Efficient Neighbourhood Consensus Networks via Submanifold Sparse Convolutions

Ignacio Rocco[1(✉)], Relja Arandjelović[2], and Josef Sivic[1,3]

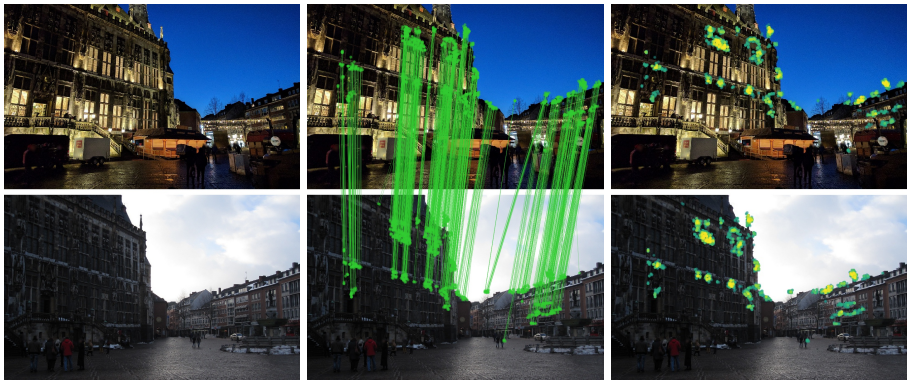
[1] WILLOW, Inria, DI-ENS, CNRS, PSL Research University, Paris, France
ignacio.rocco@inria.fr
[2] DeepMind, London, United Kingdom
[3] Czech Institute of Informatics, Robotics and Cybernetics, CTU, Prague, Czechia

Abstract. In this work we target the problem of estimating accurately localised correspondences between a pair of images. We adopt the recent Neighbourhood Consensus Networks that have demonstrated promising performance for difficult correspondence problems and propose modifications to overcome their main limitations: large memory consumption, large inference time and poorly localised correspondences. Our proposed modifications can reduce the memory footprint and execution time more than 10×, with equivalent results. This is achieved by *sparsifying* the correlation tensor containing tentative matches, and its subsequent processing with a 4D CNN using submanifold sparse convolutions. Localisation accuracy is significantly improved by processing the input images in higher resolution, which is possible due to the reduced memory footprint, and by a novel two-stage correspondence relocalisation module. The proposed Sparse-NCNet method obtains state-of-the-art results on the HPatches Sequences and InLoc visual localisation benchmarks, and competitive results on the Aachen Day-Night benchmark.

Keywords: Image matching, neighbourhood consensus, sparse CNN.

1 Introduction

Finding correspondences between images depicting the same 3D scene is one of the fundamental tasks in computer vision [24, 29, 35] with applications in 3D reconstruction [50, 51, 57], visual localisation [15, 47, 53] or pose estimation [14, 18, 40]. The predominant approach currently consists of first *detecting* salient local features, by selecting the local extrema of some form of feature selection function, and then *describing* them by some form of feature descriptor [7, 28, 45]. While hand-crafted features such as Hessian affine detectors [30] with SIFT descriptors [28] achieve impressive performance under strong viewpoint changes and constant illumination [31], their robustness to illumination changes is limited [31, 63]. More recently, a variety of trainable keypoint detectors [26, 27, 33, 56] and descriptors [5, 6, 22, 32, 54, 59] have been proposed, with the purpose of obtaining increased robustness over hand-crafted methods. While this approach has achieved some success, extreme illumination changes such as day-to-night



(a) Input images (b) Output matches (c) Match confidence

Fig. 1: **Correspondence estimation with Sparse-NCNet.** Given an input image pair (a), we show the *raw* output correspondences produced by Sparse-NCNet (b) which contain groups of spatially coherent matches. These groups tend to form around highly-confident matches, which are shown in yellow shades (c).

matching combined with changes in camera viewpoint remain a challenging open problem [4, 13, 15]. In particular, all local feature methods, whether hand-crafted or trained, suffer from missing detections under these extreme appearance changes.

In order to overcome this issue, the detection stage can be avoided and, instead, features can be extracted on a dense grid across the image. This approach has been successfully used for both place recognition [1, 15, 36, 55] and image matching [44, 47, 57]. However, extracting features densely comes with additional challenges: it is memory intensive and the localisation accuracy of the features is limited by the sampling interval of the grid used for the extraction.

In this work we adopt the dense feature extraction approach. In particular, we build on the recent Neighbourhood Consensus Networks (NCNet) [44], that allow for jointly trainable feature extraction, matching, and match-filtering to directly output a strong set of (mostly) correct correspondences. Our proposed approach, Sparse-NCNet, seeks to overcome the limitations of the original NCNet formulation, namely: large memory consumption, high execution time and poorly localised correspondences.

Our contributions are the following. First, we propose the efficient Sparse-NCNet model, which is based on a 4D convolutional neural network operating on a *sparse* correlation tensor, which is obtained by storing only the most promising correspondences, instead of the set of all possible correspondences. Sparse-NCNet processes this sparse correlation tensor with submanifold sparse convolutions [21] and can obtain equivalent results to NCNet while being several times faster (up to 10×) and requiring much less memory (up to 20×) without decrease in performance compared to the original NCNet model. Second, we propose a two-stage relocalisation module to improve the localisation accuracy of the correspondences

output by Sparse-NCNet. Finally, we show that the proposed model significantly outperforms state-of-the-art results on the HPatches Sequences [3] benchmark for image matching with challenging viewpoint and illumination changes and the InLoc [53] benchmark for indoor localisation and camera pose estimation. Furthermore, we show our model obtains competitive results on the Aachen Day-Night benchmark [47], which evaluates day-night feature matching for the task of camera localisation. An example of the correspondences produced by our method is presented in Fig. 1. **Our code and models are available online [43].**

2 Related work

In this section, we review the relevant related work.

