S2DNet: Learning Accurate Correspondences for Sparse-to-Dense Feature Matching

Hugo Germain[1], Guillaume Bourmaud[2], and Vincent Lepetit[3]

[1] LIGM, École des Ponts, Univ Gustave Eiffel, CNRS, Marne-la-vallée, France
[2] Laboratoire IMS, Université de Bordeaux, France

Abstract. Establishing robust and accurate correspondences is a fundamental backbone to many computer vision algorithms. While recent learning-based feature matching methods have shown promising results in providing robust correspondences under challenging conditions, they are often limited in terms of precision. In this paper, we introduce S2DNet, a novel feature matching pipeline, designed and trained to efficiently establish both robust and accurate correspondences. By leveraging a sparse-to-dense matching paradigm, we cast the correspondence learning problem as a supervised classification task to learn to output highly peaked correspondence maps. We show that S2DNet achieves state-of-the-art results on the HPatches benchmark, as well as on several long-term visual localization datasets.

Keywords: Feature matching, classification, visual localization

1 Introduction

Establishing both accurate and robust correspondences across images is an underpinning step to many computer vision algorithms, such as Structure-from-Motion (SfM) [19,45,46,52], visual tracking [21,61] and visual localization [42,50, 51]. Yet, obtaining such correspondences in long-term scenarios where extreme visual changes can appear remains an unsolved problem, as shown by recent benchmarks [41, 53]. In particular, illumination (e.g. daytime to nighttime), cross-seasonal and structural changes are very challenging factors for feature matching.

The correspondences accuracy plays a major role in the performance of the aforementioned algorithms. Indeed, the noise perturbation experiment of Figure 1 (left) shows the highly damaging impacts of errors of a few pixels on visual localization. A traditional and very commonly used paradigm for feature matching between two images consists in detecting a set of keypoints [7,11,13,18,25,27,28, 35,44,62], followed by a description stage [6,8,11–13,25,31,32,35,47,48,62] in each image. Sparse sets of keypoints and their descriptors are then matched using for instance approximate nearest neighbours. This *sparse-to-sparse* matching approach has the main advantage of being both computationally and memory efficient. For long-term scenarios, this requires detecting repeatable keypoints and computing robust descriptors, which is very challenging.
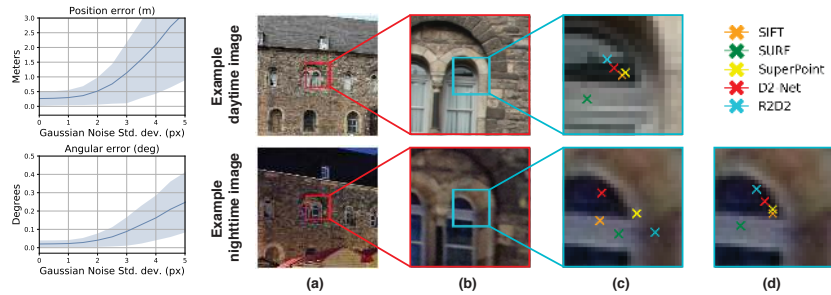
Fig. 1: Learning accurate correspondences. On the left, we report the impact of adding a gaussian noise of increasing variance on ground-truth 2D-3D correspondences for the task of visual localization, on Aachen Day-Night [41,43] images. This experiment highlights the importance of having very accurate correspondences, as offsets of a few pixels can lead to localization errors of several meters. Yet as shown on the right, sparse-to-sparse methods fail to make such accurate predictions. We show in (a) and (b) local regions of interest for a day-night image pair. In (c) [top], we display the keypoint detections being the nearest to the center of the patch in the daytime image for each detector; [bottom] we show the closest correspondent detected keypoints for each detector in the nighttime image. In (d), we show the correspondent image locations found by S2DNet in the nighttime image for daytime keypoint detections. S2DNet manages to find much more accurate correspondences than sparse-to-sparse methods.

With the advent of convolutional neural networks (CNNs), learning-based sparse-to-sparse matching methods have emerged, attempting to improve robustness of both detection and description stages in an end-to-end fashion. Several single-CNN pipelines [11,13,35] were trained with pixel-level supervision to jointly detect *and* describe interest points. These methods have yielded very competitive results especially in terms of number of correct matches, but fail to deliver highly accurate correspondences [13,35]. Indeed detecting the same keypoints repeatably across images is very challenging under strong visual changes, as illustrated in Figure 1 (right). Thus, the accuracy of such methods becomes highly reliant on the feature detector's precision and repeatability.

A recently proposed alternative [16] to solve the repeatable keypoint detection issue is to shift the sparse-to-sparse paradigm into a *sparse-to-dense* approach: Instead of trying to detect consistent interest points across images, feature detection is performed asymmetrically and correspondences are searched exhaustively in the other image. That way, all the information in the challenging image is preserved, allowing each pixel to be a potential correspondent candidate. Thanks to the popularization and development of GPUs, this exhaustive search can be done with a small computational overhead compared to other learning-based methods. This approach has showed to give competitive results [16] when trained in a weakly supervised fashion for the task of image retrieval.

In this paper, we reuse the sparse-to-dense idea while also addressing the accuracy issue. We introduce a novel feature matching pipeline which we name *S2DNet*.
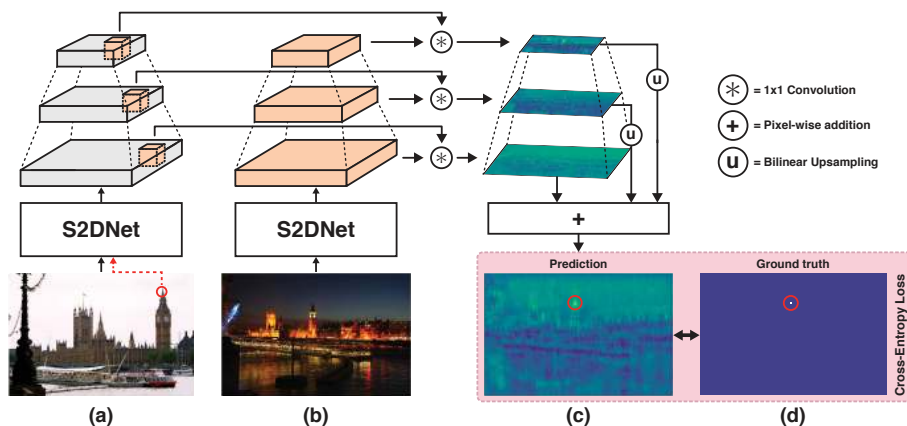
Fig. 2: S2DNet feature matching pipeline overview. Given an image and a set of detections coming from an off-the-shelf keypoint detector (a), we first extract a set of sparse multi-level descriptors with S2DNet. We then compute dense feature maps for a covisible image (b), and compute multi-level correspondence maps (c), which we aggregate using bilinear upsampling and addition. Correspondences can the be retrieved using a simple argmax operator. We explicitly train S2DNet to generate accurate and discriminative correspondence maps using a supervised classification approach (d).

