dense Query Hypercolumn Descriptors	*Sparse Reference Hypercolumn Descriptors (offline)*	*Correspondence Maps (Exhaustive search)*	*Ratio Test (non-optimized)*	*PnP Solving*
Runtime (ms)	107.29	114.71	10.8	169.14	3.08

Table 4: **Runtime measurements.** We report the average runtimes for our sparse-to-dense matching approach on RobotCar Season, with 512×512 input images. Operations in italic are run for each of the top-ranked images.

tended CMU-Seasons, we use $N = 10$ because of the large amount of images to evaluate.

Implementation details. We use a Pytorch implementation of NetVLAD to compute the global image descriptors as well as the intermediate VGG-16 features used to compute the hypercolumns. As in [52], we reduce the dimensionality of all produced descriptors to a size of 1024 using PCA, learned on the reference set. When retraining NetVLAD on RobotCar Seasons and Extended CMU-Seasons, images are rescaled to a maximum size of 512 pixels, while preserving image ratio. At inference time, we again rescale images to a maximum size of 512 pixels for all datasets, both to compute the global image descriptors and to extract intermediate dense features. The offline point cloud triangulation and the online 2D-3D correspondences are done using the original images resolutions.

We use different ratio test values for each dataset. For RobotCar Seasons we use a factor of $f = 0.006$. For Extended CMU-Seasons we use a value of 0.12, as we found much more ambiguous matches and using selective thresholds were leading to a high number of rejections. As in [52], for both datasets, the RANSAC [20] loop stops when a pose has a minimum number of inliers of 15.

Performance. We run our experiments on a PC equipped with an Intel(R) Xeon(R) E5-2630 CPU (2.20GHz) CPU with 128GB of RAM and an NVIDIA GeForce GTX 1080Ti GPU. We pre-compute compressed global image descriptors for a faster image retrieval at inference time. Our main bottleneck in terms of computation times in our current implementation lies in the VGG-16 inference. As shown in [52], this part can be sped up using a teacher network with little loss in accuracy. Our ratio test method could also be replaced by a faster, more traditional non-maxima suppression scheme computed on GPU. The computation of the correspondence map is done on GPU through a convolution operation and takes on average 30ms in our implementation (depending on the input image resolution and ratio). We report the average measured runtimes in Table 4.

Results. We report the localization results in Table 2. Our method outperforms all baselines in very challenging scenarios such as nighttime for RobotCar Seasons. We also show significant improvements for the *park* scene of Extended CMU-Seasons, which is arguably the most difficult with strong changes in vegetation, at *medium* and *coarse* precision thresholds. For other categories, the performance is usually on par with state-of-the-art structure-based or hierarchical methods such as SMC [69] or NV+SP [52] respectively. On easier categories, such as *day-all* for RobotCar Seasons or *urban* for CMU, our approach is not as accurate as other feature-point based approaches, especially at a finer threshold. It is therefore more adapted to complex correspondence problems. On less challenging cases, the standard approach which relies on a detector with sub-pixel accuracy for the query image can still be more accurate.

5.3. Ablation Study

Having presented the results of our full pipeline, we now evaluate the impact of each element of our pipeline in the localization step. We run this ablation study on RobotCar Season [44] and report our results in Table 3.

NetVLAD backbones. We first discuss the impact of having a retrained image-retrieval backbone. As shown in Table 3, the pre-trained Pittsburgh30k [2] weights (NV) provides a good coarse pose estimation in daytime, but still very mild results at nighttime. We can already see that this will be a very limiting factor when performing 2D-3D matching, as the selected point cloud subsets will not be overlapping with the query image. When retraining NetVLAD (NV-r) on nighttime sequences from RobotCar,

