DualRC: A Dual-Resolution Learning Framework with Neighbourhood Consensus for Visual Correspondences

Xinghui Li, Kai Han, Shuda Li, Victor Prisacariu

Abstract—We address the problem of establishing accurate correspondences between two images. We present a flexible framework that can easily adapt to both geometric and semantic matching. Our contribution consists of three parts. Firstly, we propose an end-to-end trainable framework that uses the coarse-to-fine matching strategy to accurately find the correspondences. We generate feature maps in two levels of resolution, enforce the neighbourhood consensus constraint on the coarse feature maps by 4D convolutions and use the resulting correlation map to regulate the matches from the fine feature maps. Secondly, we present three variants of the model with different focuses. Namely, a universal correspondence model named DualRC that is suitable for both geometric and semantic matching, an efficient model named DualRC-L tailored for geometric matching with a lightweight neighbourhood consensus module that significantly accelerates the pipeline for high-resolution input images, and the DualRC-D model in which we propose a novel dynamically adaptive neighbourhood consensus module (DyANC) that dynamically selects the most suitable non-isotropic 4D convolutional kernels with the proper neighbourhood size to account for the scale variation. Last, we thoroughly experiment on public benchmarks for both geometric and semantic matching, showing superior performance in both cases.

Index Terms—Geometric matching, semantic matching, dense matching, correspondence estimation.

1 Introduction

Establishing correspondences across images is a fundamental problem in computer vision. A common objective is to find the projections of the same 3D geometric points of the objects in different images. Many computer vision applications rely on robust correspondences, such as structure from motion [1], [2] and visual localization [3], [4]. This problem is accompanied by several critical challenges, including viewpoint change, illumination variation, and the presence of repetitive patterns. The key to tackling this problem is generating discriminative features that distinguish different keypoints. Many handcrafted features, such as SIFT [5], SURF [6], BRISK [7], have been proposed in the past two decades. However, they fail to demonstrate sufficient robustness under the aforementioned challenging conditions. Modern methods [8], [9], [10], [11], [12], [13], [14], [15], [16], [17] are dominated by features generated by the convolutional neural network (CNN), demonstrating superior performance over handcrafted features. Despite improving the performance of establishing correspondences between two different images of the same object (a.k.a. geometric correspondences), the advancement of the learned CNN features also enables the possibility of establishing correspondences between images of different instances from the same category (a.k.a. semantic correspondences), which is more challenging than geometric correspondences due to the extra challenges of huge intra-class appearance and scale variation. Many efforts [18], [19], [20], [21], [22], [23], [24], [25] have been devoted to tackling this challenge in the past few years, showing encouraging results.

A classic pipeline to this problem consists of three stages, namely, detection, description, and matching, including methods like [5], [6], [8], [9], [11], [12], [17]. These methods first detect keypoints across two images and then describe these keypoints by extracting feature vectors from local regions around them. Correspondence pairs are finally established based on similarities between features. Although these methods have demonstrated great successes, they still suffer from the missing detection problem [26]. Matches can only be found between detected keypoints, limiting the matching space. Alternatively, works like [18], [19], [20], [21], [22], [23], [24], [25], [26], [27] focus on establishing dense correspondences, i.e. finding the correspondence for every pixel in the query image. The searching space in these works is the entire target image, remedying the missing detection problem, which, however, introduces more computational overhead, limiting their practical applications.

Among these approaches, the Neighbourhood Consensus (NC) family [24], [25], [26], [28] have shown promising results. These methods employ a CNN to extract features from two images before calculating a 4D correlation tensor representing the entire matching space where each cell records the cosine correlation score between a pair of feature vectors. This tensor is then processed by a neighbourhood consensus module, which consists of a sequence of 4D convolution operations, to effectively filter incorrect matches. However, there are two main limitations in the vanilla 4D convolution [24]. (i) Due to the $O(n^4)$ complexity, it is prohibitively expensive to process high-resolution images, which is vital for geometric correspondence estimation. (ii)

- *Xinghui Li, Shuda Li and Victor Prisacariu are members of Active Vision Lab of the University of Oxford*
  Email:{xinghui, shuda, victor}@robots.ox.ac.uk
- *Kai Han is with The University of Hong Kong*
  Email: kaihanx@hku.hk
- *Corresponding author: Kai Han*

The neighbourhoods of the same size are considered for any pair of locations from the two images. This is inappropriate because the scales and shapes of objects vary a lot, especially for semantic correspondence estimation, as the irrelevant neighbourhood may negatively affect matching.

