Learning Feature Descriptors using Camera Pose Supervision

Qianqian Wang[1,2], Xiaowei Zhou[3], Bharath Hariharan[1], and Noah Snavely[1,2]

[1]Cornell University [2]Cornell Tech [3]Zhejiang University

Abstract. Recent research on learned visual descriptors has shown promising improvements in correspondence estimation, a key component of many 3D vision tasks. However, existing descriptor learning frameworks typically require ground-truth correspondences between feature points for training, which are challenging to acquire at scale. In this paper we propose a novel *weakly-supervised* framework that can learn feature descriptors *solely* from relative camera poses between images. To do so, we devise both a new loss function that exploits the epipolar constraint given by camera poses, and a new model architecture that makes the whole pipeline differentiable and efficient. Because we no longer need pixel-level ground-truth correspondences, our framework opens up the possibility of training on much larger and more diverse datasets for better and unbiased descriptors. We call the resulting descriptors **CA**mera **P**ose **S**upervised, or **CAPS**, descriptors. Though trained with weak supervision, CAPS descriptors outperform even prior *fully-supervised* descriptors and achieve state-of-the-art performance on a variety of geometric tasks.[1]

Keywords: Local Features · Feature Descriptors · Correspondence · Image Matching · Camera Pose

1 Introduction

Finding local feature correspondence is a fundamental component of many computer vision tasks, such as structure from motion (SfM) [56] and visual localization [54]. Recently, learned feature descriptors [42,58,65] have shown significant improvements over hand-crafted ones [4,26,37] on standard benchmarks. However, other recent work has observed that, when applied to real-world unseen scenarios, learned descriptors do not always generalize well [39,57].

One potential cause of such limited generalization is the insufficiency of high-quality training data in both quantity and diversity [57]. Ideally, one would train descriptors on fully accurate, dense ground-truth correspondence between image pairs. However, it is hard to collect such data for real imagery, and only a few datasets of this form exist [6,10]. As an alternative, many previous methods resort to SfM datasets that provide pseudo ground-truth correspondences given by matched and reconstructed feature points [39,42,48,65], but these correspondences are sparse and potentially biased by the keypoints used in the SfM pipeline.

[1] Project page: `https://qianqianwang68.github.io/CAPS/`

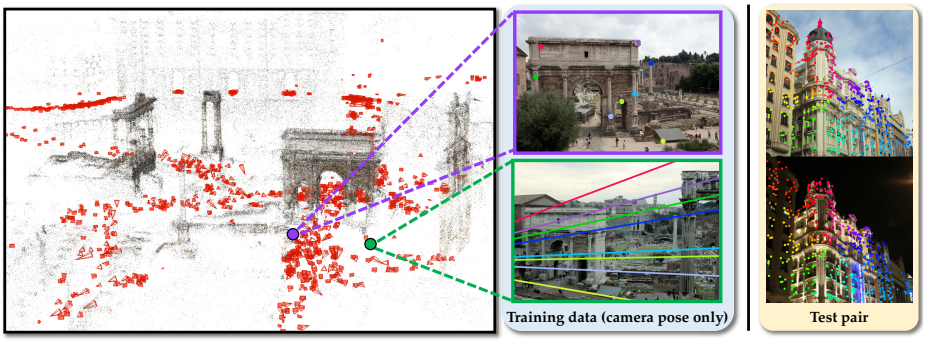


Fig. 1: **Overview of our method**. Our model can learn descriptor using only relative camera poses (e.g., from SfM reconstructions **(a)**). Knowing camera poses, we obtain epipolar constraints illustrated in **(b)**, where points in the first image correspond to the epipolar lines in same color in the second image. We utilize such epipolar constraints as our supervision signal (see Fig. 2). **(c)** shows that at inference, our descriptors establish reliable correspondences even for challenging image pairs.

Another option for obtaining correspondence annotations is synthetic image pairs warped by homographies [13,40]. However, homographies do not capture the full range of geometric and photometric variations observed in real images.

In this paper, we address the challenge of limited training data in descriptor learning by relaxing this requirement of ground-truth pixel-level correspondences. We propose to learn descriptors solely from relative camera poses between pairs of images. Camera poses can be obtained via a variety of non-vision-based sensors, such as IMUs and GPS, and can also be estimated reliably using SfM pipelines [56]. By reducing the supervision requirement to camera poses, it becomes possible to learn better descriptors on much larger and more diverse datasets.

However, existing metric learning based methods for learning descriptors cannot utilize camera poses as supervision, as the triplet or contrastive losses used in such methods cannot be defined with respect to camera poses. Hence, we propose a novel framework to leverage camera pose supervision. Specifically, we translate the relative camera pose between an image pair into an epipolar constraint on pixel locations of matched points as our supervision signal (Fig. 2). The remaining challenge is to make the locations of matched points differentiable with respect to descriptors for training, for which we introduce a new differentiable matching layer (Fig. 3(a)). To further reduce the computation cost and accelerate training, we use a coarse-to-fine matching scheme (Fig. 3(b)) that computes the correspondence at a lower resolution, then locally refines at a finer scale.

Once trained, our system can generate dense feature descriptors for an arbitrary input image, which can then be combined with existing keypoint detectors for downstream tasks. Despite the fact that we only train with *weak* camera pose supervision, our learned descriptors are on par with or even outperform prior *fully-supervised* state-of-the-art methods that train with ground-truth corre-

spondence annotations. Furthermore, while enabling training with solely camera poses, our framework can also be trained with ground-truth correspondences, yielding even better results.

Fig. 1 summarizes our approach. To conclude, our main contributions are:

- We show that camera poses alone suffice to learn good descriptors, which has not been explored in the literature to our knowledge.
- To enable learning from camera poses, we depart from existing metric learning-based approaches and design a novel loss function as well as a new, efficient network architecture.
- We achieve state-of-the-art performance across a range of geometric tasks.

2 Related Work

Descriptor Learning. The dominant paradigm for learning feature descriptors is essentially deep metric learning [8], which encourages matching points to be close whereas non-matching points to be far away in the feature space. Various loss functions (e.g., pairwise and triplet loss [3,8,13,31,66], structured loss [42,47,59,61]) have been developed. Based on the input type, current descriptor learning approaches roughly fall into two categories, *patch-based* and *dense* descriptor methods. Patch-based methods [3,15,21,27,39,42,43,44,48,58,61,65] produce a feature descriptor for each *patch* defined by a keypoint detector, which can be viewed as direct counterparts for hand-crafted feature descriptors [4,5,37,53]. Dense descriptor methods [9,13,14,16,34,50,55] instead use fully-convolutional neural networks [35] to extract dense feature descriptors for the whole image in one forward pass. Our method gives dense descriptors, and unlike the prior work that requires ground-truth correspondence annotations to train, we are able to learn descriptors from the weak supervision of camera pose.