We explicitly designed it to learn both robust and accurate correspondence maps. Instead of losses operating at a sparse descriptor level, we train S2DNet to learn both accurate and discriminative correspondence maps. By casting the feature matching problem as a classification problem, we give rich feedback on correspondences errors at training time. Compared to other approaches, we show that our pipeline generates much more accurate correspondences with most off-the-shelf feature detectors. Our evaluations show that we achieve state-of-the-art results on image matching and long-term visual localization benchmarks.

2   Related Work

Establishing 2D to 2D correspondences between images is a key step for many applications in computer vision, whose performance often directly rely on the quantity and accuracy of such correspondences [15, 59]. We can distinguish three categories for obtaining such correspondences, which either rely on a bilateral, no keypoint or asymmetrical detection stage.

Sparse-to-sparse feature matching. The most popular and studied approach for feature matching is a two-stage pipeline that first detects interest point locations and assigns a patch-based descriptor to each of them. Detection is applied on both images to be matched, and we refer to these *detect-then-describe* approaches as *sparse-to-sparse* feature matching methods. To perform keypoint

detection, a variety of both hand-crafted [7, 18, 25, 27] and learning-based [32, 62] detectors have been developed, each aiming to detect accurate keypoints in both a repeatable as well as illumination, scale and affine invariant fashion. For feature description, methods using histograms of local gradients [7, 9, 25, 37] or learning-based patch description [4, 32, 55, 56, 62] have been widely used.

Yet, when working on long-term scenarios where very strong visual changes can appear, such methods fail to give reliable correspondences [41], motivating the need for data-driven methods leveraging information beyond patch-level. Among them, end-to-end learning-based pipelines such as LIFT [62] propose to jointly learn the detection and description stages. Methods like LF-Net [32] or SuperPoint [11] learn detection and description in a self-supervised way, using spatial augmentation of images through affine transformations. With D2-Net [13], Dusmanu *et al.* showed that a single-branch CNN could both perform detection and description, in a paradigm referred to as *detect-and-describe*. Their network is trained in a supervised way with a contrastive loss on the deep local features, using ground-truth pixel-level correspondences provided by Structure-from-Motion reconstructions [24]. R2D2 [35] builds on the same paradigm and formulates the learning of keypoint reliability and repeatability together with the detection and description, using this time a listwise ranking loss.

In order to preserve the accuracy of their correspondences, both D2-Net [13] and R2D2 [35] use dilated convolutions. Still when looking at feature-matching benchmarks like HPatches [5], the mean matching accuracy at error thresholds of one or two pixels is quite low. This indicates that their detection stage is often off by a couple of pixels. As shown in Figure 1, these errors have direct repercussions on the subsequent localization or reconstruction algorithms.

Dense-to-dense feature matching. Dense-to-dense matching approaches get rid of the detection stage altogether by finding mutual nearest neighbors in dense feature maps. This can be done using densely extracted features from a pre-trained CNN, combined with guided matching from late layers to earlier ones [53]. NCNet [36] trains a CNN to search in the 4D space of all possible correspondences, with the use of 4D convolutions. While they can be trained with weak supervision, dense-to-dense approaches carry high computational cost and memory consumption which make them hardly scalable for computer vision applications. Besides, the quadratic complexity of this approach limits the resolution of the images being used, resulting in correspondences with low accuracy.

Sparse-to-dense feature matching. Very recently [16] proposed to perform the detection stage asymmetrically. In such setting, correspondences are searched exhaustively in the counterpart image, by running for instance a cross-correlation operation on dense feature maps with a sparse set of local hypercolumn descriptors. While this exhaustive search used to be a costly operation, it can now be efficiently computed on GPUs, using batched 1×1 convolutions. The main appeal for this approach is that under strong visual changes, the need for repeatability in keypoint detection is alleviated, allowing each pixel to be a detection. Instead, finding
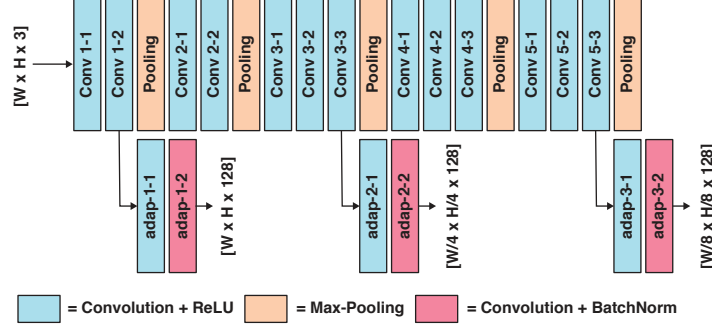
Fig. 3: S2DNet: Architecture overview. We feed images through a standard VGG-16 [49] backbone, and set three extraction points to process intermediate features. These features are sent to small, adaptation layers which help with the convergence and provide more condensed descriptors.

corresponding keypoint locations is left to the dense descriptor map. Preliminary work on sparse-to-dense matching [16] has shown that reusing intermediate CNN representations trained with weak supervision for the task of image retrieval can lead to significant gains in performance for visual localization. However, these features are not explicitly learned for feature matching, and thus fail to give pixel-accurate correspondences. We will refer to this method as S2DHM, for Sparse-to-Dense Hypercolumn Matching, in the rest of the paper.

In this paper, we propose to explicitly learn correspondence maps for the task of pixel-level matching.

3 Method

In this section we introduce and describe our novel sparse-to-dense feature-matching pipeline, which we call S2DNet.

3.1 The Sparse-to-Dense paradigm

Given an image pair $(\mathtt{I}_A, \mathtt{I}_B)$, our goal is to obtain a set of 2D correspondences which we write as $\{(\mathbf{p}_A^n, \mathbf{p}_B^n)\}_{n=1}^N$. Let us consider the case where a feature detector (*e.g.* the SuperPoint detector [11]) has been applied on image A, producing a set of N keypoints $\{\mathbf{p}_A^n\}_{n=1}^N$. In this case, the feature matching problem reduces to a *sparse-to-dense* matching problem of finding a correspondent $\mathbf{p}_B^n$ in image B for each detection $\mathbf{p}_A^n$. We propose to cast this correspondence learning problem as a supervised classification task by restricting the set of admissible locations to the pixel coordinates of $\mathtt{I}_B$. This leads to the following categorical distribution:

$$p\left(\mathbf{p}_B^n | \mathbf{p}_A^n, \mathtt{I}_A, \mathtt{I}_B, \Theta\right) = \frac{\exp\left(\mathtt{C}_n[\mathbf{p}_B^n]\right)}{\sum_{\mathbf{q} \in \Omega} \exp\left(\mathtt{C}_n[\mathbf{q}]\right)} \,, \tag{1}$$

where $\mathtt{C}_n$ is a correspondence map of the size of $\mathtt{I}_\mathrm{B}$ produced by our novel S2DNet matching pipeline and Ω is the set of pixel locations of $\mathtt{I}_\mathrm{B}$. S2DNet takes as input $\mathbf{p}_\mathrm{A}^n$, $\mathtt{I}_\mathrm{A}$, $\mathtt{I}_\mathrm{B}$ and its parameters Θ.