In this work, we address the above limitations of existing methods based on neighbourhood consensus and make the following three main contributions. (1) We propose a dual-resolution framework to generate feature maps at two levels of resolution for matching. The coarse-resolution feature maps are used to form a complete 4D correlation tensor, which is refined by a 4D neighbourhood consensus module. The refined 4D tensor is used to guide matching on fine-resolution feature maps. This significantly improves the matching performance without requiring the expensive 4D tensor on the fine-resolution feature maps. (2) We present three variants of our framework. The first one is a universal model named DualRC which has the basic NC module consisting of isotropic 4D convolutional kernels. It is suitable for both geometric and semantic correspondence estimation. The second one is named as DualRC-L, which consists of a lightweight NC module based on sparse correlation [26] with the aim to accelerate the pipeline in case of high-resolution input images in geometric matching. The third one is DualRC-D, in which we propose the novel dynamically adaptive neighbourhood consensus (DyANC) module that leverages non-isotropic 4D convolutional kernels and dynamically selects the best neighbourhood region to handle scale variation in semantic tasks. (3) We thoroughly evaluate our approach on public benchmarks for both geometric and semantic matching obtaining superior performance, including HPatches [29], InLoc [4], Aachen Day-Night [30], PF-PASCAL [31], PF-WILLOW [31] and SPair-71k [32].

Preliminary results of our basic DualRC have been published in [33]. In this paper, we have made three main extensions. First, we propose DualRC-L, a lightweight variant of DualRC using sparse 4D convolution kernels to reduce computational complexity significantly; second, we propose DualRC-D, a scale-aware variant of DualRC, at the core of which is our novel dynamically adaptive neighbourhood consensus (DyANC) module, allowing the model to dynamically choose 4D filters of varying sizes, which is different from [25] that only possesses fixed-size neighbourhoods without a self-adaptation mechanism to the actual context; third, we extend the evaluation of our approach from the geometric correspondence estimation task to the semantic correspondence estimation task, and also carry out cross-task evaluation for the state-of-the-art methods developed respectively for each of the task. Unlike existing methods that can only work well on one task, our method consistently shows promising results on both tasks, demonstrating that DualRC is a strong method for general visual correspondence estimation.

Our code is publicly available at https://code.active.vision.

2 Related Works

The classic pipeline for correspondence estimation consists of three stages: detection, description, and matching. In the early years, the works in the literature focused on manually developing robust local feature vectors, which is the description stage. A common objective of the design [5], [6], [34] is the invariance of the feature to the rotation and scale of the local patches extracted around the keypoints. These are achieved through rotating the patches back to canonical angles and scale normalization on patches at multiple scales. Common types of information encoded in the features are orientation histogram [5], output of the haar wavelet [6] or result of random binary test [35]. Correspondences are then established based on the nearest neighbour between features, using metrics such as the L2 distance or the cosine similarity. Additional verification like the mutual nearest neighbour or the second nearest neighbour ratio test could be applied to increase the robustness and accuracy of the matching. Although handcrafted features are popular due to their speed and simplicity of implementation, they fail to demonstrate sufficient robustness and distinctiveness under challenging cases like large viewpoint and illumination changes as well as the intra-category appearance change.

learned local features [11], [12], [13], [14], [36] are usually generated by the convolutional neural network (CNN). The majority of the works are supervised by triplet loss, which is designed to minimise the distance between features of the positive pair of correspondences while maximise the distance between features of the negative pair. A number of variations of the triplet loss have been proposed. Tian et al. [11] proposed to maximise positive feature pairs from intermediate layers of the model alongside the pairs from the output layer. Later, they further proposed to maximise the second-order similarity of the positive feature pair, the similarity between distances of the select feature to all other features [13]. Compared with handcrafted features, learned features have shown superiority under challenging conditions. However, since features are only based on local image patches, without contextual information, they are indiscriminative to the repetitive pattern whose local information is almost identical.