Matching with trainable local features. Most recent work in trainable local features has focused on learning more robust keypoint *descriptors* [5, 6, 22, 32, 54, 59]. Initially these descriptors were used in conjunction with classic hand-crafted keypoint detectors, such as DoG [28]. Recently, trainable keypoint *detectors* where also proposed [26, 27, 33, 56], as well as methods providing both *detection and description* [12, 13, 37, 41, 58]. From these, some adopt the classic approach of first performing detection on the whole image and then computing descriptors from local image patches, cropped around the detected keypoints [37, 58], while the most recent methods compute a joint representation from which both detections and descriptors are computed [12, 13, 41]. In most cases, local features obtained by these methods are independently matched using nearest-neighbour search with the Euclidean distance [5, 6, 32, 54], although some works have proposed to learn the distance function as well [22, 59]. As discussed in the previous section, local features are prone to loss of detections under extreme lighting changes [15]. In order to alleviate this issue, in this work we adopt the usage of densely extracted features, which are described next.

Matching with densely extracted features. Motivated by applications in large-scale visual search, others have found that using densely extracted features provides additional robustness to illumination changes compared to local features extracted at detected keypoints, which suffer from low repeatability under strong illumination changes [55, 62]. This approach was also adopted by later work [1, 36]. Such densely extracted features used for image retrieval are typically computed on a coarse low resolution grid (*e.g.* 40×30). However, such coarse localisation of the dense features is not an issue for visual retrieval, as the dense features are not directly matched, but rather aggregated into a single image-level descriptor, which is used for retrieval. Recently, densely extracted features have been also employed directly for 3D computer vision tasks, such as 3D reconstruction [57], indoor localisation and camera pose estimation [53], and outdoor localisation with night queries [15, 47]. In these methods, correspondences are obtained by nearest-neighbour search performed on extracted descriptors, and filtered by the mutual nearest-neighbour criterion [38]. In this work, we build on the NCNet

method [44], where the match filtering function is learnt from data. Recent methods for learning to filter matches are discussed next.

Learning to filter incorrect matches. When using both local features extracted at keypoints or densely extracted features, the obtained matches by nearest-neighbour search contain a certain portion of incorrect matches. In the case of local features, a heuristic approach such as Lowe's ratio test [28] can be used to filter these matches. However the ratio threshold value needs to be manually tuned for each method. To avoid this issue, filtering by mutual nearest neighbours can be used instead [13]. Recently, trainable approaches have also been proposed for the task of filtering local feature correspondences [9, 34, 46, 60]. Yi *et al*. [34] propose a neural-network architecture that operates on 4D match coordinates and classifies each correspondence as either correct or incorrect. Brachmann *et al*. [9] propose the Neural-guided RANSAC, which extends the previous method to produce weights instead of classification labels, which are used to guide RANSAC sampling. Zhang *et al*. [60] also extend the work of Yi *et al*. in their proposed Order-Aware Networks, which capture local context by clustering 4D correspondences onto a set of ordered clusters, and global context by processing these clusters with a multi-layer perceptron. Finally, Sarlin *et al*. [46] describe a graph neural network followed by an optimisation procedure to estimate correspondences between two set of local features. These methods were specifically designed for filtering local features extracted at keypoint locations and not features extracted on a dense grid. Furthermore, these methods are focused only on learning match filtering, and are decoupled from the problem of learning how to detect and describe the local features.

In this paper we build on the NCNet method [44] for filtering incorrect matches, which was designed for dense features. Furthermore, contrary to the above described methods, our approach performs feature extraction, matching and match filtering in a single pipeline.

Improved feature localisation. Recent methods for local feature detection and description which use a joint representation [12, 13] as well as methods for dense feature extraction [44, 57] suffer from poor feature localisation, as the features are extracted on a low-resolution grid. Different approaches have been proposed to deal with this issue. The D2-Net method [13] follows the approach used in SIFT [28] for refining the keypoint positions, which consists of locally fitting a quadratic function to the feature detection function around the feature position and solving for the extrema. The Superpoint method [12] uses a CNN decoder that produces a one-hot output for each 8×8 pixel cell of the input image (in case a keypoint is effectively detected in this region), therefore achieving pixel-level accuracy. Others [57] use the intermediate higher resolution features from the CNN to improve the feature localisation, by assigning to each pooled feature the position of the feature with highest L2 norm from the preceding higher resolution map (and which participated in the pooling). This process can be repeated up to the input image resolution.

The relocalisation approach of NCNet [44] is based on a max-argmax operation on the 4D correlation tensor of exhaustive feature matches. This approach can only increase the resolution of the output matches by a factor of 2. In contrast, we describe a new two-stage relocalisation module that builds on the approach used in NCNet, by combining a hard relocalisation stage that has similar effects to NCNet's max-argmax operation, with a soft-relocalisation stage that obtains sub-feature-grid accuracy via interpolation.

Sparse Convolutional Neural Networks were recently used for the purpose of processing sparse 2D data, such as handwritten characters [20]; 3D data, such as 3D point-clouds [19]; or even 4D data, such as temporal sequences of 3D point clouds [10]. These models have shown great success in 3D point-cloud processing tasks such as semantic segmentation [10, 21] and point-cloud registration [11, 17]. In this work, we use networks with *submanifold sparse convolutions* [21] for the task of filtering correspondences between images, which can be represented as a sparse set of points in a 4D space of image coordinates. In submanifold sparse convolutions, the active sites remain constant between the input and output of each convolutional layer. As a result, the sparsity level remains fixed and does not change after each convolution operation. To the best of our knowledge this is the first time these models are applied to the task of match filtering.

3 Sparse Neighbourhood Consensus Networks

In this section we detail the proposed Sparse Neighbourhood Consensus Networks. We start with a brief review of Neighbourhood Consensus Networks [44] identifying their main limitations. Next, we describe our approach which overcomes these limitations.

3.1 Review: Neighbourhood Consensus Networks

The Neighbourhood Consensus Network [44] is a method for feature extraction, matching and match filtering. Contrary to most methods, which operate on local features, NCNet operates on dense feature maps $(f^A, f^B) \in \mathbb{R}^{h \times w \times c}$ with c channels, which are extracted over a regular grid of $h \times w$ spatial resolution. These are obtained from the input image pair $(I_A, I_B) \in \mathbb{R}^{H \times W \times 3}$ by a fully convolutional feature extraction network. The resolution $h \times w$ of the extracted dense features is typically 1/8 or 1/16 of the input image resolution $H \times W$, depending on the particular feature extraction network architecture used.