Correspondence Learning. Our differentiable matching layer is related to the correlation layer and cost volume that are widely used to compute stereo correspondences [7,28] or optical flow [17,22,60] in a differentiable manner. However, the search space in these problems is limited to either a single scanline or a local patch, while in wide-baseline matching we must search for matches over the whole image. This necessitates the efficient coarse-to-fine architecture we use. Our method is also related to weakly-supervised semantic correspondence approaches [24,29,46,51,69]. However, they usually assume a simpler parametric transformation between images and tolerate much coarser correspondences than what is required for geometric tasks. Recent work [40,52] explores dense geometric correspondence, but focuses on global optimization of the estimated correspondences rather than the descriptors themselves. In contrast to these prior work, we propose a new architecture that is more suitable for descriptor learning.

Epipolar Constraint. Epipolar constraint has been shown to be useful for learning local features [23,64] and optical flow [67]. MONET [23] proposes the epipolar divergence for learning semantic keypoints, but this loss does not apply to

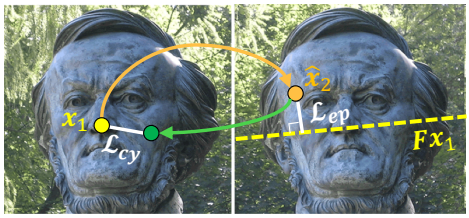


Fig. 2: **Epipolar loss and cycle consistency loss.** $\mathbf{x}_1$ (yellow) is the query point, and $\hat{\mathbf{x}}_2$ (orange) is the predicted correspondence. The epipolar loss $\mathcal{L}_{ep}$ is the distance between $\hat{\mathbf{x}}_2$ and ground-truth epipolar line $\mathbf{F}\mathbf{x}_1$. The cycle consistency loss $\mathcal{L}_{cy}$ is the L_2 distance between $\mathbf{x}_1$ and its forward-backward corresponding point (green).

dense descriptor learning. [64] leverages epipolar constraints to generate pseudo-groundtruth correspondences but this process is non-differentiable. In contrast, we enable differentiable training of dense descriptors using the epipolar constraint.

3 Method

Given only image pairs with camera pose, standard deep metric learning methods do not apply. Therefore, we devise a new method to exploit the geometric information of camera pose for descriptor learning. Specifically, we translate relative camera pose into an epipolar constraint between image pairs, and enforce the predicted matches to obey this constraint (Sec. 3.1). Since this constraint is imposed on pixel coordinates, we must make the coordinates of correspondences differentiable with respect to the feature descriptors. For this we devise a differentiable matching layer (Sec. 3.2). To further improve efficiency, we introduce a coarse-to-fine architecture (Sec. 3.3) to accelerate training, which also boosts the descriptor performance. We elaborate on our method below.

3.1 Loss Formulation

Our training data consists of image pairs with relative camera poses. To train our correspondence system with such data, we propose to use two complimentary loss terms: a novel epipolar loss, and a cycle consistency loss (Fig. 2).

Given the relative pose and camera intrinsics for a pair of images $\mathbf{I}_1$ and $\mathbf{I}_2$, one can compute the fundamental matrix $\mathbf{F}$. The epipolar constraint states that $\mathbf{x}_2^T\mathbf{F}\mathbf{x}_1 = 0$ holds if $\mathbf{x}_1$ and $\mathbf{x}_2$ is a true match, where $\mathbf{F}\mathbf{x}_1$ can be interpreted as the epipolar line corresponding to $\mathbf{x}_1$ in $\mathbf{I}_2$.[2] We treat $\mathbf{x}_1$ as the query point and re-fashion this constraint into an epipolar loss based on the distance between the predicted correspondence location and the ground-truth epipolar line:

$$\mathcal{L}_{ep}(\mathbf{x}_1) = \text{dist}(h_{1\to 2}(\mathbf{x}_1), \mathbf{F}\mathbf{x}_1), \tag{1}$$

[2] For simplicity, we use the same symbols for homogeneous and Cartesian coordinates.

where $h_{1\to2}(\mathbf{x}_1)$ is the predicted correspondence in $\mathbf{I}_2$ for the point $\mathbf{x}_1$ in $\mathbf{I}_1$, and $\text{dist}(\cdot,\cdot)$ is the distance between a point and a line.

The epipolar loss alone only encourages a predicted match to lie on the epipolar line, rather than near the ground-truth correspondence location (which is at an unknown location on the line). To provide additional supervision, we additionally introduce a cycle consistency loss. This loss encourages the forward-backward mapping of a point to be spatially close to itself [63]:

$$\mathcal{L}_{cy}(\mathbf{x}_1) = ||h_{2\to1}(h_{1\to2}(\mathbf{x}_1)) - \mathbf{x}_1||_2. \tag{2}$$

This term encourages the network to find true correspondences and suppress other outputs, especially those that satisfy the epipolar constraint alone.

Full Training Objective. For each image pair, our total objective is a weighted sum of epipolar and cycle consistency losses, totaled over n sampled query points:

$$\mathcal{L}(\mathbf{I}_1, \mathbf{I}_2) = \sum_{i=1}^{n} [\mathcal{L}_{ep}(\mathbf{x}_1^i) + \lambda \mathcal{L}_{cy}(\mathbf{x}_1^i)], \tag{3}$$

where $\mathbf{x}_1^i$ is the i-th training point in $\mathbf{I}_1$, and λ is a weight for the cycle consistency loss term. At the end of Sec. 3.2, we further show how we can reweight individual training instances in Eq. (3) to improve training.

3.2 Differentiable Matching Layer

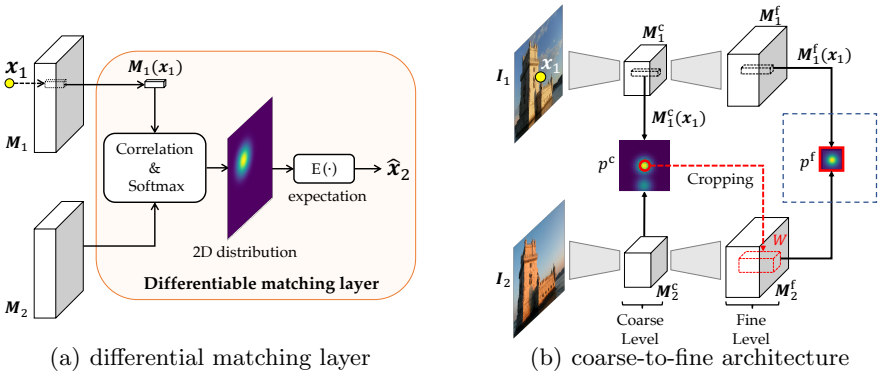
The objective defined above is a simple function of the pixel locations of the predicted correspondences. Minimizing this objective through gradient descent therefore requires these locations to be differentiable with respect to the network parameters. Many prior methods establish correspondence by identifying nearest neighbor matches, which unfortunately is a non-differentiable operation.