3.2   S2DNet matching pipeline

We introduce S2DNet, a pipeline built specifically to perform sparse-to-dense matching which we illustrate in Figure 2. Given a pair of images $(\mathtt{I}_\mathrm{A}, \mathtt{I}_\mathrm{B})$, we apply a convolutional backbone $\mathcal{F}$ on both images using shared network weights *i.e.* $\{\mathtt{H}_\mathrm{A}^m\}_{m=1}^M = \mathcal{F}(\mathtt{I}_\mathrm{A}; \Theta)$ and $\{\mathtt{H}_\mathrm{B}^m\}_{m=1}^M = \mathcal{F}(\mathtt{I}_\mathrm{B}; \Theta)$, where $\{\mathtt{H}_\mathrm{A}^m\}_{m=1}^M$ and $\{\mathtt{H}_\mathrm{B}^m\}_{m=1}^M$ correspond to intermediate feature maps extracted at multiple levels (see Figure 3). Θ denotes the parameters of $\mathcal{F}$. Such representations are sometimes referred to as hypercolumns [17,39]. While the earlier layers encode little semantic meaning, they preserve high-frequency local details which is crucial for retrieving accurate keypoints. Conversely in the presence of max-pooling, later layers loose in resolution but benefit from a wider receptive field and thus context.

For each detected keypoint $\mathbf{p}_\mathrm{A}^n$ in $\mathtt{I}_\mathrm{A}$, we extract a set of sparse descriptors in the dense intermediate feature maps $\mathtt{H}_\mathrm{A}^m$ and compute the dense correspondence map $\mathtt{C}_n$ against $\mathtt{H}_\mathrm{B}^m$, by processing each level independently, in the following way:

$$\mathtt{C}_n = \sum_{m=1}^M \mathcal{U}\left(\mathtt{H}_\mathrm{A}^m\left[\mathbf{p}_\mathrm{A}^{n,m}\right] * \mathtt{H}_\mathrm{B}^m\right) , \tag{2}$$

where $\mathcal{U}$ refers to the bilinear upsampling operator to $\mathtt{I}_\mathrm{B}$ resolution, $\mathbf{p}_\mathrm{A}^{n,m}$ corresponds to downscaling the 2D coordinates $\mathbf{p}_\mathrm{A}^n$ to the resolution of $\mathtt{H}_\mathrm{A}^m$, and $*$ is the 1×1 convolution operator.

3.3   Training-time

While state of the art approaches employ either a local contrastive or a listwise ranking loss [13,32,35] to train their network, we directly optimize for the task of sparse-to-dense correspondence retrieval by maximizing the log-likelihood in eq.(1) which results in a single multi-class cross-entropy loss. From a practical point of view, for every training sample, this corresponds to computing the softmaxed correspondence map and evaluate the cross-entropy loss using the ground truth correspondence $\mathbf{p}_\mathrm{B}^n$. This strongly penalizes wrong predictions regardless of their closeness to the ground-truth, forces the network to generate highly localized and peaked predictions and helps computing accurate correspondences.

3.4   Test-time

At test-time, to retrieve the correspondences in $\mathtt{I}_\mathrm{B}$, we proceed as follows for each detected keypoint $\mathbf{p}_\mathrm{A}^n$:

$$\mathbf{p}_\mathrm{B}^{n*} = \underset{\mathbf{p}_\mathrm{B}^n}{\operatorname{argmax}}\, p\left(\mathbf{p}_\mathrm{B}^n | \mathbf{p}_\mathrm{A}^n, \mathtt{I}_\mathrm{A}, \mathtt{I}_\mathrm{B}, \Theta\right) = \underset{\mathbf{p}}{\operatorname{argmax}}\, \mathtt{C}_n\left[\mathbf{p}\right] , \tag{3}$$

where $\mathtt{C}_n = \mathrm{S2DNet}(\mathbf{p}_A^n, \mathtt{I}_A, \mathtt{I}_B; \Theta)$. By default, S2DNet does not apply any type of filtering and delivers one correspondence for each detected keypoint in the source image. Since we do not explicitly deal with co-visibility issues, we filter out some ambiguous matches if the following condition is not satisfied:

$$ p\left(\mathbf{p}_B^{n*} | \mathbf{p}_A^n, \mathtt{I}_A, \mathtt{I}_B, \Theta\right) > \tau\,, \tag{4} $$

where τ is a threshold between 0 and 1.

3.5   S2DNet architecture

As [13, 16], we use a VGG-16 [49] architecture as our convolutional backbone. We place our intermediate extraction points at three levels, in `conv_1_2`, `conv_3_3` and `conv_5_3`, after the ReLU activations. Note that `conv_1_2` comes before any spatial pooling layer, and thus preserves the full image resolution. To both help with the convergence and reduce the final descriptors sizes, we feed these intermediate tensors to adaptation layers. They consist of two convolutional layers and a final batch-normalization [20] activation, with an output size of 128 channels. Refer to Figure 3 for an illustration of our architecture.

3.6   Differences with Sparse-to-Dense Hypercolumn Matching [16]

Sparse-to-Dense Hypercolumn Matching (S2DHM) [16] described a weakly supervised approach to learn hypercolumn descriptors and efficiently obtain correspondences using the sparse-to-dense paradigm. In this paper, we propose a supervised alternative which aims at directly learning accurate correspondence maps. As we will show in our experiments, this leads to significantly superior results. Moreover, in its pipeline S2DHM upsamples and concatenates intermediate feature maps before computing correspondence maps. In comparison, S2DNet computes correspondence maps at multiple levels before merging the results by addition. We will later show that the latter approach is much more memory and computationally efficient.

4   Experiments

In this section, we evaluate S2DNet on several challenging benchmarks. We first evaluate our approach on a commonly used image matching benchmark, which displays changes in both viewpoint and illumination. We then evaluate the performance of S2DNet on long-term visual localization tasks, which display even more severe visual changes.