Next, the exhaustive set of all possible matches between the dense feature maps f^A and f^B is computed and stored in a 4D correlation tensor $c^{AB} \in \mathbb{R}^{h \times w \times h \times w}$. Finally, the correspondences in c^{AB} are filtered by a 4D CNN. This network can detect coherent spatial matching patterns and propagate information from the most certain matches to their neighbours, robustly identifying the correct correspondences. This last filtering step is inspired by the neighbourhood consensus procedure [8, 48, 49, 52, 61], where a particular match is verified by

analysing the existence of other coherent matches in its spatial neighbourhood in both images.

Despite its promising results, the original formulation of Neighbourhood Consensus Networks has three main drawbacks that limit its practical application: it is (i) memory intensive, (ii) slow, and (iii) matches are poorly localised. These points are discussed in detail next.

High memory requirements. The high memory requirements are due to the computation of the correlation tensor $c^{AB} \in \mathbb{R}^{h \times w \times h \times w}$ which stores all matches between the densely extracted image features $(f^A, f^B) \in \mathbb{R}^{h \times w \times c}$. Note that the number of elements in the correlation tensor $(h \times w \times h \times w)$ grows quadratically with respect to the number of features $(h \times w)$ of the dense feature maps (f^A, f^B), therefore limiting the ability to increase the feature resolution. For instance, for dense feature maps of resolution 200×150, the correlation tensor would require by itself 3.4GB of GPU memory in the standard 32-bit float precision. Furthermore, processing this correlation tensor using the subsequent 4D CNN would require more than 50GB of GPU memory, which is much more than what is currently available on most standard GPUs. While 16-bit half-float precision could be used to halve these memory requirements, they would still be prohibitively large.

Long processing time. In addition, Neighbourhood Consensus Networks are slow as the full dense correlation tensor must be processed. For instance, processing the $100 \times 75 \times 100 \times 75$ correlation tensor containing matches between a pair of dense feature maps of 100×75 resolution takes approximately 10 seconds on a standard Tesla T4 GPU.

Poor match localisation. Finally, the high-memory requirements limit the maximum feature map resolution that can be processed, which in turn limits the localisation accuracy of the estimated correspondences. For instance, for a pair images with 1600×1200px resolution, where correspondences are computed using a dense feature map with a resolution of 100×75, the output correspondences are localised within an error of 8 pixels. This can be problematic if correspondences are used for tasks such as pose estimation, where small errors in the localisation of correspondences in image-space can yield high camera pose errors in 3D space.

In this paper, we devise strategies to overcome the limitations of the original NCNet method, while keeping its main advantages, such as the usage of dense feature maps which avoids the issue of missing detections, and the processing of multiple matching hypotheses to avoid early matching errors. Our efficient Sparse-NCNet approach is described next.

3.2 Sparse-NCNet: Efficient Neighbourhood Consensus Networks

In this section, we describe the Sparse-NCNet approach in detail. An overview is presented in Fig. 2. Similar to NCNet, the first stage of our proposed method consists in dense feature extraction. Given a pair of RGB input images $(I^A, I^B) \in$

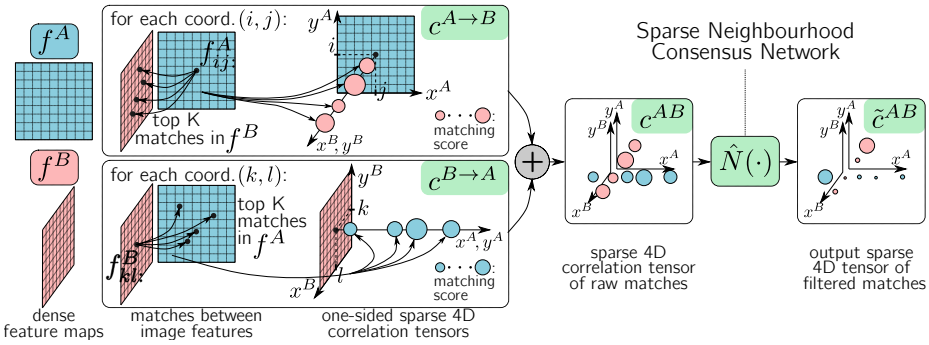


Fig. 2: **Overview of Sparse-NCNet.** From the dense feature maps f^A and f^B, their top K matches are computed and stored in the one-sided sparse 4D correlation tensors $c^{A\to B}$ and $c^{B\to A}$, which are later combined to obtain the symmetric sparse correlation tensor c^{AB}. The raw matching score values in c^{AB} are processed by the 4D Sparse-NCNet $\hat{N}(\cdot)$ producing the output tensor $\tilde{c}^{AB}$ of filtered matching scores.

$\mathbb{R}^{H\times W\times 3}$, $L2$-normalized dense features $(f^A, f^B) \in \mathbb{R}^{h\times w\times c}$ are extracted via a fully convolutional network $F(\cdot)$:

$$f^A = F(I^A), f^B = F(I^B). \tag{1}$$

Then, these dense features are matched and stored into a *sparse correlation tensor*. Contrary to the original NCNet formulation, where *all* the pairwise matches between the dense features are stored and processed, we propose to keep *only the top K matches* for a given feature, measured by the cosine similarity. In detail, each feature $f^A_{ij:}$ from image A at position (i,j) is matched with its K *nearest-neighbours* in f^B, and vice versa. The one-sided sparse correlation tensor, matching from image A to image B $(A \to B)$ is then described as:

$$c^{A\to B}_{ijkl} = \begin{cases} \langle f^A_{ij:}, f^B_{kl:}\rangle & \text{if } f^B_{kl:} \text{ within K-NN of } f^A_{ij:} \\ 0 & \text{otherwise} \end{cases}. \tag{2}$$

To make the sparse correlation map invariant to the ordering of the input images, we also perform this in the reverse direction $(B \to A)$, and add the two one-sided correlation tensors together to obtain the final (symmetric) *sparse correlation tensor*:

$$c^{AB} = c^{A\to B} + c^{B\to A}. \tag{3}$$

This tensor uses a sparse representation, where only non-zero elements need to be stored. Note that the number of stored elements is, at most, $h \times w \times K \times 2$ which is in practice much less than the $h \times w \times h \times w$ elements of the dense correlation tensor, obtaining great memory savings in both the storage of this tensor and its subsequent processing. For example, for a feature map of size 100×75 and $K = 10$, the sparse representation takes 3.43MB vs. 215MB of the dense representation, resulting in a 12× reduction of the processing time. In the

case of feature maps with 200×150 resolution, the sparse representation takes 13.7MB vs. 3433MB for the dense representation. This allows Sparse-NCNet to also process feature maps at this resolution, something that was not possible with NCNet due to the high memory requirements. The proposed *sparse correlation tensor* is a compromise between the common procedure of taking the best scoring match and the approach taken by NCNet, where all pairwise matches are stored. In this way, we can keep sufficient information in order avoid early mistakes, while keeping low memory consumption and processing time.