To address this challenge, we propose a differentiable matching layer, illustrated in Fig. 3(a). Given a pair of images, we first use convolutional networks with shared weights to extract dense feature descriptors $\mathbf{M}_1$ and $\mathbf{M}_2$. To compute the correspondence for a query point $\mathbf{x}_1$ in $\mathbf{I}_1$, we correlate the feature descriptor at $\mathbf{x}_1$, denoted by $\mathbf{M}_1(\mathbf{x}_1)$, with all of $\mathbf{M}_2$. Following a 2D softmax operation [19], we obtain a distribution over 2D pixel locations of $\mathbf{I}_2$, indicating the probability of each location being the correspondence of $\mathbf{x}_1$. We denote this probability distribution as $p(\mathbf{x}|\mathbf{x}_1, \mathbf{M}_1, \mathbf{M}_2)$:

$$p(\mathbf{x}|\mathbf{x}_1, \mathbf{M}_1, \mathbf{M}_2) = \frac{\exp\left(\mathbf{M}_1(\mathbf{x}_1)^{\mathrm{T}} \mathbf{M}_2(\mathbf{x})\right)}{\sum_{\mathbf{y}\in\mathbf{I}_2} \exp\left(\mathbf{M}_1(\mathbf{x}_1)^{\mathrm{T}} \mathbf{M}_2(\mathbf{y})\right)}, \tag{4}$$

where $\mathbf{y}$ varies over the pixel grid of $\mathbf{I}_2$. A single 2D match can then be computed as the expectation of this distribution:

$$\hat{\mathbf{x}}_2 = h_{1\to2}(\mathbf{x}_1) = \sum_{\mathbf{x}\in\mathbf{I}_2} \mathbf{x} \cdot p(\mathbf{x}|\mathbf{x}_1, \mathbf{M}_1, \mathbf{M}_2). \tag{5}$$



(a) differential matching layer (b) coarse-to-fine architecture

Fig. 3: **Network architecture design. (a) differentiable matching layer**. For a query point, its correspondence location is represented as the expectation of a distribution computed from the correlation between feature descriptors. **(b) The coarse-to-fine module**. We use the location of highest probability at coarse level (red circle) to determine the location of a local window W at the fine level. During training, we compute the correspondence locations at both coarse and fine level from distribution p^c and p^f, respectively, and impose our loss functions on both. This allows us to train both coarse- and fine-level features simultaneously.

This makes the entire system end-to-end trainable. Since the correspondence location is computed from the correlation between feature descriptors, enforcing it to be correct would facilitate descriptor learning.

Leveraging Uncertainty during Training. This differentiable matching also provides an interpretable measure of *uncertainty*. For each query point $\mathbf{x}_1$, we can calculate the total variance $\sigma^2(\mathbf{x}_1)$ as an uncertainty measure, which is defined as the trace of the covariance matrix of the 2D distribution $p(\mathbf{x}|\mathbf{x}_1, \mathbf{M}_1, \mathbf{M}_2)$. High variance indicates multiple or diffuse modes, signifying an unreliable prediction.

This uncertainty can help identify unreliable correspondences and improve training. In particular, due to the lack of ground-truth correspondence annotations, it is unknown if a query point has a true correspondence in the other image during training (which could be missing due to occlusion or truncation). Minimizing the loss for such points can lead to incorrect training signals. To alleviate this issue, we reweight the losses for each individual point using the total variance defined above, resulting in the final weighted loss function:

$$\mathcal{L}(\mathbf{I}_1, \mathbf{I}_2) = \sum_{i=1}^{n} \frac{1}{\sigma(\mathbf{x}_1^i)} [\mathcal{L}_{ep}(\mathbf{x}_1^i) + \lambda \mathcal{L}_{cy}(\mathbf{x}_1^i)], \tag{6}$$

where the weight $1/\sigma(\mathbf{x}_1^i)$ are normalized so that they sum up to one. This weighting strategy weakens the effect of infeasible and non-discriminative training points, which we find to be critical for rapid convergence. Prior work [25,46] on semantic correspondence leverages the uncertainty in a similar way, but their

uncertainty is predicted using extra network parameters whereas ours is directly derived from the learned descriptors.

3.3 Coarse-to-Fine Architecture

During training, we impose supervision only on sparsely sampled query points for each pair of images. While the computational cost is made manageable in this way, having to search correspondence over the entire image space is still costly. To overcome this issue, we propose a coarse-to-fine architecture that significantly improves computational efficiency, while preserving the resolution of learned descriptors. Fig. 3(b) illustrates the coarse-to-fine module. Instead of generating a flat feature descriptor map, we produce both coarse-level feature descriptors $\mathbf{M}_1^{\mathrm{c}}, \mathbf{M}_2^{\mathrm{c}}$ and fine-level feature descriptors $\mathbf{M}_1^{\mathrm{f}}, \mathbf{M}_2^{\mathrm{f}}$.

Coarse-to-fine matching works as follows. Given a query point $\mathbf{x}_1$, we first compute the distribution $p^{\mathrm{c}}(\mathbf{x}|\mathbf{x}_1, \mathbf{M}_1^{\mathrm{c}}, \mathbf{M}_2^{\mathrm{c}})$ over *all locations* of the coarse feature map. At the fine level, on the contrary, we compute the fine-level distribution only in a *local window* W centered at the highest probability location in the coarse-level distribution (with coordinates rescaled appropriately). Given coarse- and fine-level distributions, correspondences at both levels can be computed. We then impose our loss function (Eq. (6)) on correspondences at both levels, which allows us to train both coarse and fine features descriptors simultaneously.

This architecture allows us to learn high-resolution descriptors without evaluating full correlation between large feature maps, significantly reducing computational cost. In addition, as observed by Liu et al. [33], we find that coarse-to-fine reasoning not only improves efficiency but also boosts matching accuracy (Sec. 4.3). By concatenating both coarse- and fine-level descriptors, we obtain the final hierarchical descriptors [16] that capture both abstract and detailed information.