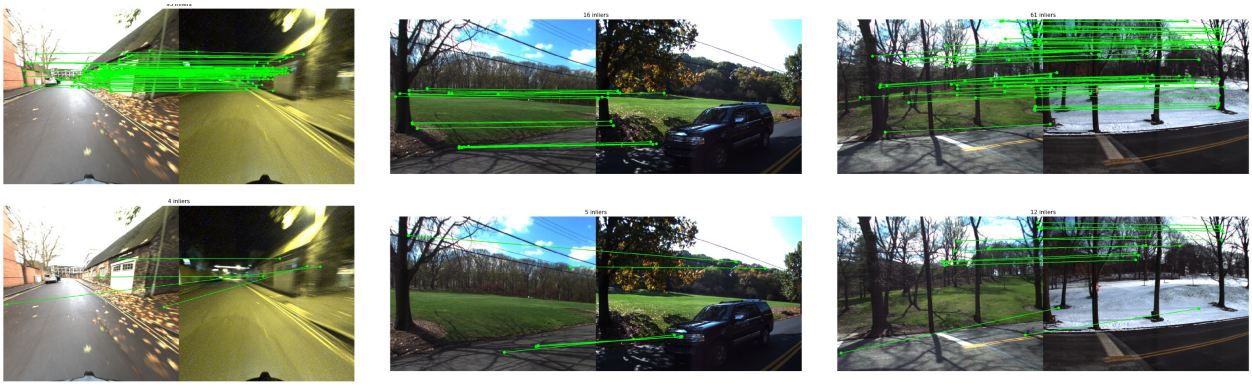


Figure 4: **Examples of inlier correspondences obtained using RANSAC+PnP.** Top-row shows correspondences obtained with our method, bottom row shows correspondences obtained with SuperPoint detection and descriptors.

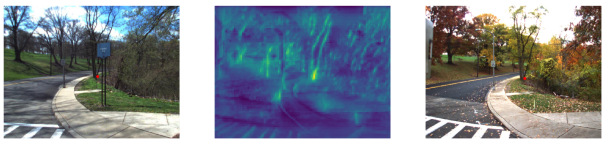


Figure 5: **Example of correlation map.** Left image shows a Superpoint in the reference image. The corresponding sparse hypercolumn descriptor is used to compute the correlation map (middle) and retrieve the 2D correspondent in the query image (right).

this gives a significant boost in performance, especially at a coarse precision level.

However, this is also tightly linked with the database spatial sampling: A dataset sampled much more sparsely would yield poor results at a coarse level even at daytime. We also tried retraining NetVLAD with a ResNet-50 [25] backbone, and / or a GeM [50] layer activation, but this always yielded slightly poorer retrieval results than a VGG-16 [64] network with a VLAD activation layer.

'Sparse-to-Sparse' Matching. We evaluate adding a subsequent camera pose estimation using 2D-3D matches coming from standard 'Sparse-to-Sparse' (S-S) matching using SuperPoint [17] detections and two different descriptors: SuperPoint descriptors (S-S + SP) and Hypercolumn descriptors (S-S + H). Both approaches (S-S + SP) and (S-S + H) allow to significantly improve the daytime results. For nighttime results, even if the performance improved, they remain limited compared to the daytime. We argue that this discrepancy between daytime and nighttime results comes from the difficulty to detect and match sparse feature points extracted from two images captured under very different conditions. This motivates our novel 'Sparse-to-Dense' matching approach. Finally, one can see that the aggregation of dense features into hypercolumns at different levels provides improvements. This shows the advantage of using hypercolumns for description rather than the Superpoint descriptors. This advantage is likely due to the large receptive fields of the hypercolumns computed by VGG, and the way they are learned to be condition-invariant.

'Sparse-to-Dense' Matching. We finally evaluate replacing the standard 'Sparse-to-Sparse' matching with our novel 'Sparse-to-Dense' matching for both Superpoint descriptors (S-D + SP) and Hypercolumn descriptors (S-D + H). As shown in Table 3, our novel approach is a way to partially remove the nighttime detection bottleneck: Compared to 'Sparse-to-Sparse' Hypercolumn matching (NV-r + S-S + H), our 'Sparse-to-Dense' Hypercolumn matching (NV-R + S-D + H) increases the recall by 7.5% and 17.3% for the *high* and *medium* thresholds respectively at nighttime.