Then the sparse correlation tensor is processed by a permutation-invariant CNN ($\hat{N}(\cdot)$), to produce the output filtered correlation map $\tilde{c}^{AB}$:

$$\tilde{c}^{AB} = \hat{N}(c^{AB}). \tag{4}$$

The permutation invariant CNN $\hat{N}(\cdot)$ consists of applying the 4D CNN $N(\cdot)$ twice such that the same output matches are obtained regardless of the order of the input images:

$$\hat{N}(c^{AB}) = N(c^{AB}) + \big(N\big((c^{AB})^T\big)\big)^T, \tag{5}$$

where by transposition we mean exchanging the first two dimensions with the last two dimensions, which correspond to the coordinates of the two input images. The 4D CNN $N(\cdot)$ operates on the 4D space of correspondences, and is trained to perform the neighbourhood consensus filtering. Note that while $N(\cdot)$ is a sparse CNN using submanifold sparse convolutions [21], where the active sites between the sparse input and output remain constant, the convolution kernel filters are dense (*i.e.* hypercubic).

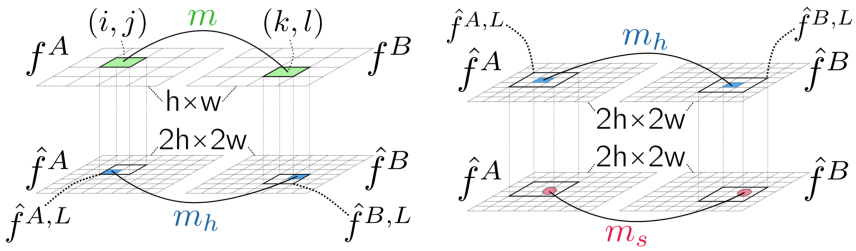
While in the original NCNet method, a soft mutual nearest-neighbour operation $M(\cdot)$ is also performed, we have removed it as we noticed its effect was not significant when operating on the sparse correlation tensor. From the output correlation tensor $\tilde{c}^{AB}$, the output matches are computed by applying argmax at each coordinate:

$$\big((i,j),(k,l)\big) \text{ a match if } \begin{cases} (i,j) = \underset{(a,b)}{\operatorname{argmax}}\ \tilde{c}^{AB}_{abkl}, \text{ or} \\ (k,l) = \underset{(c,d)}{\operatorname{argmax}}\ \tilde{c}^{AB}_{ijcd} \end{cases}, \tag{6}$$

where (i,j) is the match coordinate in the sampling grid of f^A, and (k,l) is the match coordinate in the sampling grid of f^B.

3.3 Match relocalisation by guided search

While the sparsification of the correlation tensor presented in the previous section allows processing higher resolution feature maps, these are still several times smaller in resolution than the input images. Hence, they are not suitable for applications that require (sub)pixel feature localisation such as camera pose estimation or 3D-reconstruction.



(a) Hard relocalisation (b) Soft relocalisation

Fig. 3: **Two-stage relocalisation module.** (a) The hard relocalisation step allows to increase by 2× the localisation accuracy of the matches m outputted by Sparse-NCNet, which are defined on the $h \times w$ feature maps f^A and f^B. This is done by keeping the most similar match m_h between two 2×2 local features $\hat{f}^{A,L}$ and $\hat{f}^{B,L}$, cropped from the $2h \times 2w$ feature maps $\hat{f}^A$ and $\hat{f}^B$. (b) The soft relocalisation step then refines the position of these matches in the $2h \times 2w$ grid, by computing sub-feature-grid soft localisation displacements based on the softargmax operation.

To address this issue, in this paper we propose a two-stage relocalisation module based on the idea of guided search. The intuition is that we search for accurately localised matches on $2h \times 2w$ resolution dense feature maps, guided by the coarse matches output by Sparse-NCNet at $h \times w$ resolution. For this, dense features are first extracted at twice the normal resolution $(\hat{f}^A, \hat{f}^B) \in \mathbb{R}^{2h \times 2w \times c}$, which is done by upsampling the input image by 2× before feeding it into the feature extraction CNN $F(\cdot)$. Note that these higher resolution features are used for relocalisation only, *i.e.* they are not used to compute the correlation tensor or processed by the 4D CNN for match-filtering, which would be too expensive. Then, these dense features are downsampled back to the normal $h \times w$ resolution by applying a 2×2 max-pooling operation with a stride of 2, obtaining f^A and f^B. These low resolution features $(f^A, f^B) \in \mathbb{R}^{h \times w \times c}$ are processed by Sparse-NCNet, which outputs matches in the form $m = \big((i,j),(k,l)\big)$, with the coordinates (i,j) and (k,l) indicating the position of the match in f^A and f^B, respectively, as described by (6).

Having obtained the output matches in $h \times w$ resolution, the first step (hard relocalisation) consists in finding the best equivalent match in the $2h \times 2w$ resolution grid. This is done by analysing the matches between two local crops of the high resolution features $\hat{f}^A$ and $\hat{f}^B$, and keeping the highest-scoring one. The second step (soft relocalisation) then refines this correspondence further, by obtaining a sub-feature accuracy in the $2h \times 2w$ grid. These two relocalisation steps are illustrated in Fig. 3, and are now described in detail.