Recently, there are works [8], [9], [15], [16], [17] that combine detection and description stages. Both positions of keypoints and feature vectors are produced by a single forward pass of the image through the network, rather than the detection first then followed by the description. They start with extracting a dense feature map from image using feature extractor like VGG [37] or ResNet [38], and keypoints' locations and feature vectors are produced by different modules. Dusmanu et al. [8] and Luo et al. [15] interpret the feature map not only as the feature map but also as a response map like that in DoG [5] detector. Keypoints are selected at the location where the response is maximum in both spatial and channel dimensions. Works done by Detone et al. [17] and Revaud et al. [9] append two modules after the feature extractor, one to predict keypoints location and one to generate feature respectively. The advantage of this line of work is that features are encoded with not only local information but also contextual information. Nonetheless, the pipeline may still suffer from the missing detection problem as matching is still confined within detected keypoints.

Detector-free dense correspondence matching has drawn more and more attention in the last few years. The model is

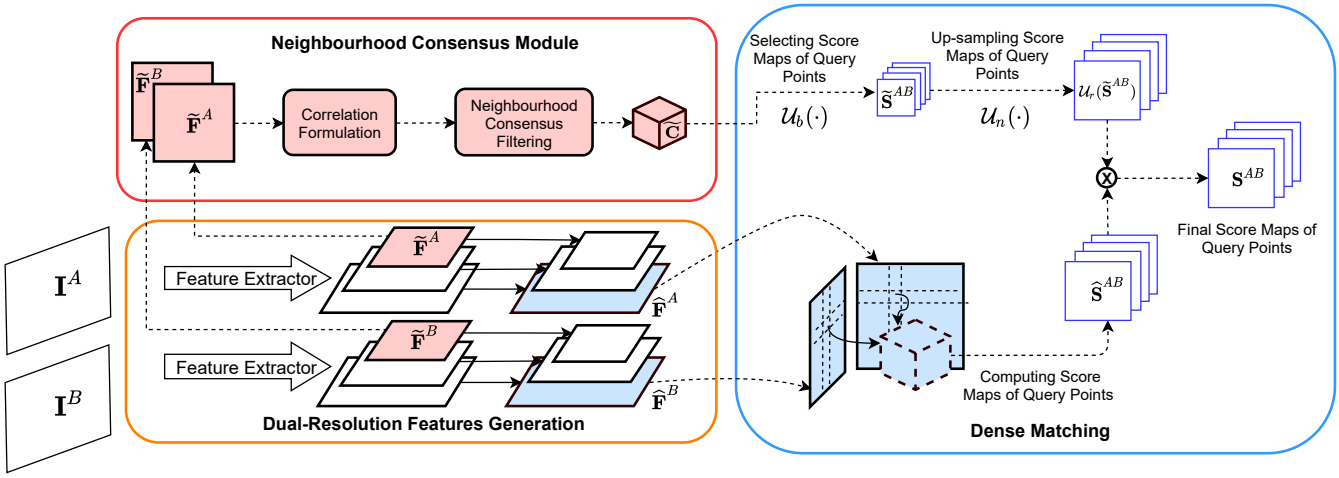


Fig. 1: The architecture of our model. It consists of three major parts: dual-resolution features generation; neighbourhood consensus module and dense matching. Firstly, feature maps at two levels of resolution are generated. The coarse resolution feature maps form 4D correlation tensor which is filtered by neighbourhood consensus module. We present three options for NC module, which are illustrated in fig. 3. After the filtering, the matching score maps of queries are extracted from filtered 4D tensor, upsampled and multiplied with the score maps computed from fine resolution feature maps, obtaining the final score maps.