6. Conclusion

We have introduced a novel hierarchical localization method, which reformulates the 2D-3D matching stage to improve long-term localization capabilities. We showed that breaking the paradigm of detecting feature points in both images to match, we can significantly improve the number of correct matches. While this approach was demonstrated in this paper in the context of localization, it is very likely to be useful for other applications.

Acknowledgement

This project has received funding from the Bosch Research Foundation (*Bosch Forschungsstiftung*). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research. The authors would also like to warmly thank the authors of the benchmark [58] for providing support with the evaluation tools. Vincent Lepetit is a senior member of the *Institut Universitaire de France* (IUF).

References

[1] A. Anoosheh, T. Sattler, R. Timofte, M. Pollefeys, and L. Van Gool. Night-To-Day Image Translation for Retrieval-Based Localization. *CoRR*, abs/1809.09767, 2018. 6

[2] R. Arandjelovic, P. Gronát, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN Architecture for Weakly Supervised Place Recognition. In *Conference on Computer Vision and Pattern Recognition*, pages 5297–5307, 2016. 2, 3, 4, 6, 7

[3] R. Arandjelovic and A. Zisserman. Three Things Everyone Should Know to Improve Object Retrieval. In *Conference on Computer Vision and Pattern Recognition*, pages 2911–2918, 2012. 2, 3, 5

[4] R. Arandjelovic and A. Zisserman. All About VLAD. In *Conference on Computer Vision and Pattern Recognition*, pages 1578–1585, 2013. 3

[5] R. Arandjelovic and A. Zisserman. Dislocation: Scalable Descriptor Distinctiveness for Location Recognition. In *Asian Conference on Computer Vision*, 2014. 2

[6] R. Arandjelovic and A. Zisserman. Visual Vocabulary with a Semantic Twist. In *Asian Conference on Computer Vision*, 2014. 3

[7] A. Babenko and V. S. Lempitsky. Aggregating Local Deep Features for Image Retrieval. In *International Conference on Computer Vision*, pages 1269–1277, 2015. 3, 4

[8] H. Badino, D. F. Huber, and T. Kanade. Visual Topometric Localization. *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 794–799, 2011. 5, 6

[9] V. Balntas, S. Li, and V. Prisacariu. Relocnet: Continuous Metric Learning Relocalisation Using Neural Nets. In *European Conference on Computer Vision*, 09 2018. 3

[10] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother. DSAC — Differentiable RANSAC for Camera Localization. In *Conference on Computer Vision and Pattern Recognition*, pages 2492–2500, 2017. 3

[11] E. Brachmann and C. Rother. Learning Less is More - 6D Camera Localization via 3D Surface Regression. *CoRR*, abs/1711.10228, 2017. 3

[12] M. Bui, S. Albarqouni, S. Ilic, and N. Navab. Scene Coordinate and Correspondence Learning for Image-Based Localization. In *British Machine Vision Conference*, 2018. 3

[13] M. Bujnak, Z. Kukelova, and T. Pajdla. New Efficient Solution to the Absolute Pose Problem for Camera with Unknown Focal Length and Radial Distortion. In *Asian Conference on Computer Vision*, 2010. 2, 4

[14] D. M. Chen, G. Baatz, K. Köser, S. S. Tsai, R. Vedantham, T. Pylvänäinen, K. Roimela, X. Chen, J. Bach, M. Pollefeys, B. Girod, and R. Grzeszczuk. City-Scale Landmark Identification on Mobile Devices. In *Conference on Computer Vision and Pattern Recognition*, pages 737–744, 2011. 2, 3

[15] S. Choudhary and P. J. Narayanan. Visibility Probability Structure from Sfm Datasets and Applications. In *European Conference on Computer Vision*, pages 130–143, 2012. 3

[16] M. J. Cummins and P. Newman. FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. *I. J. Robotics Res.*, 27:647–665, 2008. 6