3.4 Discussion

Effectiveness of Epipolar Constraint. The seemingly weak epipolar constraint actually provides empirically sufficient supervision for descriptor learning, as suggested by results in Sec. 4. One key reason is that the epipolar constraint suppresses a large number of incorrect correspondence—i.e., every point not on the epipolar line. Moreover, among all valid predictions that satisfy the epipolar constraint, true correspondences are most likely to have similar feature encodings given their local appearance similarity. Therefore, by aggregating such a geometric constraint over all training data, the network learns to encode the similarity between true correspondences, leading to effective learned descriptors.

Training with Ground-truth Correspondence Annotations. Although the focus of this paper is on learning from camera poses alone, our system can also be trained with ground-truth correspondence annotations when such data is available. In this case, we can replace our loss functions with an L_2 distance between the pixel locations of the predicted and ground-truth correspondence. As shown in Fig. 7, our method trained with groundtruth correspondences achieves

even better performance than our method trained with camera poses, with both outperforming prior fully supervised methods.

Matching at Test Time. The descriptors learned by our system can be integrated in standard feature matching pipelines. Given a detected keypoint, feature vectors in the coarse and fine feature maps are extracted by interpolation and concatenated to form the final descriptor. We then match features using the standard Euclidean distance between them.

3.5 Implementation Details

Architecture. We use a ImageNet-pretrained ResNet-50 [11,20,49] architecture, truncated after `layer3`, as our backbone. With an additional convolutional layer we obtain the coarse-level feature map. The fine-level feature map is obtained by further convolutional layers along with up-sampling and skip-connections. The sizes of the coarse- and fine-level feature map are 1/16 and 1/4 of the original image size, respectively. They both have a feature dimensionality of 128. The size of the local window W at fine level is $1/8\times$ the size of the fine-level feature map.

Training Data. We train using the MegaDepth dataset [32], which consists of 196 different scenes reconstructed from over 1M internet photos using COLMAP [56]. 130 out of 196 scenes are used for training and the rest are for validation and testing. This gives us millions of training pairs with known camera poses. We train our system on these pairs using only the provided camera poses and intrinsics.

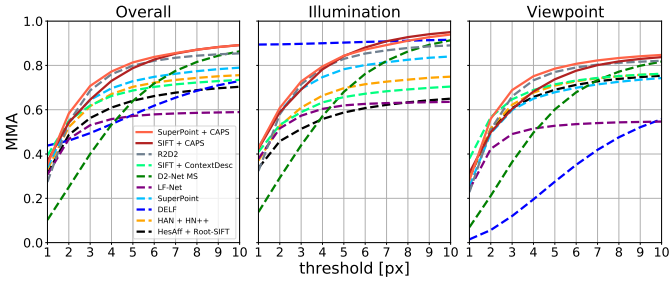
Training Details. We train the network using Adam [30] with a base learning rate of 10^{-4}. The weight λ for the cycle consistency term is set to 0.1. $n = 500$ query points are used in each training image pair due to memory constraints. These query points consist of 90% SIFT [37] keypoints and 10% random points.

For more implementation details, please refer to the supplementary material.

4 Experimental Results

To evaluate our descriptors, referred to as CAPS, we conduct three sets of experiments:

1. **Feature matching experiments:** The most direct evaluation of CAPS is in terms of how accurately they can be matched between images. We evaluate both sparse and dense feature matching on the HPatches dataset [2].
2. **Experiments on downstream tasks:** Feature matches are rarely the end-goal. Instead, they form a core part of many 3D reconstruction tasks. We evaluate the impact of CAPS on downstream tasks (homography estimation on HPatches as well as relative pose estimation on MegaDepth [32] and ScanNet [10]) and 3D reconstruction (as part of an SfM pipeline in the ETH local feature benchmark [57]).
3. **Ablation study:** We evaluate the impact of each proposed contribution using the HPatches dataset.



Methods	# Feature	# Match
HesAff + Root-SIFT	6.7K	2.8K
HAN + HN++	3.9K	2.0K
LF-Net	0.5K	0.2K
SuperPoint	1.7K	0.9K
DELF	4.6K	1.9K
SIFT + ContextDesc	4.4K	1.9K
D2-Net MS	8.3K	2.8K
R2D2	5.0K	1.8K
SuperPoint + CAPS	1.7K	0.9K
SIFT + CAPS	4.4K	1.5K

Fig. 4: **Mean matching accuracy (MMA) on HPatches [2]**. For each method, we show the MMA with varying pixel error thresholds. We also report the mean number of detected features and mutual nearest neighbor matches. With SuperPoint [13] keypoints, our approach achieves the best overall performance after 2px.

4.1 Feature Matching Results

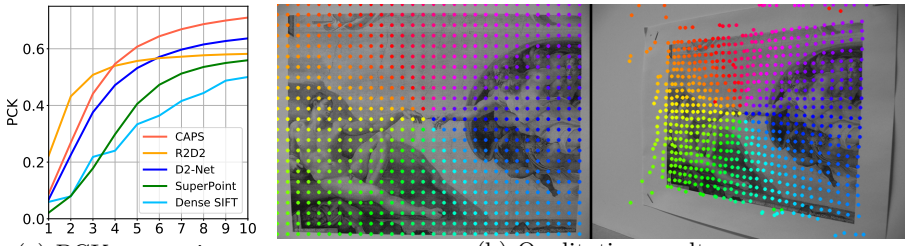
We evaluate our descriptors on both sparse and dense feature matching on the HPatches dataset [2]. HPatches is a homography dataset containing 116 sequences, where 57 sequences have illumination changes and 59 have viewpoint changes.

Sparse Feature Matching. Given a pair of images, we extract keypoints in both images and match them using feature descriptors. We follow the same evaluation protocol as in D2-Net [14] and use the mean matching accuracy (MMA) as the evaluation metric. The MMA score is defined as the average percentage of correct matches per image pair under a certain pixel error threshold. Only mutual nearest neighbor matches are considered.

We combine CAPS with SIFT [37] and SuperPoint [13] keypoints which are representative of hand-crafted and learned keypoints, respectively. We compare to several baselines: Hessian affine detector [41] with RootSIFT descriptor [37,1] (HesAff + RootSIFT), HesAffNet [43] regions with HardNet++ descriptors [42] (HAN + HN++), DELF [45], SuperPoint [13], LF-Net [48], multi-scale D2-Net [14] (D2-Net MS), SIFT detector with ContextDesc descriptors [38] (SIFT + ContextDesc), as well as R2D2 [50].