Hard relocalisation. The first step is hard relocalisation, which can improve localisation accuracy by 2×. For each match $m = \big((i,j),(k,l)\big)$, the 2× upsampled coordinates $\big((2i,2j),(2k,2l)\big)$ are first computed, and 2×2 local feature crops $\hat{f}^{A,L}, \hat{f}^{B,L} \in \mathbb{R}^{2 \times 2 \times c}$ are sampled around these coordinates from the high

resolution feature maps $\hat{f}^A$ and $\hat{f}^B$:

$$\hat{f}^{A,L} = (\hat{f}^A_{ab:})_{\substack{2i \leq a \leq 2i+1, \\ 2j \leq b \leq 2j+1}}, \tag{7}$$

and similarly for $\hat{f}^{B,L}$. This is done using a ROI-pooling operation [16]. Finally, exhaustive matches between the local feature crops $\hat{f}^{A,L}$ and $\hat{f}^{B,L}$ are computed, and the output of the hard relocalisation module is the displacement associated with the maximal matching score:

$$\Delta m_h = \big((\delta i, \delta j), (\delta k, \delta l)\big) = \underset{(a,b),(c,d)}{\text{argmax}} \langle \hat{f}^{A,L}_{ab:}, \hat{f}^{B,L}_{cd:} \rangle. \tag{8}$$

Then, the final match location from the hard relocalisation stage is computed as:

$$m_h = 2m + \Delta m_h = \big((2i + \delta i, 2j + \delta j), (2k + \delta k, 2l + \delta l)\big). \tag{9}$$

Note that the relocalised matches m_h are defined in a $2h \times 2w$ grid, therefore obtaining a $2\times$ increase in localisation accuracy with respect to the initial matches m, which are defined in a $h \times w$ grid. Also note that while the implementation is different, the effect of the proposed hard relocalisation is similar to the max-argmax operation used in NCNet [44], while being more memory efficient as it avoids the computation of the a dense correlation tensor in high resolution.

Soft relocalisation. The second step consists of a soft relocalisation operation that obtains sub-feature localisation accuracy in the $2h \times 2w$ grid of high resolution features $\hat{f}^A$ and $\hat{f}^B$. For this, new 3×3 local feature crops $(\hat{f}^{A,L}, \hat{f}^{B,L}) \in \mathbb{R}^{3\times3\times c}$ are sampled around the coordinates of the estimated matches m_h from the previous relocalisation stage. Note that no upsampling of the coordinates is done in this case, as the matches are already in the $2h \times 2w$ range. Then, soft relocalisation displacements are computed by performing the softargmax operation [58] on the matching scores between the central feature of $\hat{f}^{A,L}$ and the whole of $\hat{f}^{B,L}$, and vice versa:

$$\Delta m_s = \big((\delta i, \delta j), (\delta k, \delta l)\big) \text{ where } \begin{cases} (\delta i, \delta j) = \underset{(a,b)}{\text{softargmax}} \langle \hat{f}^{A,L}_{ab:}, \hat{f}^{B,L}_{11:} \rangle \\ (\delta k, \delta l) = \underset{(c,d)}{\text{softargmax}} \langle \hat{f}^{A,L}_{11:}, \hat{f}^{B,L}_{cd:} \rangle \end{cases} \tag{10}$$

The intuition of the softargmax operation is that it computes a weighted average of the candidate positions in the crop where the weights are given by the softmax of the matching scores. The final matches from soft relocalisation are obtained by applying the soft displacements to the matches from hard relocalisation: $m_s = m_h + \Delta m_s$.

4 Experimental evaluation

We evaluate the proposed Sparse-NCNet method on three different benchmarks: (i) HPatches Sequences, which evaluates the matching task directly, (ii) InLoc, which

targets the problem of indoor 6-dof camera localisation and (iii) Aachen Day-Night, which targets the problem of outdoor 6-dof camera localisation with challenging day-night illumination changes. We first present the implementation details followed by the results on these three benchmarks. Additional 3D reconstruction results are in the extended version of this work [42].

Implementation details. We train the Sparse-NCNet model following the training protocol from [44]. We use the IVD dataset with the weakly-supervised mean matching score loss for training [44]. The 4D CNN $N(\cdot)$ has two sparse convolution layers with 3^4 sized kernels, with 16 output channels in the hidden layer. A value of $K = 10$ is used for computing c^{AB} (3). The model is implemented using PyTorch [39], MinkowskiEngine [10] and Faiss [23], and trained for 5 epochs using Adam [25] with a learning rate of 5×10^{-4}. A pretrained ResNet-101 (up to `conv_4_23`) with no strided convolutions in the last block is used as the feature extractor $F(\cdot)$. This feature extraction model is not finetuned as the training dataset is small (3861 image pairs) and that would lead to overfitting and loss of generalisation. The softargmax operation in (10) uses a temperature value of 10. In the following experiments, all correspondences are first obtained according to (6), and then only the top-scored correspondences according to the value of $\tilde{c}^{AB}$ are kept (typically between 500-2000).

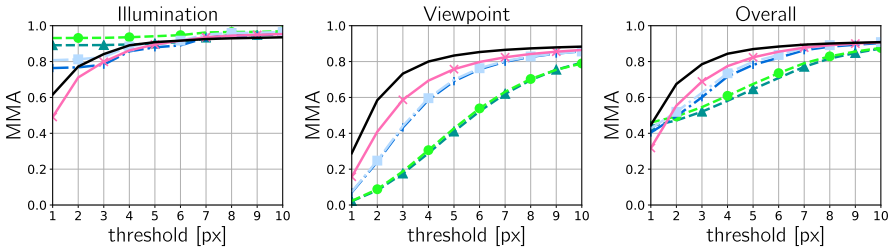
4.1 HPatches Sequences

The HPatches Sequences [3] benchmark assesses the matching accuracy under strong *viewpoint* and *illumination* variations. We follow the evaluation procedure from [13], where 108 image sequences are employed, each from a different planar scene, and each containing 6 images. The first image from each sequence is matched against the remaining 5 images. The benchmark employs 56 sequences with viewpoint changes, and constant illumination conditions, and 52 sequences with illumination changes and constant viewpoint. The metric used for evaluation is the mean matching accuracy (MMA) [13]. Further details about this metric are provided in the extended version [42].