[17] D. DeTone, T. Malisiewicz, and A. Rabinovich. Superpoint: Self-Supervised Interest Point Detection and Description. *CoRR*, abs/1712.07629, 2017. 2, 3, 4, 5, 6, 8

[18] M. Donoser and D. Schmalstieg. Discriminative Feature-To-Point Matching in Image-Based Localization. In *Conference on Computer Vision and Pattern Recognition*, pages 516–523, 2014. 3

[19] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24:381–395, 1981. 2

[20] M. A. Fischler and R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM*, 24:381–395, 1981. 4, 7

[21] H. Germain, G. Bourmaud, and V. Lepetit. Improving Nighttime Retrieval-Based Localization. 2018. 2, 3

[22] A. Gordo, J. Almazán, J. Revaud, and D. Larlus. End-To-End Learning of Deep Visual Representations for Image Retrieval. *International Journal of Computer Vision*, 124:237–254, 2017. 3, 4

[23] R. M. Haralick, C.-N. Lee, K. Ottenberg, and M. Nölle. Review and Analysis of Solutions of the Three Point Perspective Pose Estimation Problem. *International Journal of Computer Vision*, 13:331–356, 1994. 2, 4

[24] B. Hariharan, P. A. Arbeláez, R. B. Girshick, and J. Malik. Hypercolumns for Object Segmentation and Fine-Grained Localization. In *Conference on Computer Vision and Pattern Recognition*, pages 447–456, 2015. 2, 4

[25] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 8

[26] I. Heisterklaus, N. Qian, and A. Miller. Image-Based Pose Estimation Using a Compact 3D Model. In *International Conference on Consumer Electronics Berlin*, pages 327–330, 2014. 3

[27] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof. From structure-from-motion point clouds to fast location recognition. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2599–2606, 2009. 2, 3

[28] H. Jégou and O. Chum. Negative Evidences and Cooccurrences in Image Retrieval: the Benefit of PCA and Whitening. In *European Conference on Computer Vision*, 2012. 3

[29] H. Jégou, M. Douze, and C. Schmid. Product Quantization for Nearest Neighbor Search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33:117–128, 2011. 3

[30] Y. Kalantidis, C. Mellina, and S. Osindero. Cross-Dimensional Weighting for Aggregated Deep Convolutional Features. In *European Conference on Computer Vision*, 2016. 3, 4

[31] A. Kendall and R. Cipolla. Geometric Loss Functions for Camera Pose Regression with Deep Learning. In *Conference on Computer Vision and Pattern Recognition*, pages 6555–6564, 2017. 3

[32] A. Kendall, M. K. Grimes, and R. Cipolla. Posenet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization. In *International Conference on Computer Vision*, pages 2938–2946, 2015. 3

[33] L. Kneip, D. Scaramuzza, and R. Siegwart. A Novel Parametrization of the Perspective-Three-Point Problem for a Direct Computation of Absolute Camera Position and Ori-

entation. In *Conference on Computer Vision and Pattern Recognition*, pages 2969–2976, 2011. 4

[34] N. Kobyshev, H. Riemenschneider, and L. Van Gool. Matching Features Correctly through Semantic Understanding. In *International Conference on 3D Vision*, pages 472–479, 2014. 3

[35] Z. Kukelova, M. Bujnak, and T. Pajdla. Real-Time Solution to the Absolute Pose Problem with Unknown Radial Distortion and Focal Length. In *International Conference on Computer Vision*, pages 2816–2823, 2013. 2, 4

[36] V. Larsson, J. Fredriksson, C. Toft, and F. Kahl. Outlier Rejection for Absolute Pose Estimation with Known Orientation. In *British Machine Vision Conference*, 2016. 3

[37] V. Lepetit, F. Moreno-Noguer, and P. Fua. Epnp: An accurate o(n) solution to the pnp problem. *International Journal of Computer Vision*, 81:155–166, 2008. 2