Fig. 4 shows MMA results on the HPatches dataset. We report results for the whole dataset, as well as for subsets corresponding to illumination and viewpoint changes. Following D2-Net [14], we additionally present the mean number of detected features per image and mutual nearest neighbor matches per pair. With SuperPoint keypoints CAPS achieves the best overall performance, and with SIFT keypoints CAPS also achieves competitive performance. In addition, with the same detectors, CAPS shows clear improvements over its counterparts ("SIFT + CAPS" vs. "SIFT + ContextDesc", "SuperPoint + CAPS" vs. "SuperPoint").

Dense Feature Matching. To evaluate our dense matching capability, we extract keypoints on image grids in the first image and find their nearest neighbor match in the full second image. The percentage of correct keypoints (PCK)



(a) PCK comparison (b) Qualitative result

Fig. 5: **Dense feature matching on HPatches. (a)** PCK comparison. CAPS outperforms other methods at larger pixel thresholds ($>$ 4px). **(b)** Qualitative result of dense feature matching. Color indicates correspondence.

metric [8,36,68] is used to measure performance: the predicted match for a query point is deemed correct if it is within a certain pixel threshold of the true match.

We compare to baseline methods that produce dense descriptors: Dense SIFT [37], SuperPoint [13], D2-Net [14] and R2D2 [50]. Fig. 5(a) shows the mean PCK over all image pairs on HPatches. CAPS achieves the overall best performance and is only worse than R2D2 [50] at small thresholds ($\leq$ 4px). This is because the R2D2 we use here computes descriptor maps at the full input image resolution, whereas ours are 4x downsampled. Fig. 5(b) shows the qualitative performance of our dense correspondence.

4.2 Results on Downstream Tasks

Next, we evaluate how well CAPS facilitates downstream tasks. We focus on two tasks related to two-view geometry estimation: homography estimation and relative camera pose estimation, and a third task related to 3D reconstruction.

Homography Estimation. We use the same HPatches dataset as in Sec. 4.1 for the homography estimation task. We follow the corner correctness metric used in SuperPoint [12,13]. The four corners of one image are transformed to the other image using the estimated homography and compared with the corners computed using the groundtruth homography. The estimated homography is deemed correct if the average error of the four corners is less than ϵ pixels.

Following SuperPoint [13], we extract a maximum of 1,000 keypoints from each image, and robustly estimate the homography from mutual nearest neighbor matches. The comparison of homography accuracy between CAPS and other methods is shown in Tab. 1. As can be seen, CAPS improves over both SIFT and SuperPoint descriptors. With SuperPoint keypoints, CAPS achieves the overall best performance even without training on annotated correspondences.

Relative Pose Estimation. We also evaluate the performance of CAPS on the task of relative camera pose estimation. Note that we train only on MegaDepth [32] but test on both MegaDepth and ScanNet [10], an indoor dataset that we use

Table 1: **Homography estimation accuracy [%] at 1, 3, 5 pixels on HPatches.** CAPS with SuperPoint keypoints achieves the overall best performance.

Methods	$\epsilon = 1$	$\epsilon = 3$	$\epsilon = 5$
SIFT [37]	40.5	68.1	77.6
LF-Net [48]	34.8	62.9	73.8
SuperPoint [13]	37.4	73.1	82.8
D2-Net [14]	16.7	61.0	75.9
ContextDesc [38]	41.0	73.1	82.2
R2D2 [50]	40.0	**75.0**	84.7
CAPS w/ SIFT kp.	34.6	72.2	81.7
CAPS w/ SuperPoint kp.	**44.8**	74.5	**85.7**

Table 2: **Relative pose estimation accuracy on ScanNet [10] and MegaDepth [32].** Each cell shows the accuracy of estimated rotations and translations (as *rotation accuracy / translation accuracy*). Each value shown is the percentage of pairs with relative pose error under a certain threshold (5° for ScanNet and 10° for MegaDepth). **Higher is better.** d_{frame} represents the interval between frames. Larger frame intervals imply harder pairs for matching.

Methods	**Accuracy on ScanNet [%]**			**Accuracy on MegaDepth [%]**		
	$d_{frame} = 10$	$d_{frame} = 30$	$d_{frame} = 60$	*easy*	*moderate*	*hard*
SIFT [37]	91.0 / 14.1	65.1 / 15.6	41.4 / 11.9	58.9 / 20.2	26.9 / 11.8	13.6 / 9.6
SIFT w/ ratio test [37]	91.2 / 15.9	67.1 / 19.8	44.3 / 15.9	63.9 / 25.6	36.5 / 17.0	20.8 / 13.2
SuperPoint [13]	94.4 / 17.5	75.9 / 26.3	53.4 / 22.1	67.2 / 27.1	38.7 / 18.8	24.5 / 14.1
HardNet [42]	95.8 / 18.2	79.0 / 24.7	55.6 / 21.8	66.3 / 26.7	39.3 / 18.8	22.5 / 12.3
LF-Net [48]	93.6 / 17.4	76.0 / 22.4	49.9 / 18.0	52.3 / 18.6	25.5 / 13.2	15.4 / 11.1
D2-Net [14]	91.6 / 13.3	68.4 / 19.5	42.0 / 14.6	61.8 / 23.6	35.2 / 19.2	19.1 / 12.2
ContextDesc [38]	91.5 / 16.3	73.8 / 21.8	51.4 / 18.5	68.9 / 27.1	43.1 / 21.5	27.5 / 14.1
R2D2 [50]	**97.4** / **22.3**	**86.1** / **31.7**	**62.9** / **28.8**	69.4 / 30.3	48.3 / 23.9	32.6 / 17.4
CAPS w/ SIFT kp.	92.3 / 16.3	74.8 / 22.5	50.8 / 20.9	70.0 / 30.5	50.2 / 24.8	36.8 / 16.1
CAPS w/ SuperPoint kp.	96.1 / 17.1	79.5 / 27.2	59.3 / 26.1	**72.9** / **30.5**	**53.5** / **27.9**	**38.1** / **19.2**

to test the generalization of CAPS. For MegaDepth, we generate overlapping image pairs from test scenes, and sort them into three subsets according to relative rotation angle: *easy* ($[0°, 15°]$), *moderate* ($[15°, 30°]$) and *hard* ($[30°, 60°]$). For ScanNet, we follow LF-Net [48] and randomly sample image pairs at three different frame intervals, 10, 30, and 60. Each subset in MegaDepth and ScanNet consists of 1,000 image pairs.