Ablations. In Fig. 4 we present ablations and a comparison with NCNet. The benefits of sparsification are shown by comparing Sparse-NCNet and NCNet under equal conditions, both without relocalisation (methods A1 vs. A2), and with hard relocalisation only (methods B1 vs. B2). The results in Fig. 4 show that Sparse-NCNet can obtain significant reductions in processing time and memory consumption, while keeping almost the same matching performance. Furthermore, we show that Sparse-NCNet+hard-relocalisation (B1) produces superior results to Sparse-NCNet alone (A1). Finally, we show that using the two-stage relocalisation (C1) produces higher matching accuracy than only using hard relocalisation (B1), with minimal impact on runtime or memory requirements. We have also experimented with replacing our relocalisation module with the one from DenseSfM [57]. This resulted in a drop of 11% of the MMA@5px on HPatches, from 87% to 76%, showing the superiority of our approach.

	Method	Feature resolution	Reloc. method	Reloc. resolution	Mean time (s)	Peak VRAM (MB)
A1.	Sparse-NCNet	100 × 75	—	—	0.83	251
A2.	NCNet	100 × 75	—	—	9.81	5763
B1.	Sparse-NCNet	100 × 75	H	200 × 150	1.55	1164
B2.	NCNet	100 × 75	H	200 × 150	10.56	7580
C1.	Sparse-NCNet	100 × 75	H+S	200 × 150	1.56	1164
C2.	Sparse-NCNet	200 × 150	H+S	400 × 300	7.51	2391

(a) Time and GPU memory comparison (Tesla T4 GPU)



(b) MMA on HPatches Sequences

Fig. 4: **Ablations and comparison with NCNet**. Sparse-NCNet can obtain equivalent results to NCNet, both without relocalisation (*c.f.* A1 vs. A2), and with hard relocalisation (H) (*c.f.* B1 vs. B2), while greatly reducing execution time and memory consumption. The proposed two-stage relocalisation (H+S) brings an improvement in matching accuracy with a minor increase in execution time (*c.f.* C1 vs. B1). Finally, the reduced memory consumption in Sparse-NCNet allows for processing in higher resolution, which produces the best results, while still being faster and more memory efficient than NCNet (*c.f.* C2 vs. B2).

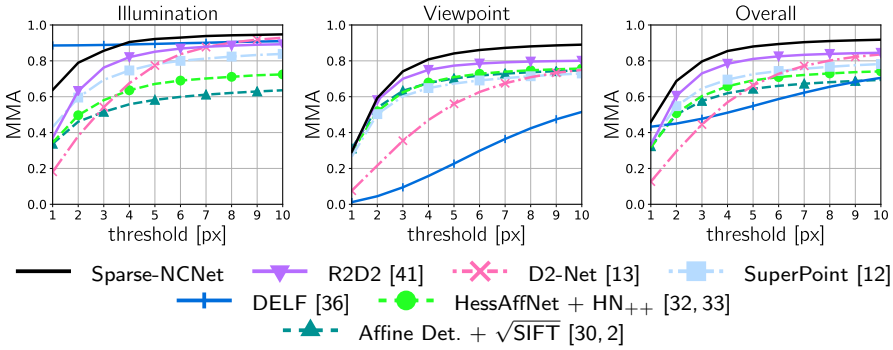


Fig. 5: **Sparse-NCNet vs. state-of-the-art on HPatches.** The MMA of Sparse-NCNet and several state-of-the-art methods is shown. Sparse-NCNet obtains the best results overall with a large margin over the recent R2D2 method.

Sparse-NCNet vs. state-of-the-art methods. In addition, we compare the performance of Sparse-NCNet against several methods, including state-of-the-art trainable methods such as SuperPoint [12], D2-Net [13] or R2D2 [41]. The mean-matching accuracy results are presented in Fig. 5. For all other methods, the top 2000 features points where selected from each image, and matched enforcing mutual nearest-neighbours, yielding approximately 1000 correspondences per image pair. For Sparse-NCNet, the top 1000 correspondences where selected for each image pair, for a fair comparison. Sparse-NCNet obtains the best results for the *illumination* sequences for thresholds higher than 4 pixels, and in the *viewpoint* sequences for all threshold values. Sparse-NCNet obtains the best results overall, with a large margin over the state-of-the-art R2D2 method. We believe this could be attributed to the usage of dense descriptors (which avoid the loss of detections) together with an increased matching robustness from performing neighbourhood consensus. Qualitative examples and a comparison with other methods are presented in the extended version of this work [42].

4.2 InLoc benchmark

The InLoc benchmark [53] targets the problem of indoor localisation. It contains a set of *database* images of a building, obtained with a 3D scanner, and a set of *query* images from the same building, captured with a cell-phone several months later. The task is then to obtain the 6-dof camera positions of the query images. We follow the DensePE approach proposed [53] to find the top 10 candidate database images for each query, and employ Sparse-NCNet to obtain matches between them. Then, we follow again the procedure in [53] to obtain the final estimated 6-dof query pose, which consists of running PnP [14] followed by dense pose verification [53].

The results are presented in Fig. 6. First, we observe that Sparse-NCNet with hard relocalisation (H) and a resolution of 100×75 obtains equivalent results to NCNet (methods B vs. C), while being almost $7\times$ faster and requiring $6.5\times$ less memory, confirming what was already observed in the HPatches benchmark (*c.f.* B1 vs. B2 in Fig. 4a). Moreover, our proposed Sparse-NCNet method with two-stage relocalisation (H+S) in the higher 200×150 resolution (method A) obtains the best results and sets a new state-of-the-art for this benchmark. Recall that it is impossible to use the original NCNet on the higher resolution due to its excessive memory requirements. Qualitative examples are included in the extended version [42].

4.3 Aachen Day-Night

The Aachen Day-Night benchmark [47] targets 6-dof outdoor camera localisation under challenging illumination conditions. It contains 98 night-time query images from the city of Aachen, and a shortlist of 20 day-time images for each night-time query. Sparse-NCNet is used to obtain matches between the query and images in the short-list. The resulting matches are then processed by the 3D reconstruction software COLMAP [50] to obtain the estimated query poses.