required to find the correspondence for every pixel in the query image. Similar to combined detection and description methods, the model normally starts with a feature extractor to generate feature maps of both images, followed by the dense matching module. Since matching space is extended to the entire target image, the missing detection issue is trivial under this pipeline, with the trade off being the speed of the model. Some works focus specifically on geometric matching problem. Germain et al. [39] computed and merged a collection of matching score maps from outputs of different layers of the feature backbone. Wang et al. [40] proposed a coarse-to-fine architecture to find correspondences progressively and used epipolar lines as a part of the supervision. Wiles et al. [10] generated features conditioned on target images using the attention mechanism [41] to increase the accuracy of the matching. Very recently, inspired by the success of employing the transformer [41] on image classification and object detection [42], [43], [44], several works [45], [46] that apply the transformer to the correspondence task have been proposed. They vectorize the feature map, send the sequence to transformer and aim to capture the relationship between features. The results of these works are very promising, but the computing expense of the transformer forbids the use of high resolution feature maps. Although the final positions of the correspondences can be fine-tuned by recursively applying the model, such strategy would induce a very long computing time.

There are also works which focus on semantic matching. Han et al. [18] incorporated Probablistic Hough Matching into their trainable model to vote on extracted patch pairs, based on their geometric offsets and appearance similarity. Min et al. [20] employed a similar idea, but improved it with learned geometric offset from 6D convolution, greatly increasing the accuracy. Liu et al. [19] computed a primary correlation map from feature maps and formulated problems as an optimal transport problem. Rocco et al. [22] instead appended a regressor after the correlation map to estimate the parameters of geometric transformation. Inspired by RANSAC, they improved the pipeline in [23] by adding a soft-inlier counting module to reduce the impact the outliers on the result. Seo et al. [21] used an order-aware attentive module to predict the global transformation.

Our DualRC and its variants belong to the line of detector-free dense correspondence matching methods. The most relevant works to our paper is the Neighbourhood Consensus Network [24] and its variants [25], [26], [47]. Rocco et al. [26] replaced the dense 4D correlation tensor with a sparse one constructed by online k nearest neighbour search, which greatly accelerated the model. However, the performance in semantic matching dropped significantly. Li et al. [25] used a collection of non-isotropic 4D kernels to deal with different sizes in neighbourhood areas. Although it boosts the accuracy in semantic matching, the additional computation brought by extra 4D kernels makes the use of high resolution feature maps even harder. Tinchev et al. [47] de-parameterised the neighbourhood consensus filtering modules and systematically studied the effects of different image resolutions for geometric correspondence estimation.

3 Method

In this section, we elaborate our method in detail. The architecture of our model is illustrated in fig. 1. Our model mainly consists of three modules, namely, dual-resolution features generation (section 3.1), neighbourhood consensus (NC) module (section 3.2), and dense matching (section 3.3). The model starts with extracting coarse-resolution feature maps $\widetilde{\mathbf{F}}^A, \widetilde{\mathbf{F}}^B$ and fine-resolution feature maps $\widehat{\mathbf{F}}^A, \widehat{\mathbf{F}}^B$ from images $\mathbf{I}^A, \mathbf{I}^B$. The coarse feature maps form 4D correlation tensor $\mathbf{C}$, which is further processed by the NC module, obtaining refined correlation tensor $\widetilde{\mathbf{C}}$. We present three options for NC module, which are described in detail in section 3.2. After filtering, the matching score maps $\widetilde{\mathbf{S}}^{AB}$

of queries are extracted from $\widetilde{\mathbf{C}}$ which are upsampled to reweigh the matching score maps $\widehat{\mathbf{S}}^{AB}$ computed from fine-resolution feature maps, obtaining the final score map $\mathbf{S}^{AB}$. In such way, we can obtain a high-resolution correlation map to accurately localize correspondences, without computing and filtering the expensive high-resolution 4D correlation tensor.

3.1 Dual Resolution Feature Generation

It has been reported in the literature [24], [26], [32] that the resolution of feature maps that constitute the 4D correlation tensor affects the accuracy. However, the high memory and computation cost of 4D tensors prevents vanilla 4D convolution from scaling to large feature maps. To avoid calculating the expensive 4D tensor of the high-resolution feature maps, we propose to use the dual-resolution feature maps, from which we can enjoy the benefits of both 4D convolution based neighbourhood consensus on coarse-resolution feature maps and more reliable matching from the fine-resolution feature maps.