4.1   Training data

We use the same training data as D2-Net [13] to train S2DNet, which comes from the MegaDepth dataset [24]. This dataset consists of 196 outdoor scenes and

$1,070,568$ images, for which *SfM* was run with COLMAP [45,46] to generate a sparse 3D reconstruction. A depth-check is run using the provided depth maps to remove occluded pixels. As D2-Net, we remove scenes which overlap with the PhotoTourism [1,54] test set. Compared to D2-Net and to provide strong scale changes, we train S2DNet on image pairs with an arbitrary overlap. At each training iteration, we extract random crops of size 512×512, and randomly sample a maximum of 128 pixel correspondences. We train S2DNet for 30 epochs using Adam [22]. We use an initial learning rate of 10^{-3} and apply a multiplicative decaying factor of $e^{-0.1}$ at every epoch.

4.2   Image Matching

We first evaluate our method on the popular image matching benchmark HPatches [5]. We use the same 108 sequences of images as D2-Net [13], each sequence consisting of 6 images. These images either display changes in illumination (for 52 sequences) or changes in viewpoint (for 56 sequences). We consider the first frame of each sequence to be the reference image to be matched against every other, resulting in 540 pairs of images to match.
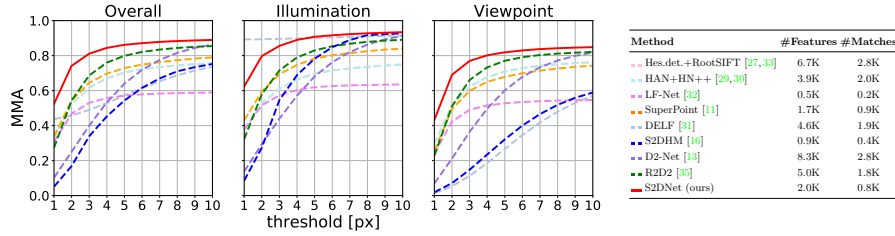
Evaluation protocol. We apply the SuperPoint [11] keypoint detector on the first image of each sequence. For each subsequent pair of images, we perform sparse-to-dense matching using S2DNet (see section 3.4). Additionally, we filter out correspondences which do not pass the cyclic check of matching back on their source pixel, which is equivalent to performing a mutual nearest-neighbor verification as it is done with D2-Net [13] and R2D2 [35].

   We compute the number of matches which fall under multiple reprojection error thresholds using the ground-truth homographies provided by the dataset, and report the Mean Matching Accuracy (or MMA) in Figure 4.

   We compare S2DNet to multiple sparse-to-sparse matching baselines. We report the performance of RootSIFT [3,33] with a Hessian Affine detector [27] (Hes.det. + RootSIFT), HardNet++ [29] coupled with HesAffNet regions [30] (HAN + HN++), DELF [31], LF-Net [32], SuperPoint [11], D2-Net [13] and R2D2 [35]. We also include results from the sparse-to-dense method S2DHM [16].

Results. We find that the best results were achieved when combining Super-Point [11] with a threshold of $\tau = 0.20$ (see Equation 4), which are the results reported in Figure 4. We experimentally found that above this threshold, some sequences obtain very few to no correspondence at all, which biases the results. We show that overall our method outperforms every baselines at any reprojection threshold. The gain in performance is particularly noticeable at thresholds of 1 and 2 pixels, indicating the correspondences we predict tend to be much more accurate. DELF [31] achieves competitive results under changes in illumination, which can be explained by the fact that keypoints are sampled on a fixed grid and that the images undergo no changes in viewpoint. On the other hand, it performs poorly under viewpoint changes.

Method	#Features	#Matches
Hes.det.+RootSIFT [27,33]	6.7K	2.8K
HAN+HN++ [29,30]	3.9K	2.0K
LF-Net [32]	0.5K	0.2K
SuperPoint [11]	1.7K	0.9K
DELF [31]	4.6K	1.9K
S2DHM [16]	0.9K	0.4K
D2-Net [13]	8.3K	2.8K
R2D2 [35]	5.0K	1.8K
S2DNet (ours)	2.0K	0.8K

Fig. 4: **HPatches Mean Matching Accuracy (MMA) comparison.** We report in this table the best results for S2DNet, obtained when combined with SuperPoint detections. S2DNet outperforms all other baselines, especially at thresholds of one or two pixels. This study highlights the power of working in a sparse-to-dense setting, where every pixel in the target image becomes a candidate keypoint.

Keypoint detector influence. We run an ablation study to evaluate the impact of different feature detectors, confidence thresholds as well as using a sparse-to-sparse approach, and report the results in Table 1 (left). We find that S2DNet tends to work best when combined with SuperPoint [11]. We also experimentally find $\tau = 0.2$ to be a good compromise of correspondence rejection while also maintaining a high number of matches.
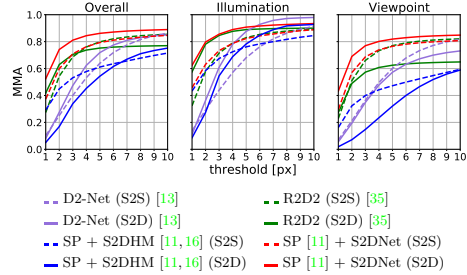
Sparse-to-sparse vs. Sparse-to-dense. We find that using S2DNet in a sparse-to-sparse setting (*i.e.* applying a detector on the image undergoing illumination or viewpoint changes) damages the results (see Table 1, left). This phenomenon translates the errors made by keypoint detectors, and motivates the sparse-to-dense setting. S2DNet efficiently leverages this paradigm and can find corresponding keypoints that would not have been detected otherwise. Conversely, we study the impact of using sparse-to-sparse learning-based methods D2-Net [13] and R2D2 [35] in a sparse-to-dense setting (see Table 1, right). We find that using the sparse-to-dense paradigm systematically improves their performance under illumination changes, where images are aligned. This suggests that their descriptor maps are robust to illumination perturbations. On the other hand, performance is damaged for both methods under viewpoint changes, suggesting that their descriptor maps are not highly localized and discriminative. Concerning S2DHM, which was trained in a weakly supervised manner, running it in a sparse-to-sparse setting improves the accuracy. This highlights the importance of our main contribution, *i.e.* casting the sparse-to-dense matching problem as a supervised classification task.