To estimate relative pose, we first estimate the essential matrix from mutual nearest neighbor matches (RANSAC [18] is applied), and then decompose it to get the relative pose. For SIFT [37] we additionally prune matches using the ratio test [37], since that is the common practice for camera pose estimation (i.e, we report results of both plain SIFT and SIFT with a carefully-tuned ratio test).

Following UCN [8], we evaluate the estimated camera pose using angular deviation for both rotation and translation. We consider a rotation or translation to be correct if the angular deviation is less than a threshold, and report the

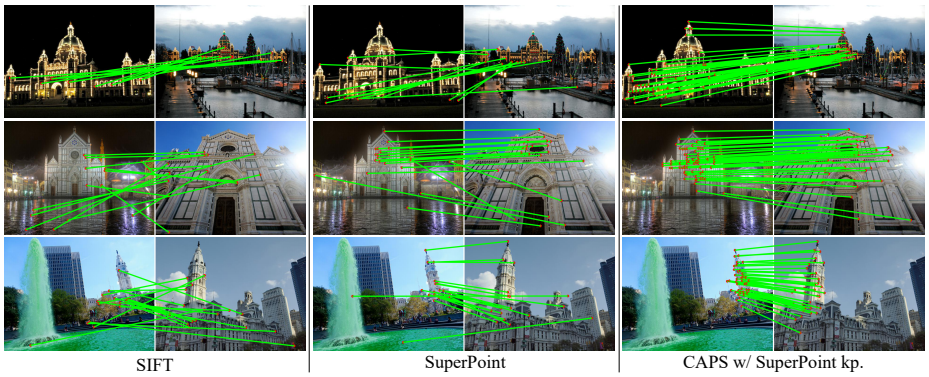


Fig. 6: **Sparse feature matching results after RANSAC**. The test image pairs are from MegaDepth [32]. Green lines indicate correspondences. Our method works well even under challenging illumination and viewpoint changes.

average accuracy for that threshold. We set a threshold of 5° for ScanNet and 10° for MegaDepth, as MegaDepth is harder due to larger illumination changes. Results for all methods are reported in Tab. 2. CAPS improves performance over SIFT and SuperPoint descriptors, and "CAPS w/ SuperPoint keypoints" outperforms all other methods but is outperformed by R2D2 [50] on ScanNet. Qualitative results on MegaDepth test images are shown in Fig. 6.

3D Reconstruction. Finally, we evaluate the effectiveness of CAPS descriptors in the context of 3D reconstruction using the ETH local features benchmark [57]. We extract CAPS descriptors at keypoint locations provided by [57] and feed them into the protocol. Following [39], we do not conduct the ratio test, in order to investigate the direct matching performance of the descriptors. To quantify the quality of SfM, we report the number of registered images (# Registered), sparse 3D points (#Sparse Points) and image observations (# Obs), the mean track lengths (Track Len.), and the mean reprojection error (Reproj. Err.).

We use SIFT [37], GeoDesc [39], D2-Net [14] and SOSNet [62] as baselines and show the results in Tab. 3. CAPS is comparable to or even outperforms our baselines in terms of the completeness of the sparse reconstruction (i.e., the number of registered images, sparse points and observations). However, we do not achieve the lowest reprojection error. A similar situation is observed in [39,62], which can be explained by the trade-off between completeness of reconstruction and low reprojection error: fewer matches tend to lead to lower reprojection error. Taking all metrics into consideration, the performance of CAPS for SfM is competitive, indicating the advantages of CAPS even trained with only weak pose supervision.

4.3 Ablation Analysis

In this section, we conduct ablation analysis to demonstrate the effectiveness of our proposed camera pose supervision and architectural designs. We follow the

Table 3: **Evaluation on the ETH local features benchmark [57].** Note that SIFT and D2-Net [14] apply ratio test but other methods do not. Overall, CAPS performs on par with state-of-the-art local features on this task.

		#Registered	#Sparse points	#Obs.	Track Len.	Reproj. Err.
Madrid Metropolis 1,344 images	SIFT [37]	500	116K	734K	6.32	**0.61**px
	GeoDesc [39]	809	307K	1,200K	3.91	0.66px
	D2-Net [14]	501	84K	-	**6.33**	1.28px
	SOSNet [62]	844	**335K**	1,411K	4.21	0.70px
	CAPS	**851**	242K	**1,489K**	6.16	1.03px
Gendarmen-markt 1,463 images	SIFT	1,035	339K	1,872K	**5.52**	**0.70**px
	GeoDesc	1,208	780K	2,903K	3.72	0.74px
	D2-Net	1,053	250K	-	5.08	1.19px
	SOSNet	**1,201**	**816K**	3,255K	3.98	0.77px
	CAPS	1,179	627K	**3,330K**	5.31	1.00px
Tower of London 1,576 images	SIFT	804	240K	1,863K	**7.77**	**0.62**px
	GeoDesc	1,081	**622K**	**2,852K**	4.58	0.69px
	D2-Net	785	180K	-	5.32	1.24px
	CAPS	**1,104**	452K	2,627K	5.81	0.98px

evaluation protocol in Sec. 4.1 and report MMA and PCK score over all image pairs in the HPatches dataset [2]. For sparse feature matching, we combine our descriptors with SIFT [37] keypoints. The variants of our default method (*Ours*) are introduced below. For fair comparison, we train each variant on the same training data (∼20K image pairs) from MegaDepth [32] for 10 epochs.

Variants. *Ours from scratch* is trained from scratch instead of using ImageNet [11] pretrained weights. *Ours supervised* is trained on sparse ground-truth correspondences provided by the SfM models of MegaDepth [32]. We simply change the epipolar loss to a L_2 loss between predicted and groundtruth correspondence locations. *Triplet Loss* is also trained on sparse ground-truth correspondences, but using a standard triplet loss and a hard negative mining strategy [8]. *Ours w/o c2f* is a single-scale version of our method, where the coarse-level feature maps are removed and only the fine-level feature maps are trained and used as descriptors. *Ours w/o cycle* does not use the cycle consistency loss term ($\lambda = 0$), and *Ours w/o reweighting* does not use the uncertainty re-weighting strategy, but uses uniform weights during training. Below we provide a detailed analysis based on these variants. The results are shown in Fig. 7.