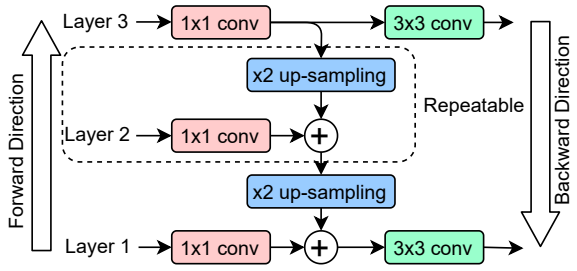


Fig. 2: The architecture of dual resolution feature generation. Features are converted to a uniform dimension by 1×1 convolutional kernel. The top feature map is upsampled and added with the bottom feature map. Finally, both feature maps are processed by 3×3 convolutional kernels for smoothing.

In particular, we adopt a FPN-like [48] feature backbone, which is illustrated in fig. 2. The feature extractor generates a pyramid of feature maps in forward direction. Top layers are spatially smaller but contain rich contextual information, while bottom layers are spatially larger with low-level details. The intuition of using FPN is that it fuses the contextual information from top into bottom layers in backward direction. In this way, the bottom layers can not only preserve their resolution but also contain rich high-level information, which makes them more robust for matching.

As shown in fig. 2, feature maps firstly pass through a 1×1 convolutional kernel to be converted into a uniform feature dimension. The top layer is then upsampled to the same size as the next layer and two feature maps are added together. This process starts from coarse-resolution feature map and is repeated until reaching the fine-resolution feature map. Two feature maps are finally processed by 3×3 convolutional kernels for smoothing. Such a structure gives great flexibility. In principle, we can select any layer as the coarse or fine resolution feature map and final feature maps can have any feature dimension.

3.2 Enforcing Neighbourhood Consensus

Enforcing neighbourhood consensus constraint by 4D convolutions is firstly proposed by [24]. It has shown promising results in both semantic and geometric tasks. By combining the basic NC module with the dual-resolution feature structure, we show that the accuracy of the correspondences can be boosted with a larger matching resolution, as demonstrated in our preliminary study [33]. Here, we present three choices for NC module. The first one is the basic NC module [24] with isotropic 4D convolutional kernels which has a balanced capability on both geometric and semantic tasks. In order to further optimize the performance in these two types of tasks, we introduce another two variants to tackle the challenges encountered in them respectively. For geometric task, which requires high-resolution input as well as fast-inference time, we present a lightweight NC module based on sparse correlation and convolution [26] to accelerate the model. For semantic task, where scale variation is prominent, we propose the DyANC module to automatically select the best neighbourhood region. We illustrate these three options in fig. 3. We denote the model with the basic NC module as DualRC, the model with lightweight NC module as DualRC-L, and the one with DyANC module as DualRC-D. Next, we introduce each part of our framework in detail, namely, coarse correlation, mutual nearest neighbour filtering, and the three variants of NC module.

3.2.1 *Coarse Correlation Tensor*

Assume that the coarse-resolution feature maps computed by the feature backbone are $\widetilde{\mathbf{F}}^A \in \mathbb{R}^{d \times h_a \times w_a}$ and $\widetilde{\mathbf{F}}^B \in \mathbb{R}^{d \times h_b \times w_b}$, with $\tilde{\mathbf{f}}^A_{ij}, \tilde{\mathbf{f}}^B_{kl} \in \mathbb{R}^d$ being the features in spatial location (i, j) and (k, l) in $\widetilde{\mathbf{F}}^A$ and $\widetilde{\mathbf{F}}^B$ respectively. We compute the coarse 4D correlation tensor by:

$$\mathbf{C}_{ijkl} = \frac{\tilde{\mathbf{f}}^{A\top}_{ij} \tilde{\mathbf{f}}^B_{kl}}{\|\tilde{\mathbf{f}}^A_{ij}\|_2 \|\tilde{\mathbf{f}}^B_{kl}\|_2} \tag{1}$$

where $\mathbf{C} \in \mathbb{R}^{h_a \times w_a \times h_b \times w_b}$ and $\|.\|_2$ represents the L-2 norm operation. Each $\mathbf{C}_{ijkl}$ in $\mathbf{C}$ is the cosine similarity between features $\tilde{\mathbf{f}}^A_{ij}$ and $\tilde{\mathbf{f}}^B_{kl}$ after L-2 normalization.