Table 1: **Results on Aachen Day-Night.** Sparse-NCNet is able to localise a similar number of queries as R2D2 and D2-Net.

Method	Localised (%) 0.5m, 2°	1.0m, 5°	5.0m, 10°
RootSIFT [28, 2]	36.7	54.1	72.5
DenseSfM [47]	39.8	60.2	84.7
HessAffNet + HN++ [32, 33]	39.8	61.2	77.6
DELF [36]	38.8	62.2	85.7
SuperPoint [12]	42.8	57.1	75.5
D2-Net [13]	44.9	66.3	**88.8**
D2-Net (Multi-scale) [13]	44.9	64.3	**88.8**
R2D2 (patch = 16) [41]	44.9	67.3	87.8
R2D2 (patch = 8) [41]	**45.9**	66.3	**88.8**
Sparse-NCNet (H, 200 × 150)	44.9	**68.4**	86.7

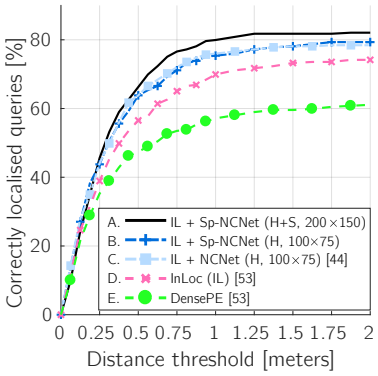


Fig. 6: **Results on the InLoc benchmark.** Our proposed method (A) obtains state-of-the-art results on this benchmark.

The results are presented in Table 1. Sparse-NCNet presents a similar performance to the state-of-the-art methods D2-Net [13] and R2D2 [41]. Note that the results of these three different methods differ by only a few percent, which represents only 1 or 2 additionally localised queries, from the 98 total night-time queries. The proposed Sparse-NCNet obtains state-of-the-art results for the 1m and 5° threshold, being able to localise 68.4% of the queries (67 out of 98). Qualitative examples are shown in Fig. 1 and in the extended version [42].

5 Conclusion

In this paper we have developed Sparse Neighbourhood Consensus Networks for efficiently estimating correspondences between images. Our approach overcomes the main limitations of the original Neighbourhood Consensus Networks that demonstrated promising results on challenging matching problems, making these models practical and widely applicable. The proposed model jointly performs feature extraction, matching and robust match filtering in a computationally efficient manner, outperforming state-of-the-art results on two challenging matching benchmarks. The entire pipeline is end-to-end trainable, which opens-up the possibility for including additional modules for specific downstream problems such as camera pose estimation or 3D reconstruction.

Acknowledgements. This work was partially supported by the European Regional Development Fund under project IMPACT (reg. no. CZ.02.1.01/0.0/0.0/15 003/0000468), Louis Vuitton ENS Chair on Artificial Intelligence, and the French government under management of Agence Nationale de la Recherche as part of the "Investissements d'avenir" program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute).

References

1. Arandjelović, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: NetVLAD: CNN architecture for weakly supervised place recognition. In: CVPR (2016)
2. Arandjelović, R., Zisserman, A.: Three things everyone should know to improve object retrieval. In: Proc. CVPR. pp. 2911–2918 (2012)
3. Balntas, V., Lenc, K., Vedaldi, A., Mikolajczyk, K.: HPatches: A Benchmark and Evaluation of Handcrafted and Learned Local Descriptors. In: Proc. CVPR (2017)
4. Balntas, V., Hammarstrand, L., Heijnen, H., Kahl, F., Maddern, W., Mikolajczyk, K., Pajdla, T., Pollefeys, M., Sattler, T., Schönberger, J.L., Speciale, P., Sivic, J., Toft, C., Torii, A.: Workshop in Long-Term Visual Localization under Changing Conditions, CVPR 2019. https://www.visuallocalization.net/workshop/cvpr/2019/
5. Balntas, V., Johns, E., Tang, L., Mikolajczyk, K.: PN-Net: Conjoined triple deep network for learning local image descriptors. arXiv preprint arXiv:1601.05030 (2016)
6. Balntas, V., Riba, E., Ponsa, D., Mikolajczyk, K.: Learning local feature descriptors with triplets and shallow convolutional neural networks. In: Proc. BMVC. (2016)
7. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: Speeded up robust features. In: ECCV (2006)
8. Bian, J., Lin, W.Y., Matsushita, Y., Yeung, S.K., Nguyen, T.D., Cheng, M.M.: GMS: Grid-based motion statistics for fast, ultra-robust feature correspondence. In: Proc. CVPR (2017)
9. Brachmann, E., Rother, C.: Neural-guided RANSAC: Learning where to sample model hypotheses. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4322–4331 (2019)
10. Choy, C., Gwak, J., Savarese, S.: 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In: Proc. CVPR (2019)
11. Choy, C., Park, J., Koltun, V.: Fully convolutional geometric features. In: Proc. ICCV (2019)
12. DeTone, D., Malisiewicz, T., Rabinovich, A.: SuperPoint: Self-Supervised Interest Point Detection and Description. In: CVPR Workshops (2018)
13. 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. In: Proc. CVPR (2019)
14. Gao, X.S., Hou, X.R., Tang, J., Cheng, H.F.: Complete solution classification for the perspective-three-point problem. IEEE PAMI **25**(8), 930–943 (2003)
15. Germain, H., Bourmaud, G., Lepetit, V.: Sparse-to-Dense Hypercolumn Matching for Long-Term Visual Localization. In: 3DV (2019)
16. Girshick, R.: Fast R-CNN. In: Proc. ICCV (2015)
17. Gojcic, Z., Zhou, C., Wegner, J.D., Guibas, L.J., Birdal, T.: Learning multiview 3D point cloud registration. arXiv preprint arXiv:2001.05119 (2020)
18. Grabner, A., Roth, P.M., Lepetit, V.: 3D Pose Estimation and 3D Model Retrieval for Objects in the Wild. In: Proc. CVPR (2018)
19. Graham, B.: Sparse 3D convolutional neural networks. arXiv preprint arXiv:1505.02890 (2015)
20. Graham, B.: Spatially-sparse convolutional neural networks. arXiv preprint arXiv:1409.6070 (2014)
21. Graham, B., Engelcke, M., van der Maaten, L.: 3D semantic segmentation with submanifold sparse convolutional networks. In: Proc. CVPR (2018)
22. Han, X., Leung, T., Jia, Y., Sukthankar, R., Berg, A.C.: MatchNet: Unifying feature and metric learning for patch-based matching. In: Proc. CVPR (2015)

23. Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with GPUs. arXiv preprint arXiv:1702.08734 (2017)
24. Julesz, B.: Towards the automation of binocular depth perception. In: Proc. IFIP Congress. pp. 439–444 (1962)
25. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015)
26. Laguna, A.B., Riba, E., Ponsa, D., Mikolajczyk, K.: Key.Net: Keypoint detection by handcrafted and learned CNN filters. In: Proc. ICCV (2019)
27. Lenc, K., Vedaldi, A.: Learning covariant feature detectors. In: ECCV Workshop on Geometry Meets Deep Learning (2016)
28. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV **60**(2), 91–110 (2004)
29. Marr, D., Poggio, T.: Cooperative computation of stereo disparity. Science **194**(4262), 283–287 (1976)
30. Mikolajczyk, K., Schmid, C.: An affine invariant interest point detector. In: Proc. ECCV (2002)
31. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Van Gool, L.: A comparison of affine region detectors. IJCV **65**(1-2), 43–72 (2005)
32. Mishchuk, A., Mishkin, D., Radenovic, F., Matas, J.: Working hard to know your neighbor's margins: Local descriptor learning loss. In: NIPS (2017)
33. Mishkin, D., Radenović, F., Matas, J.: Repeatability Is Not Enough: Learning Discriminative Affine Regions via Discriminability. In: Proc. ECCV (2018)
34. Moo Yi, K., Trulls, E., Ono, Y., Lepetit, V., Salzmann, M., Fua, P.: Learning to find good correspondences. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2666–2674 (2018)
35. Mori, K.I., Kidode, M., Asada, H.: An iterative prediction and correction method for automatic stereocomparison. Computer Graphics and Image Processing **2**(3-4), 393–401 (1973)
36. Noh, H., Araujo, A., Sim, J., Weyand, T., Han, B.: Large-scale image retrieval with attentive deep local features. In: Proc. ICCV (2017)
37. Ono, Y., Trulls, E., Fua, P., Yi, K.M.: LF-Net: Learning local features from images. In: NIPS (2018)
38. Oron, S., Dekel, T., Xue, T., Freeman, W.T., Avidan, S.: Best-buddies similarity—robust template matching using mutual nearest neighbors. IEEE PAMI **40**(8), 1799–1813 (2017)
39. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in PyTorch (2017)
40. Persson, M., Nordberg, K.: Lambda twist: An accurate fast robust perspective three point (P3P) solver. In: Proc. ECCV (2018)
41. Revaud, J., Weinzaepfel, P., de Souza, C.R., Humenberger, M.: R2D2: repeatable and reliable detector and descriptor. In: NeurIPS (2019)
42. Rocco, I., Arandjelović, R., Sivic, J.: Efficient neighbourhood consensus networks via submanifold sparse convolutions. https://arxiv.org/abs/2004.10566 (2020)
43. Rocco, I., Arandjelović, R., Sivic, J.: Sparse neighbouhood consensus networks. https://www.di.ens.fr/willow/research/sparse-ncnet/ (2020)
44. Rocco, I., Cimpoi, M., Arandjelović, R., Torii, A., Pajdla, T., Sivic, J.: Neighbourhood consensus networks. In: NeurIPS (2018)
45. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: An efficient alternative to SIFT or SURF. In: Proc. ICCV (2011)

46. Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: Superglue: Learning feature matching with graph neural networks. arXiv preprint arXiv:1911.11763 (2019)
47. 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: Proc. CVPR (2018)
48. Schaffalitzky, F., Zisserman, A.: Automated scene matching in movies. In: International Conference on Image and Video Retrieval (2002)
49. Schmid, C., Mohr, R.: Local grayvalue invariants for image retrieval. IEEE PAMI **19**(5), 530–535 (1997)
50. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
51. Schönberger, J.L., Zheng, E., Pollefeys, M., Frahm, J.M.: Pixelwise view selection for unstructured multi-view stereo. In: European Conference on Computer Vision (ECCV) (2016)
52. Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: Proc. ICCV (2003)
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. In: Proc. CVPR (2018)
54. Tian, Y., Fan, B., Wu, F.: L2-Net: Deep learning of discriminative patch descriptor in Euclidean space. In: Proc. CVPR (2017)
55. Torii, A., Arandjelović, R., Sivic, J., Okutomi, M., Pajdla, T.: 24/7 place recognition by view synthesis. In: CVPR (2015)
56. Verdie, Y., Yi, K., Fua, P., Lepetit, V.: TILDE: A temporally invariant learned detector. In: Proc. CVPR (2015)
57. Widya, A.R., Torii, A., Okutomi, M.: Structure from motion using dense cnn features with keypoint relocalization. IPSJ Transactions on Computer Vision and Applications **10**(1), 6 (2018)
58. Yi, K.M., Trulls, E., Lepetit, V., Fua, P.: LIFT: Learned invariant feature transform. In: Proc. ECCV (2016)
59. Zagoruyko, S., Komodakis, N.: Learning to compare image patches via convolutional neural networks. In: Proc. CVPR (2015)
60. Zhang, J., Sun, D., Luo, Z., Yao, A., Zhou, L., Shen, T., Chen, Y., Quan, L., Liao, H.: Learning two-view correspondences and geometry using order-aware network. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 5845–5854 (2019)
61. Zhang, Z., Deriche, R., Faugeras, O., Luong, Q.T.: A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. Artificial intelligence **78**(1-2), 87–119 (1995)
62. Zhao, W.L., Jégou, H., Gravier, G.: Oriented pooling for dense and non-dense rotation-invariant features. In: Proc. BMVC. (2013)
63. Zhou, H., Sattler, T., Jacobs, D.W.: Evaluating local features for day-night matching. In: Proc. ECCV (2016)