4.3 Long-Term Visual Localization

We showed that S2DNet provides correspondences which are overall more accurate than other baselines. We will now study its impact for the task of visual localization under challenging conditions. We report visual localization results

Detector	Matching	τ	MMA@1	MMA@2	MMA@3	MMA@10
Harris [18]	S2D	0.20	**0.511**	**0.733**	**0.805**	**0.888**
	S2D	0.0	0.441	0.626	0.690	0.787
	S2S	-	0.278	0.464	0.565	0.763
SURF [7]	S2D	0.20	**0.511**	**0.742**	**0.823**	**0.902**
	S2D	0.0	0.436	0.639	0.718	0.828
	S2S	-	0.302	0.506	0.619	0.829
SIFT [25]	S2D	0.20	**0.487**	**0.700**	**0.771**	**0.851**
	S2D	0.0	0.441	0.626	0.690	0.787
	S2D	-	0.386	0.559	0.642	0.818
SuperPoint [11]	S2D	0.20	**0.563**	**0.747**	**0.815**	**0.895**
	S2D	0.0	0.469	0.623	0.686	0.788
	S2S	-	0.373	0.599	0.709	0.847
D2-Net [13]	S2D	0.20	**0.467**	**0.716**	**0.805**	**0.911**
	S2D	0.0	0.330	0.522	0.604	0.764
	S2S	-	0.118	0.285	0.425	0.777
R2D2 [35]	S2D	0.20	**0.478**	**0.715**	**0.799**	**0.901**
	S2D	0.0	0.341	0.522	0.598	0.746
	S2S	-	0.316	0.546	0.652	0.819

(a) S2S vs. S2D - S2DNet descriptors

(b) S2S vs. S2D - Other descriptors

Table 1: Ablation study on HPatches. In **(a)**, we evaluate the performance of several detectors in both a sparse-to-dense (S2S) and sparse-to-sparse (S2S) setting using S2DNet descriptors. We find that S2DNet works best in the S2D setting, coupled with SuperPoint (SP) [11] detections, and a confidence threshold of $\tau = 0.20$. In **(b)**, we study the impact of using sparse-to-sparse learning-based methods in a sparse-to-dense setting. Results lead to the conclusion that D2-Net [13] and R2D2 [35] descriptor maps are robust to illumination changes but not highly discriminative locally.

under day-night changes and complex indoor scenes.

Datasets. We evaluate our approach on two challenging outdoor localization datasets which feature day-to-night changes, and one indoor dataset. The first dataset is Aachen Day-Night [41, 43]. It features 4,328 daytime reference images taken with a handheld smartphone, for which ground truth camera poses are provided. The dataset also provides a 3D reconstruction of the scene [41], built using SIFT [25] features and *SfM*. The evaluation is done on 824 daytime and 98 nighttime images taken in the same environment. The second dataset is RobotCar Seasons [26]. It features 6,954 daytime reference images taken with a rear-facing camera mounted on a car driving through Oxford. Similarly, ground truth camera poses and a sparse 3D model of the world is provided [41] and we localize 3,978 images captured throughout a year. These images do not only exhibit nighttime conditions, but also cross-seasonal evolutions such as snow or rain. Lastly, we evaluate our pipeline on the challenging InLoc [53, 60] dataset. This indoor dataset is difficult because of its large scale, illumination and long-term changes as well as the presence of repetitive patterns such as corridors (see Figure 5). It contains $9,972$ database and 356 high-resolution query images, as well as dense depth maps which can be used to perform dense pose verification. We report for each datasets the pose recall at three position and orientation thresholds for daytime and nighttime query images, as per [41].

Method	InLoc (fixed pipeline)		
	Threshold Accuracy		
	0.25m 2°	0.5m 5°	5m 10°
Direct PE - Aff. RootSIFT [27, 33]	18.5	26.4	30.4
Direct PE - D2-Net [13]	27.7	40.4	**48.6**
Direct PE - S2DNet (ours)	**29.3**	**40.9**	48.5
Sparse PE - Aff. RootSIFT [27, 33]	21.3	32.2	44.1
Sparse PE - D2-Net [13]	35.0	48.6	62.6
Sparse PE - S2DNet (ours)	**35.9**	**49.0**	**63.1**
Sparse PE + Dense PV - Aff. RootSIFT [27, 33]	29.5	42.6	54.5
Sparse PE + Dense PV - D2-Net [13]	38.0	**56.5**	65.4
Sparse PE + Dense PV - S2DNet (ours)	**39.4**	53.5	**67.2**
Dense PE + Dense PV - InLoc [53]	38.9	56.5	69.9

Method	Aachen Day-Night (fixed pipeline)		
	Threshold Accuracy		
	0.25m 2°	0.5m 5°	5m 10°
RootSIFT [33]	3.7	52.0	65.3
HAN+HN [30]	37.8	54.1	75.5
SuperPoint [11]	42.8	57.1	75.5
DELF [31]	39.8	61.2	85.7
D2-Net [13]	44.9	66.3	**88.8**
R2D2 [35]*	**45.9**	66.3	**88.8**
S2DNet (ours)	**45.9**	**68.4**	88.8

Table 2: InLoc [53] (left) and Local Features Benchmark [41] (right) results. We report localization recalls in percent, for three translation and orientation thresholds. On InLoc, S2DNet outperforms both baselines at the finest threshold for the sparse categories. We also include Dense PE baseline results for reference. R2D2 authors did not provide results on this benchmark. On the local features benchmark (a pre-defined localization pipeline), S2DNet achieves state-of-the-art results at the medium precision threshold. Due to the relatively small number of query images however, recent methods like D2-Net and R2D2 are saturating around the same performance. *Note that R2D2 was trained on Aachen database images.

Indoor Localization. The InLoc [53] localization benchmark comes with a pre-defined code base and several pipelines for localization. The first one is called Direct Pose Estimation (Direct PE) and performs hierarchical localization using the set of top-ranked database images obtained using image retrieval, followed by P3P-LO-RANSAC [14, 23]. The second variant applies an intermediate spatial verification step [34] to reject outliers, referred to as (Sparse PE). On top of this second variant, Dense Pose Verification (Dense PV) can be applied to re-rank pose candidates by using densely extracted RootSIFT [33] features. In each variant, we use S2DNet to generate 2D-2D correspondences between queries and database images, which are then converted to 2D-3D correspondences using the provided dense depth maps. We use a SuperPoint [11] detector and mutual nearest-neighbour filtering.

InLoc localization results are reported in Table 2. We compare our approach to the original InLoc baseline which uses affine covariant [27] detections and RootSIFT [33] descriptors, as well as results provided by D2-Net [13]. We find that S2DNet outperforms both sparse baselines at the finest threshold, and is on par with other methods at the medium and coarse thresholds. In the sparse setting, best results are achieved when combined with geometrical and dense pose verification (Sparse PE + Dense PV). In addition we include localization results that were computed by the benchmark authors using dense-to-dense feature matching (Dense PE). Due to the nature of our pipeline and the very high memory and computational consumption of this variant, we choose to limit