3.2.2 *Soft Mutual Nearest Neighbour Filtering*

After generating correlation tensor $\mathbf{C}$, we perform the differentiable soft mutual nearest neighbour filtering following [24]. The objective is to enforce reciprocal constraint on features. That is, if location (a, b) in image $\mathbf{I}^A$ and (c, d) in image $\mathbf{I}^B$ are true correspondence, the feature $\tilde{\mathbf{f}}^A_{ab}$ and $\tilde{\mathbf{f}}^B_{cd}$ should be their mutual nearest neighbour. Mathematically, this can be expressed as:

$$(\tilde{\mathbf{f}}^A_{ab}, \tilde{\mathbf{f}}^B_{cd}) \text{ M.N.N.} \iff \begin{cases} (a, b) = \text{argmax}_{ij} \|\tilde{\mathbf{f}}^A_{ij} - \tilde{\mathbf{f}}^B_{cd}\| \\ (c, d) = \text{argmax}_{kl} \|\tilde{\mathbf{f}}^A_{ab} - \tilde{\mathbf{f}}^B_{kl}\| \end{cases} \tag{2}$$

However, such (hard) mutual nearest neighbour filtering may result in excessive sparsity in the correlation tensor by eliminating majority of the correlation scores. Therefore, soft mutual nearest neighbour filtering $\widehat{\mathbf{C}} = \mathcal{M}(\mathbf{C})$ is applied instead with:

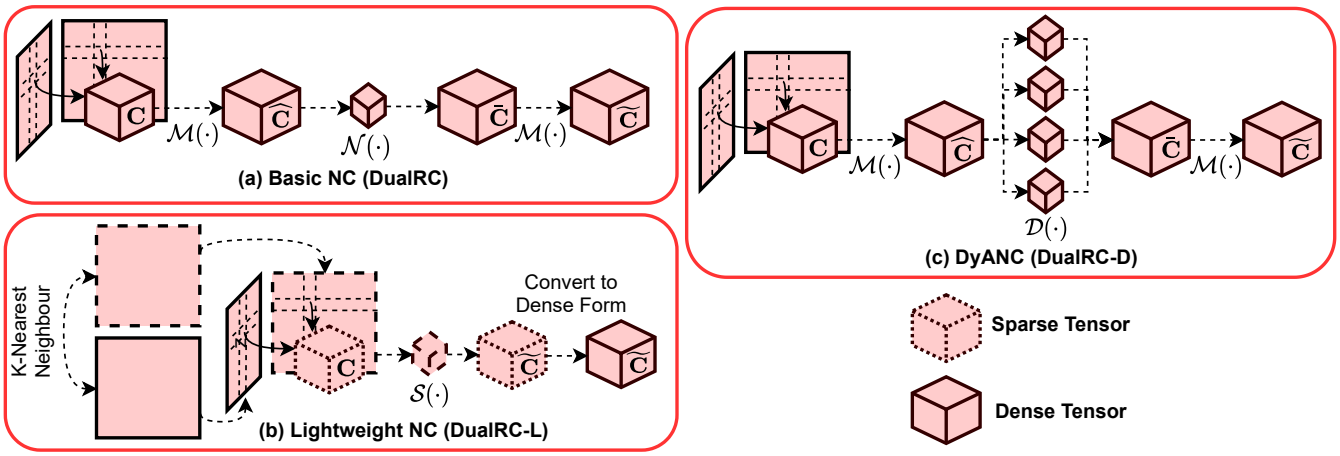


Fig. 3: Three options for neighbourhood consensus module. The first one is the basic neighbourhood consensus module [24] which works on both semantic and geometric matching. The second is a lightweight NC module based on sparse 4D correlation [26] which is to accelerate the model in the case of high resolution input images in geometric matching. The third one is our newly proposed DyANC module to tackle scale variation in semantic matching.