[38] Y. Li, N. Snavely, and D. P. Huttenlocher. Location Recognition Using Prioritized Feature Matching. In *European Conference on Computer Vision*, pages 791–804, 2010. 2, 3

[39] Y. Li, N. Snavely, D. P. Huttenlocher, and P. Fua. Worldwide Pose Estimation Using 3D Point Clouds. In *Large-Scale Visual Geo-Localization*, 2012. 2, 3

[40] H. Lim, S. N. Sinha, M. F. Cohen, and M. Uyttendaele. Real-Time Image-Based 6-DOF Localization in Large-Scale Environments. In *Conference on Computer Vision and Pattern Recognition*, pages 1043–1050, 2012. 3

[41] L. Liu, H. Li, and Y. Dai. Efficient Global 2D-3D Matching for Camera Localization in a Large-Scale 3D Map. In *International Conference on Computer Vision*, pages 2391–2400, 2017. 2

[42] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004. 2

[43] S. Lynen, T. Sattler, M. Bosse, J. A. Hesch, M. Pollefeys, and R. Siegwart. Get Out of My Lab: Large-Scale, Real-Time Visual-Inertial Localization. In *Robotics: Science and Systems*, 2015. 3

[44] W. Maddern, G. Pascoe, C. Linegar, and P. Newman. 1 Year, 1000 Km: the Oxford Robotcar Dataset. *I. J. Robotics Res.*, 36:3–15, 2017. 5, 6, 7

[45] C. McManus, W. Churchill, W. P. Maddern, A. D. Stewart, and P. Newman. Shady Dealings: Robust, Long-Term Visual Localisation Using Illumination Invariance. In *IEEE International Conference on Robotics and Automation*, pages 901–906, 2014. 1

[46] S. Middelberg, T. Sattler, O. Untzelmann, and L. Kobbelt. Scalable 6-DOF Localization on Mobile Devices. In *European Conference on Computer Vision*, 2014. 1, 2, 3

[47] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han. Large-scale image retrieval with attentive deep local features. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3476–3485, 2017. 3

[48] Y. Ono, E. Trulls, P. Fua, and K. M. Yi. LF-Net: Learning Local Features from Images. In *NeurIPS*, 2018. 2, 3

[49] F. Radenovic, A. Iscen, G. Tolias, Y. S. Avrithis, and O. Chum. Revisiting Oxford and Paris: Large-Scale Image Retrieval Benchmarking. *CoRR*, abs/1803.11285, 2018. 3

[50] F. Radenovic, G. Tolias, and O. Chum. Fine-Tuning CNN Image Retrieval with No Human Annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 3, 4, 8

[51] A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson. Visual Instance Retrieval with Deep Convolutional Networks. *CoRR*, abs/1412.6574, 2014. 3, 4

[52] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk. From Coarse to Fine: Robust Hierarchical Localization at Large Scale. 2019. 2, 3, 4, 5, 6, 7

[53] P.-E. Sarlin, F. Debraine, M. Dymczyk, and R. Siegwart. Leveraging deep visual descriptors for hierarchical efficient localization. In *CoRL*, 2018. 2, 3

[54] T. Sattler, M. Havlena, F. Radenovic, K. Schindler, and M. Pollefeys. Hyperpoints and Fine Vocabularies for Large-Scale Location Recognition. In *International Conference on Computer Vision*, pages 2102–2110, 2015. 2, 3

[55] T. Sattler, M. Havlena, K. Schindler, and M. Pollefeys. Large-Scale Location Recognition and the Geometric Burstiness Problem. In *Conference on Computer Vision and Pattern Recognition*, pages 1582–1590, 2016. 3

[56] T. Sattler, B. Leibe, and L. Kobbelt. Improving Image-Based Localization by Active Correspondence Search. In *European Conference on Computer Vision*, 2012. 6

[57] T. Sattler, B. Leibe, and L. Kobbelt. Efficient & Effective Prioritized Matching for Large-Scale Image-Based Localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1744–1756, 2017. 2, 3