		RobotCar Seasons						Aachen Day-Night					
		Day-All			Night-All			Day			Night		
	Method	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°
Structure-based	CSL [50]	45.3	73.5	90.1	0.6	2.6	7.2	52.3	80.0	94.3	24.5	33.7	49.0
	AS [40]	35.6	67.9	90.4	0.9	2.1	4.3	57.3	83.7	96.6	19.4	30.6	43.9
	SMC [57] *	50.3	79.3	95.2	7.1	22.4	45.3	-	-	-	-	-	-
Retrieval-based	FAB-MAP [10]	2.7	11.8	37.3	0.0	0.0	0.0	0.0	0.0	4.6	0.0	0.0	0.0
	NetVLAD [2]	6.4	26.3	90.9	0.3	2.3	15.9	0.0	0.2	18.9	0.0	2.0	12.2
	DenseVLAD [58]	7.6	31.2	91.2	1.0	4.4	22.7	0.0	0.1	22.8	0.0	2.0	14.3
Hierarchical	HF-Net [38]	53.0	79.3	95.0	5.9	17.1	29.4	79.9	88.0	93.4	40.8	56.1	74.5
	S2DHM [16] *	45.7	78.0	95.1	22.3	61.8	94.5	56.3	72.9	90.9	30.6	56.1	78.6
	D2-Net [13]	**54.5**	80.0	95.3	**20.4**	40.1	55.0	**84.8**	**92.6**	**97.5**	43.9	66.3	85.7
	S2DNet (ours)	53.9	**80.6**	**95.8**	14.5	**40.2**	**69.7**	84.3	90.9	95.9	**46.9**	**69.4**	**86.7**

Table 3: Localization results. We report localization recalls in percent, for three translation and orientation thresholds (*high*, *medium*, and *coarse*) as in [41]. We put in bold the **best** and underline the second-best performances for each threshold. S2DNet outperforms every baseline in nighttime conditions, except at the finest threshold of RobotCar Seasons. This can be explained by the extreme visual changes and blurriness that these images undergo. At daytime, S2DNet performance is on par with D2-Net [13]. *Note that S2DHM [16] was trained directly on RobotCar sequences, which explains the high nighttime performance. SMC [57] also uses additional semantic data and assumptions. R2D2 [35] authors did not provide localization results on these benchmarks.

our study to sparse correspondence methods. It is interesting to note however that S2DNet outperforms the original (Dense PE + Dense PV) InLoc baseline at the finest precision threshold, using a much lighter computation.

Day-Night Localization. We report day-night localization results with S2DNet using two localization protocols. Localization results reported in Table 2 show that S2DNet achieves state-of-the-art results, outperforming all other methods at the medium precision threshold. It is important to note that R2D2 was finetuned on Aachen database images.

We then report in Table 3 localization results using a hierarchical approach, similar to [16,38,41]. Contrary to Table 2, these results do not allow to compare the keypoint matching approaches alone since localization pipelines are different. Even the comparison with D2Net is difficult to interpret since their full localization pipeline was not released. Still, S2DNet achieves state-of-the-art results in Aachen nighttime images, and outperforms all baselines that were not trained on RobotCar nighttime images at medium and coarse precision thresholds. At daytime, where detecting repeatable and accurate keypoints is easier, S2DNet is on par with other learning-based methods. At the finest nighttime RobotCar threshold it is likely that S2DNet features struggle to compute accurate corre-

Method	Network Backbone	Descriptor Size	Forward pass on I_q	Detection step on I_q	Matching per keypoint	Total online computational time		
			t_A	t_B	t_C	$t_A + t_B + N \times K \times t_C$		
						$N=1$ $K=1000$	$N=5$ $K=1000$	$N=15$ $K=1000$
D2-Net [13]	VGG-16	512	17.8ms	5,474s	0.4μs	5,492s	5,494s	5,498s
R2D2 [35]	L2-Net	128	19.1ms	479.6ms	0.2μs	0,499s	**0,499s**	**0,501s**
S2DHM [16]	VGG-16	2048	326ms	-	0.33ms	0,656s	1,976s	5,276s
S2DNet	VGG-16 + adap.	3×128	28.2ms	-	0.31ms	**0,338s**	1,578s	4,678s

Table 4: Computational Time Study for visual localization. We compare the time performance of our method against other learning-based approaches in a visual localization scenario. For a given query image I_q and N reference images of size 1200×1600 with K detections each, we report the average measured time to perform image matching against each of them. In this standard setting, keypoint locations and descriptors have already been extracted offline from the reference images.

spondences, which can be explained by the extreme visual changes these images undergo (see Figure 5). Overall, this study shows that S2DNet achieves better performance in particularly challenging conditions such as nighttime, compared to other sparse-to-sparse alternatives.

5   Discussion

Runtime performance. To compare S2DNet against state-of-the-art approaches, we time its performance for the scenario of visual localization. We run our experiments on a machine equipped with an Intel(R) Xeon(R) E5-2630 CPU at 2.20GHz, and an NVIDIA GeForce GTX 1080Ti GPU. We report the results in Table 4. In a localization setting, we consider the keypoint detection and description step to be pre-computed offline for reference images. Thus for an incoming query image, only sparse-to-sparse methods need to perform the keypoint detection and descriptor extraction step. We find this very step to be the bottleneck of learning-based methods like D2-Net [13] or R2D2 [35]. Indeed, these methods are slowed down by the non-maxima suppression operations, which are in addition run on images of multiple scales. For S2DHM [16] and S2DNet, no keypoint detection is performed on the incoming query image and most of the computation lies in the keypoint matching step. As expected however, the matching step is much more costly for these sparse-to-dense methods. Still, for 1000 detections and 1 retrieved image, S2DNet is the fastest method while for15 retrieved image, it is on par with D2-Net.

Current limitations of the sparse-to-dense paradigm. One limitation of our current sparse-to-dense matching formulation appears for the task of multiview 3D reconstruction. Indeed, the standard approach to obtain features tracks consists in 1) detecting and describing keypoints in each image, 2) matching pairs
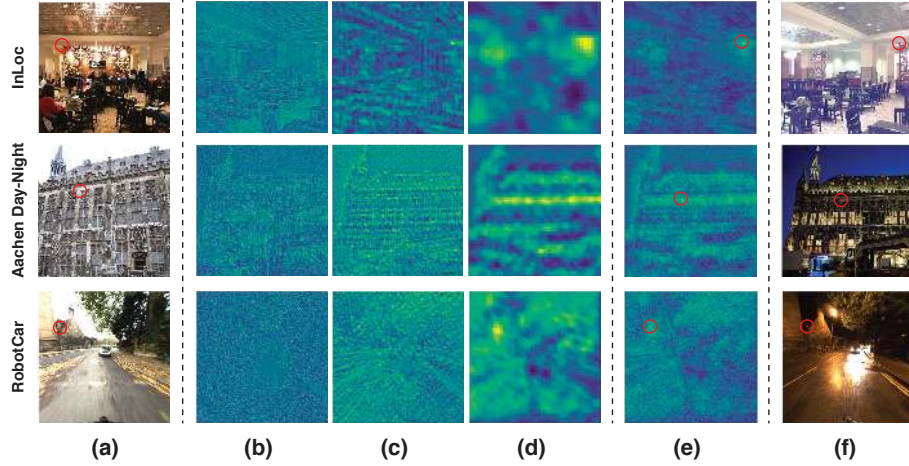
Fig. 5: Correspondence maps examples. From left to right: Reference image with a keypoint detection (a), intermediate correspondence maps predicted by S2DNet (b, c, d), aggregated pre-softmax correspondence map (e) and retrieved correspondent in the query image (f). From top to bottom: images from InLoc [53], Aachen Day-Night [43] and RobotCar Seasons [26].

of images using the previously extracted keypoints descriptors and 3) creating tracks from these matches. In our S2D matching paradigm, every pixel becomes a detection candidate which is not compatible with the standard 3D reconstruction pipeline previously described. This limitation opens novel directions of research for rethinking the standard tracks creation pipeline and enabling the use of S2D matching in 3D reconstruction frameworks.