$$\widehat{\mathbf{C}}_{ijkl} = r^A_{ijkl} r^B_{ijkl} \mathbf{C}_{ijkl} \tag{3}$$

where

$$r^A_{ijkl} = \frac{\mathbf{C}_{ijkl}}{\max_{ab} \mathbf{C}_{abkl}} \quad \text{and} \quad r^B_{ijkl} = \frac{\mathbf{C}_{ijkl}}{\max_{cd} \mathbf{C}_{ijcd}} \tag{4}$$

If (i, j) and (k, l) are in fact the mutual nearest neighbour to each other, both r^A_{ijkl} and r^B_{ijkl} would be 1 and the original correlation score would be preserved. Otherwise, either r^A_{ijkl} or r^B_{ijkl} would be low and consequently the correlation score is suppressed. Such an operation can be regarded as a filter which alleviates the impact of the incorrect matches in the correlation tensor. It is applied to the 4D tensor before and after the NC module.

3.2.3 Basic NC module

The DualRC model uses the basic NC module which consists of a sequence of isotropic 4D convolutional kernels to filter the correlation tensor. The intuition is that if two features are correctly matched, the features themselves and their surrounding neighbourhood should have high consistency and subsequently the high responses in the 4D correlation tensor. Therefore, 4D convolutions are applied to learn such consistency from training data. Let $\mathcal{N}(\cdot)$ denote the basic NC module, which consists of a sequence of 4D kernels (we refer readers to [24] for the mathematical details of $\mathcal{N}(\cdot)$). The filtered correlation tensor is then $\mathcal{N}(\widehat{\mathbf{C}})$. In order to make the model invariant to the order of the input image pair, we follow [24] and apply the NC module symmetrically:

$$\overline{\mathbf{C}} = \mathcal{N}(\widehat{\mathbf{C}}) + (\mathcal{N}(\widehat{\mathbf{C}}^\top))^\top \tag{5}$$

where $\top$ denotes swapping matching direction following the notation of [24], i.e., $\widehat{\mathbf{C}}^\top_{ijkl} = \widehat{\mathbf{C}}_{klij}$. The final tensor $\widetilde{\mathbf{C}}$ is obtained by applying another mutual nearest neighbour filtering $\widetilde{\mathbf{C}} = \mathcal{M}(\overline{\mathbf{C}})$.

3.2.4 Lightweight NC module

Although the dual-resolution feature generation increases the matching resolution without the significant extra cost, we find that the speed of the pipeline is still somewhat unsatisfying for high resolution input images. It is because that coarse-resolution correlation tensors for high-resolution images are already computationally expensive to be processed. This problem is particularly significant in geometric matching which relies on high-resolution inputs for higher accuracy. Therefore, it is desired to have a more efficient model for geometric matching while having strong performance. To do so, instead of applying the basic NC module, we introduce a computationally efficient neighbourhood consensus module, which draws inspiration from [26]. Particularly, we replace the dense correlation tensor with a sparse one, which is then filtered by sparse 4D convolutions [49]. The soft mutual nearest neighbour filtering is also dropped to improve efficiency.

Concretely, instead of maintaining scores for all possible feature pairs with a dense 4D correlation tensor, for each $\tilde{\mathbf{f}}^A_{ij}$ in source image, we only keep the correlation scores of its k nearest neighbours in $\tilde{\mathbf{F}}^B$, forming a sparse correlation tensor $\mathbf{C}^{AB}$:

$$\mathbf{C}^{AB}_{ijkl} = \begin{cases} \tilde{\mathbf{f}}^{A\top}_{ij} \tilde{\mathbf{f}}^B_{kl} & \text{if } \tilde{\mathbf{f}}^B_{kl} \text{ is within } k \text{ N.N. of } \tilde{\mathbf{f}}^A_{ij} \\ 0 & \text{otherwise} \end{cases}. \tag{6}$$