Analysis of Supervision Signal. Both *Ours supervised* and *Ours* outperform the plain version of *Triplet Loss*, where *Ours supervised* and *Triplet Loss* share the same correspondence annotations but *Ours* uses only camera pose. *Ours supervised* outperforms *Triplet Loss* because of the geometric distance-based losses (as opposed to metric learning) and the coarse-to-fine architecture. Compared to *Ours supervised*, the gains of *Ours* decrease a bit, but our epipolar loss still leverages the rich information in the epipolar constraint and allows us to outperform *Triplet Loss* and other past fully supervised work in Sec. 4. In terms of loss functions, cycle consistency only provides marginal improvement, and training with only cycle consistency loss fails. This validates the importance of

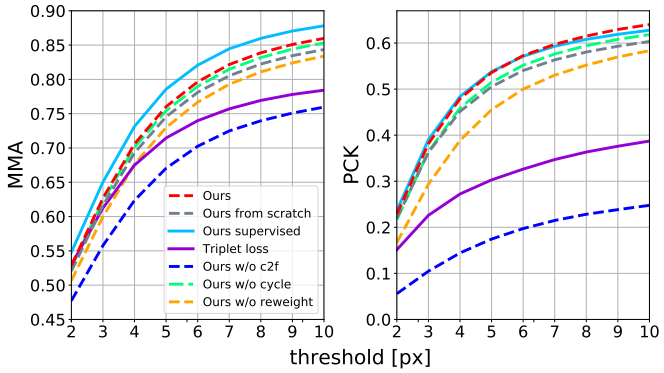


Fig. 7: **Ablation study on HPatches**. Solid lines indicate methods trained with ground-truth correspondence; dashed lines indicate ones trained with only camera pose.

epipolar constraint. *Ours from scratch* shows that even with randomly initialized weights, our network still succeeds to converge and learn descriptors, further validating the effectiveness of our loss functions.

Analysis of Architecture Design. As shown in Fig. 7, the coarse-to-fine module significantly improves performance. Two explanations for this improvement include: 1) At the fine level, correspondence is computed within a local window, which may reduce issues arising from multi-modal distributions compared to a flat model that computes expectations over the whole image; and 2) The coarse-to-fine module produces hierarchical feature descriptors that capture both global and local information, which may be beneficial for feature matching.

5 Conclusion

In this paper, we propose a novel descriptor learning framework that can be trained using only camera pose supervision. We present both new loss functions that exploit the epipolar constraints, and a new efficient architectural design that enables learning by making the correspondence differentiable. Experiments showed that our method achieves state-of-the-art performance across a range of geometric tasks, outperforming fully supervised counterparts without using any correspondence annotations for training. In future work, we will study how to further improve invariance of the learned descriptors to large geometric transformations. It is also worth investigating if the pose supervision and traditional metric learning losses are complementary to each other, and if their combination can lead to even better performance.

Acknowledgements. We thank Kai Zhang, Zixin Luo, Zhengqi Li for helpful discussion and comments. This work was partly supported by a DARPA LwLL grant, and in part by the generosity of Eric and Wendy Schmidt by recommendation of the Schmidt Futures program.

References

1. Arandjelović, R., Zisserman, A.: Three things everyone should know to improve object retrieval. In: CVPR (2012)
2. Balntas, V., Lenc, K., Vedaldi, A., Mikolajczyk, K.: Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In: CVPR (2017)
3. Balntas, V., Riba, E., Ponsa, D., Mikolajczyk, K.: Learning local feature descriptors with triplets and shallow convolutional neural networks. In: Proc. British Machine Vision Conf. (BMVC). p. 3 (2016)
4. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: Proc. European Conf. on Computer Vision (ECCV) (2006)
5. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: Brief: Binary robust independent elementary features. In: ECCV (2010)
6. Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3d: Learning from rgb-d data in indoor environments. arXiv preprint arXiv:1709.06158 (2017)
7. Chang, J.R., Chen, Y.S.: Pyramid stereo matching network. In: CVPR (2018)
8. Choy, C.B., Gwak, J., Savarese, S., Chandraker, M.: Universal correspondence network. In: NeurIPS (2016)
9. Christiansen, P.H., Kragh, M.F., Brodskiy, Y., Karstoft, H.: Unsuperpoint: End-to-end unsupervised interest point detector and descriptor. arXiv preprint arXiv:1907.04011 (2019)
10. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: CVPR (2017)
11. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009)
12. DeTone, D., Malisiewicz, T., Rabinovich, A.: Deep image homography estimation. arXiv preprint arXiv:1606.03798 (2016)
13. DeTone, D., Malisiewicz, T., Rabinovich, A.: Superpoint: Self-supervised interest point detection and description. In: CVPR Workshops (2018)
14. Dusmanu, M., Rocco, I., Pajdla, T., Pollefeys, M., Sivic, J., Torii, A., Sattler, T.: D2-net: A trainable cnn for joint detection and description of local features. arXiv preprint arXiv:1905.03561 (2019)
15. Ebel, P., Mishchuk, A., Yi, K.M., Fua, P., Trulls, E.: Beyond cartesian representations for local descriptors. In: ICCV (2019)
16. Fathy, M.E., Tran, Q.H., Zeeshan Zia, M., Vernaza, P., Chandraker, M.: Hierarchical metric learning and matching for 2d and 3d geometric correspondences. In: ECCV (2018)
17. Fischer, P., Dosovitskiy, A., Ilg, E., Häusser, P., Hazırbaş, C., Golkov, V., Van der Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. arXiv preprint arXiv:1504.06852 (2015)
18. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24**(6), 381–395 (1981)
19. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016), `http://www.deeplearningbook.org`
20. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
21. He, K., Lu, Y., Sclaroff, S.: Local descriptors optimized for average precision. In: CVPR (2018)

22. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: Flownet 2.0: Evolution of optical flow estimation with deep networks. In: CVPR (2017)
23. Jafarian, Y., Yao, Y., Park, H.S.: Monet: Multiview semi-supervised keypoint via epipolar divergence. arXiv preprint arXiv:1806.00104 (2018)
24. Jeon, S., Kim, S., Min, D., Sohn, K.: Parn: Pyramidal affine regression networks for dense semantic correspondence. In: ECCV (2018)
25. Jeon, S., Min, D., Kim, S., Sohn, K.: Joint learning of semantic alignment and object landmark detection. In: ICCV (2019)
26. Ke, Y., Sukthankar, R.: Pca-sift: A more distinctive representation for local image descriptors. In: CVPR (2004)
27. Keller, M., Chen, Z., Maffra, F., Schmuck, P., Chli, M.: Learning deep descriptors with scale-aware triplet networks. In: CVPR (2018)
28. Kendall, A., Martirosyan, H., Dasgupta, S., Henry, P., Kennedy, R., Bachrach, A., Bry, A.: End-to-end learning of geometry and context for deep stereo regression. In: ICCV (2017)
29. Kim, S., Lin, S., JEON, S.R., Min, D., Sohn, K.: Recurrent transformer networks for semantic correspondence. In: NeurIPS (2018)
30. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
31. Kumar, B., Carneiro, G., Reid, I., et al.: Learning local image descriptors with deep siamese and triplet convolutional networks by minimising global loss functions. In: CVPR (2016)
32. Li, Z., Snavely, N.: Megadepth: Learning single-view depth prediction from internet photos. In: CVPR (2018)
33. Liu, C., Yuen, J., Torralba, A., Sivic, J., Freeman, W.T.: Sift flow: Dense correspondence across different scenes. In: ECCV (2008)
34. Liu, Y., Shen, Z., Lin, Z., Peng, S., Bao, H., Zhou, X.: Gift: Learning transformation-invariant dense visual descriptors via group cnns. In: NeurIPS (2019)
35. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR (2015)
36. Long, J.L., Zhang, N., Darrell, T.: Do convnets learn correspondence? In: NeurIPS (2014)
37. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV **60**(2), 91–110 (2004)
38. Luo, Z., Shen, T., Zhou, L., Zhang, J., Yao, Y., Li, S., Fang, T., Quan, L.: Contextdesc: Local descriptor augmentation with cross-modality context. In: CVPR (2019)
39. Luo, Z., Shen, T., Zhou, L., Zhu, S., Zhang, R., Yao, Y., Fang, T., Quan, L.: Geodesc: Learning local descriptors by integrating geometry constraints. In: ECCV (2018)
40. Melekhov, I., Tiulpin, A., Sattler, T., Pollefeys, M., Rahtu, E., Kannala, J.: Dgc-net: Dense geometric correspondence network. In: WACV (2019)
41. Mikolajczyk, K., Schmid, C.: Scale & affine invariant interest point detectors. IJCV **60**(1), 63–86 (2004)
42. Mishchuk, A., Mishkin, D., Radenovic, F., Matas, J.: Working hard to know your neighbor's margins: Local descriptor learning loss. In: NeurIPS (2017)
43. Mishkin, D., Radenovic, F., Matas, J.: Repeatability is not enough: Learning affine regions via discriminability. In: ECCV (2018)
44. Mukundan, A., Tolias, G., Chum, O.: Explicit spatial encoding for deep local descriptors. In: CVPR (2019)

45. Noh, H., Araujo, A., Sim, J., Weyand, T., Han, B.: Large-scale image retrieval with attentive deep local features. In: ICCV (2017)
46. Novotny, D., Albanie, S., Larlus, D., Vedaldi, A.: Self-supervised learning of geometrically stable features through probabilistic introspection. In: CVPR (2018)
47. Oh Song, H., Xiang, Y., Jegelka, S., Savarese, S.: Deep metric learning via lifted structured feature embedding. In: CVPR (2016)
48. Ono, Y., Trulls, E., Fua, P., Yi, K.M.: Lf-net: learning local features from images. In: NeurIPS (2018)
49. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in PyTorch. In: NIPS Autodiff Workshop (2017)
50. Revaud, J., Weinzaepfel, P., de Souza, C.R., Humenberger, M.: R2D2: repeatable and reliable detector and descriptor. In: NeurIPS (2019)
51. Rocco, I., Arandjelovic, R., Sivic, J.: Convolutional neural network architecture for geometric matching. In: CVPR (2017)
52. Rocco, I., Cimpoi, M., Arandjelović, R., Torii, A., Pajdla, T., Sivic, J.: Neighbourhood consensus networks. In: NeurIPS (2018)
53. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.R.: Orb: An efficient alternative to sift or surf. In: Proc. Int. Conf. on Computer Vision (ICCV). Citeseer (2011)
54. Sattler, T., Maddern, W., Toft, C., Torii, A., Hammarstrand, L., Stenborg, E., Safari, D., Okutomi, M., Pollefeys, M., Sivic, J., et al.: Benchmarking 6dof outdoor visual localization in changing conditions. In: CVPR (2018)
55. Schmidt, T., Newcombe, R., Fox, D.: Self-supervised visual descriptor learning for dense correspondence. IEEE Robotics and Automation Letters **2**(2), 420–427 (2016)
56. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: CVPR (2016)
57. Schönberger, J.L., Hardmeier, H., Sattler, T., Pollefeys, M.: Comparative Evaluation of Hand-Crafted and Learned Local Features. In: CVPR (2017)
58. Simo-Serra, E., Trulls, E., Ferraz, L., Kokkinos, I., Fua, P., Moreno-Noguer, F.: Discriminative learning of deep convolutional feature point descriptors. In: ICCV (2015)
59. Sohn, K.: Improved deep metric learning with multi-class n-pair loss objective. In: NeurIPS (2016)
60. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In: CVPR (2018)
61. Tian, Y., Fan, B., Wu, F.: L2-net: Deep learning of discriminative patch descriptor in euclidean space. In: CVPR (2017)
62. Tian, Y., Yu, X., Fan, B., Wu, F., Heijnen, H., Balntas, V.: Sosnet: Second order similarity regularization for local descriptor learning. In: CVPR (2019)
63. Wang, X., Jabri, A., Efros, A.A.: Learning correspondence from the cycle-consistency of time. In: CVPR (2019)
64. Yang, G., Malisiewicz, T., Belongie, S., Farhan, E., Ha, S., Lin, Y., Huang, X., Yan, H., Xu, W.: Learning data-adaptive interest points through epipolar adaptation. In: CVPR Workshops (2019)
65. Yi, K.M., Trulls, E., Lepetit, V., Fua, P.: Lift: Learned invariant feature transform. In: ECCV (2016)
66. Zhang, L., Rusinkiewicz, S.: Learning local descriptors with a cdf-based dynamic soft margin. In: ICCV (2019)
67. Zhong, Y., Ji, P., Wang, J., Dai, Y., Li, H.: Unsupervised deep epipolar flow for stationary or dynamic scenes. In: CVPR (2019)

68. Zhou, T., Jae Lee, Y., Yu, S.X., Efros, A.A.: Flowweb: Joint image set alignment by weaving consistent, pixel-wise correspondences. In: CVPR (2015)
69. Zhou, T., Krahenbuhl, P., Aubry, M., Huang, Q., Efros, A.A.: Learning dense correspondence via 3d-guided cycle consistency. In: CVPR (2016)