6   Conclusion

In this paper we presented S2DNet, a new sparse-to-dense learning-based keypoint matching architecture. In contrast to other sparse-to-sparse methods we showed that this novel pipeline achieves superior performance in terms of accuracy, which helps improve subsequent long-term visual localization tasks. Under visually challenging conditions, S2DNet reaches state-of-the-art performance for image matching and localization, and advocates for the development of sparse-to-dense methods.

Acknowledgement

References

1. Phototourism Challenge, CVPR 2019 Image Matching Workshop (2019) 8
2. Arandjelovic, R., Gronát, P., Torii, A., Pajdla, T., Sivic, J.: NetVLAD: CNN Architecture for Weakly Supervised Place Recognition. In: Conference on Computer Vision and Pattern Recognition (2016) 12
3. Arandjelovic, R., Zisserman, A.: Three Things Everyone Should Know to Improve Object Retrieval. In: Conference on Computer Vision and Pattern Recognition (2012) 8
4. Balntas, V., Johns, E., Tang, L., Mikolajczyk, K.: PN-Net: Conjoined Triple Deep Network for Learning Local Image Descriptors. In: arXiv Preprint (2016) 4
5. Balntas, V., Lenc, K., Vedaldi, A., Mikolajczyk, K.: HPatches: A Benchmark and Evaluation of Handcrafted and Learned Local Descriptors. In: Conference on Computer Vision and Pattern Recognition (2017) 4, 8
6. Balntas, V., Riba, E., Ponsa, D., Mikolajczyk, K.: Learning Local Feature Descriptors with Triplets and Shallow Convolutional Neural Networks. In: British Machine Vision Conference (2016) 1
7. Bay, H., Tuytelaars, T., Gool, L.V.: SURF: Speeded Up Robust Features. In: European Conference on Computer Vision (2006) 1, 4, 10
8. Brown, M.A., Hua, G., Winder, S.A.J.: Discriminative Learning of Local Image Descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence **33** (2011) 1
9. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: BRIEF: Binary Robust Independent Elementary Features. In: European Conference on Computer Vision (2010) 4
10. Cummins, M.J., Newman, P.: FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. I. J. Robotics Res. **27** (2008) 12
11. DeTone, D., Malisiewicz, T., Rabinovich, A.: Superpoint: Self-Supervised Interest Point Detection and Description. In: CVPR Workshop (2018) 1, 2, 4, 5, 8, 9, 10, 11
12. Dong, J., Soatto, S.: Domain-Size Pooling in Local Descriptors: DSP-SIFT. In: Conference on Computer Vision and Pattern Recognition (2014) 1
13. Dusmanu, M., Rocco, I., Pajdla, T., Pollefeys, M., Sivic, J., Torii, A., Sattler, T.: D2-Net: A Trainable CNN for Joint Description and Detection of Local Features. In: Conference on Computer Vision and Pattern Recognition (2019) 1, 2, 4, 6, 7, 8, 9, 10, 11, 12, 13
14. Fischler, M.A., Bolles, R.C.: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Commun. ACM **24** (1981) 11
15. Gauglitz, S., Höllerer, T., Turk, M.: Evaluation of Interest Point Detectors and Feature Descriptors for Visual Tracking. International Journal of Computer Vision **94** (2011) 3
16. Germain, H., Bourmaud, G., Lepetit, V.: Sparse-To-Dense Hypercolumn Matching for Long-Term Visual Localization. In: International Conference on 3D Vision (2019) 2, 4, 5, 7, 8, 9, 10, 12, 13
17. Hariharan, B., Arbeláez, P.A., Girshick, R.B., Malik, J.: Hypercolumns for Object Segmentation and Fine-Grained Localization. In: Conference on Computer Vision and Pattern Recognition (2014) 6
18. Harris, C., Stephens, M.: A Combined Corner and Edge Detector. In: Proc. of Fourth Alvey Vision Conference (1988) 1, 4, 10
19. Heinly, J., Schönberger, J.L., Dunn, E., Frahm, J.M.: Reconstructing the World* in Six Days *(as Captured by the Yahoo 100 Million Image Dataset). In: Conference on Computer Vision and Pattern Recognition (2015) 1

20. Ioffe, S., Szegedy, C.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In: arXiv Preprint (2015) 7
21. Kim, H., Lee, D., Sim, J., Kim, C.: SOWP: Spatially Ordered and Weighted Patch Descriptor for Visual Tracking. In: International Conference on Computer Vision (2015) 1
22. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. CoRR **abs/1412.6980** (2014) 8
23. Lebeda, K., Matas, J.E.S., Chum, O.: Fixing the Locally Optimized RANSAC. In: British Machine Vision Conference (2012) 11
24. Li, Z., Snavely, N.: Megadepth: Learning Single-View Depth Prediction from Internet Photos. In: Conference on Computer Vision and Pattern Recognition (2018) 4, 7
25. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision **60**(2) (2004) 1, 4, 10
26. Maddern, W., Pascoe, G., Linegar, C., Newman, P.: 1 Year, 1000 Km: The Oxford Robotcar Dataset. I. J. Robotics Res. **36** (2017) 10, 14
27. Mikolajczyk, K., Schmid, C.: Scale & Affine Invariant Interest Point Detectors. International Journal of Computer Vision **60** (2004) 1, 4, 8, 9, 11
28. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Gool, L.V.: A Comparison of Affine Region Detectors. International Journal of Computer Vision **65** (2005) 1
29. Mishchuk, A., Mishkin, D., Radenović, F., Matas, J.: Working Hard to Know Your Neighbor's Margins: Local Descriptor Learning Loss. In: Advances in Neural Information Processing Systems (2017) 8, 9
30. Mishkin, D., Radenović, F., Matas, J.: Repeatability is Not Enough: Learning Affine Regions via Discriminability. In: European Conference on Computer Vision (2017) 8, 9, 11
31. Noh, H., Araujo, A., Sim, J., Weyand, T., Han, B.: Large-Scale Image Retrieval with Attentive Deep Local Features. In: International Conference on Computer Vision (2016) 1, 8, 9, 11
32. Ono, Y., Trulls, E., Fua, P., Yi, K.M.: LF-Net: Learning Local Features from Images. In: NeurIPS (2018) 1, 4, 6, 8, 9
33. Perdoch, M., Chum, O., Matas, J.: Efficient Representation of Local Geometry for Large Scale Object Retrieval. In: Conference on Computer Vision and Pattern Recognition (2009) 8, 9, 11
34. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object Retrieval with Large Vocabularies and Fast Spatial Matching. In: Conference on Computer Vision and Pattern Recognition (2007) 11
35. Revaud, J., Weinzaepfel, P., de Souza, C.R., Pion, N., Csurka, G., Cabon, Y., Humenberger, M.: R2D2: Repeatable and Reliable Detector and Descriptor. In: arXiv Preprint (2019) 1, 2, 4, 6, 8, 9, 10, 11, 12, 13
36. Rocco, I., Cimpoi, M., Arandjelović, R., Torii, A., Pajdla, T., Sivic, J.: Neighbourhood Consensus Networks. In: Advances in Neural Information Processing Systems (2018) 4
37. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.R.: ORB: An Efficient Alternative to SIFT or SURF. In: International Conference on Computer Vision (2011) 4
38. Sarlin, P.E., Cadena, C., Siegwart, R., Dymczyk, M.: From Coarse to Fine: Robust Hierarchical Localization at Large Scale. In: Conference on Computer Vision and Pattern Recognition (2019) 12
39. Sattler, T., Havlena, M., Radenovic, F., Schindler, K., Pollefeys, M.: Hyperpoints and Fine Vocabularies for Large-Scale Location Recognition. In: International Conference on Computer Vision (2015) 6