[58] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, F. Kahl, and T. Pajdla. Benchmarking 6DOF Outdoor Visual Localization in Changing Conditions. In *Conference on Computer Vision and Pattern Recognition*, June 2018. 1, 2, 3, 4, 5, 6, 8

[59] T. Sattler, C. Sweeney, and M. Pollefeys. On Sampling Focal Length Values to Solve the Absolute Pose Problem. In *European Conference on Computer Vision*, 2014. 2, 4

[60] T. Sattler, A. Torii, J. Sivic, M. Pollefeys, H. Taira, M. Okutomi, and T. Pajdla. Are Large-Scale 3D Models Really Necessary for Accurate Visual Localization? In *Conference on Computer Vision and Pattern Recognition*, page 10, July 2017. 2, 3, 4

[61] J. L. Schönberger and J.-M. Frahm. Structure-From-Motion Revisited. In *Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016. 5

[62] J. L. Schönberger, M. Pollefeys, A. Geiger, and T. Sattler. Semantic Visual Localization. *CoRR*, abs/1712.05773, 2017. 3

[63] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys. Pixelwise View Selection for Unstructured Multi-View Stereo. In *European Conference on Computer Vision*, 2016. 5

[64] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR*, abs/1409.1556, 2015. 4, 6, 8

[65] G. Singh and J. Košecká. *Semantically Guided Geo-Location and Modeling in Urban Environments*, pages 101–120. 2016. 3

[66] L. Svärm, O. Enqvist, F. Kahl, and M. Oskarsson. City-Scale Localization for Cameras with Known Vertical Direction. *IEEE Transactions on Pattern Analysis and Machine*

Intelligence, 39(7):1455–1461, 7 2017. 2, 3, 6

[67] L. Svärm, O. Enqvist, M. Oskarsson, and F. Kahl. Accurate Localization and Pose Estimation for Large 3D Models. In *Conference on Computer Vision and Pattern Recognition*, pages 532–539, 2014. 3

[68] H. Taira, M. Okutomi, T. Sattler, M. Cimpoi, M. Pollefeys, J. Sivic, T. Pajdla, and A. Torii. Inloc: Indoor Visual Localization with Dense Matching and View Synthesis. *CoRR*, abs/1803.10368, 2018. 3

[69] C. Toft, E. Stenborg, L. Hammarstrand, L. Brynte, M. Pollefeys, T. Sattler, and F. Kahl. Semantic Match Consistency for Long-Term Visual Localization. In *European Conference on Computer Vision*, 09 2018. 3, 6, 7

[70] G. Tolias, R. Sicre, and H. Jégou. Particular Object Retrieval with Integral Max-Pooling of CNN Activations. *CoRR*, abs/1511.05879, 2015. 3, 4

[71] A. Torii, R. Arandjelovic, J. Sivic, M. Okutomi, and T. Pajdla. 24/7 Place Recognition by View Synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40:257–271, 2015. 2, 3, 6

[72] F. Walch, C. Hazirbas, L. Leal-Taixé, T. Sattler, S. Hilsenbeck, and D. Cremers. Image-Based Localization Using LSTMs for Structured Feature Correlation. In *International Conference on Computer Vision*, pages 627–637, 2017. 2, 3

[73] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua. LIFT: Learned Invariant Feature Transform. In *European Conference on Computer Vision*, 2016. 2

[74] A. R. Zamir and M. Shah. Accurate Image Localization Based on Google Maps Street View. In *European Conference on Computer Vision*, 2010. 3

[75] B. Zeisl, T. Sattler, and M. Pollefeys. Camera Pose Voting for Large-Scale Image-Based Localization. In *International Conference on Computer Vision*, pages 2704–2712, 2015. 3

[76] W. Zhang and J. Kosecka. Image Based Localization in Urban Environments. *International Symposium on 3D Data Processing, Visualization, and Transmission*, pages 33–40, 2006. 3