40. Sattler, T., Leibe, B., Kobbelt, L.: Efficient & Effective Prioritized Matching for Large-Scale Image-Based Localization. IEEE Transactions on Pattern Analysis and Machine Intelligence **39** (2017) 12

41. Sattler, T., Maddern, W., Toft, C., Torii, A., Hammarstrand, L., Stenborg, E., Safari, D., Okutomi, M., Pollefeys, M., Sivic, J., Kahl, F., Pajdla, T.: Benchmarking 6DOF Outdoor Visual Localization in Changing Conditions. In: Conference on Computer Vision and Pattern Recognition (2018) 1, 2, 4, 10, 11, 12

42. Sattler, T., Torii, A., Sivic, J., Pollefeys, M., Taira, H., Okutomi, M., Pajdla, T.: Are Large-Scale 3D Models Really Necessary for Accurate Visual Localization? In: Conference on Computer Vision and Pattern Recognition (2017) 1

43. Sattler, T., Weyand, T., Leibe, B., Kobbelt, L.: Image Retrieval for Image-Based Localization Revisited. In: British Machine Vision Conference (2012) 2, 10, 14

44. Savinov, N., Seki, A., Ladicky, L., Sattler, T., Pollefeys, M.: Quad-Networks: Unsupervised Learning to Rank for Interest Point Detection. In: Conference on Computer Vision and Pattern Recognition (2016) 1

45. Schönberger, J.L., Frahm, J.M.: Structure-From-Motion Revisited. In: Conference on Computer Vision and Pattern Recognition (2016) 1, 8

46. Schönberger, J.L., Zheng, E., Pollefeys, M., Frahm, J.M.: Pixelwise View Selection for Unstructured Multi-View Stereo. In: European Conference on Computer Vision (2016) 1, 8

47. Simo-Serra, E., Trulls, E., Ferraz, L., Kokkinos, I., Fua, P., Moreno-Noguer, F.: Discriminative Learning of Deep Convolutional Feature Point Descriptors. In: International Conference on Computer Vision (2015) 1

48. Simonyan, K., Vedaldi, A., Zisserman, A.: Learning Local Feature Descriptors Using Convex Optimisation. IEEE Transactions on Pattern Analysis and Machine Intelligence **36** (2014) 1

49. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. CoRR **abs/1409.1556** (2014) 5, 7

50. Svärm, L., Enqvist, O., Kahl, F., Oskarsson, M.: City-Scale Localization for Cameras with Known Vertical Direction. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(7) (2017) 1, 12

51. Svärm, L., Enqvist, O., Oskarsson, M., Kahl, F.: Accurate Localization and Pose Estimation for Large 3D Models. In: Conference on Computer Vision and Pattern Recognition (2014) 1

52. Sweeney, C., Fragoso, V., Höllerer, T., Turk, M.: Large Scale SfM with the Distributed Camera Model. In: International Conference on 3D Vision (2016) 1

53. Taira, H., Okutomi, M., Sattler, T., Cimpoi, M., Pollefeys, M., Sivic, J., Pajdla, T., Torii, A.: Inloc: Indoor Visual Localization with Dense Matching and View Synthesis. CoRR **abs/1803.10368** (2018) 1, 4, 10, 11, 14

54. Thomee, B., Shamma, D.A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., Li, L.: YFCC100M: The New Data in Multimedia Research. Commun. ACM **59** (2016) 8

55. Tian, Y., Fan, B., Wu, F.: L2-Net: Deep Learning of Discriminative Patch Descriptor in Euclidean Space. In: Conference on Computer Vision and Pattern Recognition (2017) 4

56. Tian, Y., Yu, X., Fan, B., Wu, F., Heijnen, H., Balntas, V.: SOSNet: Second Order Similarity Regularization for Local Descriptor Learning. In: Conference on Computer Vision and Pattern Recognition (2019) 4

57. Toft, C., Stenborg, E., Hammarstrand, L., Brynte, L., Pollefeys, M., Sattler, T., Kahl, F.: Semantic Match Consistency for Long-Term Visual Localization. In: European Conference on Computer Vision (2018) 12

58. Torii, A., Arandjelovic, R., Sivic, J., Okutomi, M., Pajdla, T.: 24/7 Place Recognition by View Synthesis. IEEE Transactions on Pattern Analysis and Machine Intelligence **40** (2015) 12

59. Tuytelaars, T., Mikolajczyk, K.: Local Invariant Feature Detectors: A Survey. Foundations and Trends in Computer Graphics and Vision **3** (2007) 3

60. Wijmans, E., Furukawa, Y.: Exploiting 2D Floorplan for Building-Scale Panorama RGBD Alignment. In: Conference on Computer Vision and Pattern Recognition (2016) 10

61. Yang, H., Shao, L., Zheng, F., Wang, L., Song, Z.: Recent Advances and Trends in Visual Tracking: A Review. Neurocomputing **74** (2011) 1

62. Yi, K.M., Trulls, E., Lepetit, V., Fua, P.: LIFT: Learned Invariant Feature Transform. In: European Conference on Computer Vision (2016) 1, 4