Similarly, we can obtain $\mathbf{C}^{BA}$ by swapping the matching direction. The correlation tensor considering both matching direction is then given by:

$$\mathbf{C} = \mathbf{C}^{AB} + \mathbf{C}^{BA}. \tag{7}$$

Maitaining such a sparse 4D tensor $\mathbf{C}$ requires much less memory footprint than its dense counterpart. The sparse tensor $\mathbf{C}$ is then filtered by sparse 4D convolution filters $\mathcal{S}(\cdot)$ with a similar symmetric arrangement to ensure the order-invariant nature to the input image pair:

$$\widetilde{\mathbf{C}} = \mathcal{S}(\mathbf{C}) + \left(\mathcal{S}(\mathbf{C}^\top)\right)^\top. \tag{8}$$

3.2.5 Dynamically Adaptive NC module

The DualRC model uses isotropic 4D convolutional kernels, such as $3 \times 3 \times 3 \times 3$ or $5 \times 5 \times 5 \times 5$ which has been proven to be effective in filtering out incorrect matches. Li et al. [25] pointed out that isotropic kernels fail to adapt to scale variation across two images, a problem commonly exists in semantic tasks. They instead proposed to use a collection of both isotropic and non-isotropic kernels, kernels like $3 \times 3 \times 5 \times 5$, to tackle such an issue. Although this improves the accuracy, they directly combine the outputs of different kernels into a single 4D tensor, and each output is treated equally. We argue such a combination is sub-optimal since the scale variation changes across different pixel locations, i.e., cells in the 4D correlation map may have largely different scale changes. Therefore, it is desired to allow the model to dynamically adopt the most suitable combination of different kernels for each cell automatically, such that each pair of locations in the two images can be treated with the most suitable isotropic or non-isotropic kernels. To this end, we develop the dynamically adaptive neighbourhood consensus (DyANC) module for semantic tasks.

Our DyANC module $\mathcal{D}(\cdot)$ consists of *four* different 4D kernels as the basic components, $3\times3\times3\times3$, $3\times3\times5\times5$, $5\times5\times3\times3$ and $5\times5\times5\times5$ to account for different possible neighbourhood variations. To allow the model to automatically choose the most suitable combination of kernels, we additionally define another $3\times3\times3\times3$ kernel to predict *four* values for each cell in the 4D tensor. These values are used to weigh the outputs after applying the four different isotropic and non-isotropic kernels. Let $\mathcal{N}_i^l(\cdot)$ be the 4D convolutional operation at l-th layer with i-th kernel, where $i = 1, 2, ..., 4$ corresponding to the for different kernels, and $\mathcal{W}^l(\cdot)$ be the weight-learning kernel operation at layer l. The output $\widehat{\mathbf{C}}^l$ at layer l can be obtained by:

$$\widehat{\mathbf{C}}^l = \sum_{i=1}^{4} \mathbf{W}_i^l \mathcal{N}_i^l(\widehat{\mathbf{C}}^{(l-1)}), \tag{9}$$

where $\mathbf{W}^l = \text{softmax}(\mathcal{W}^l(\widehat{\mathbf{C}}^{(l-1)}))$, $\mathbf{W}_i^l$ is the i^{th} channel of $\mathbf{W}^l$. The softmax operation is applied along the channel dimension to normalize the 4 weight values for each cell in the 4D tensor. We illustrate an example of $\mathcal{D}(\cdot)$ in fig. 4. Again, we apply our DyANC module symmetrically to make our model invariant to the order of the input image pair:

$$\widetilde{\mathbf{C}} = \mathcal{D}(\widehat{\mathbf{C}}) + (\mathcal{D}(\widehat{\mathbf{C}}^\top))^\top. \tag{10}$$

3.3 Dense Matching

The intuition of our dual-resolution structure is to use the result of NC module to refine the correlation scores computed from fine-resolution feature maps. This allows us to establish correspondences at a fine-resolution level while considering neighbourhood consensus. Since we do not apply NC module in the fine level, the computing of the expensive fine-resolution 4D correlation tensor is avoided. Given a query point, we only need to compute the 2D matching score map for it, and reweigh the score map by the output of NC module.

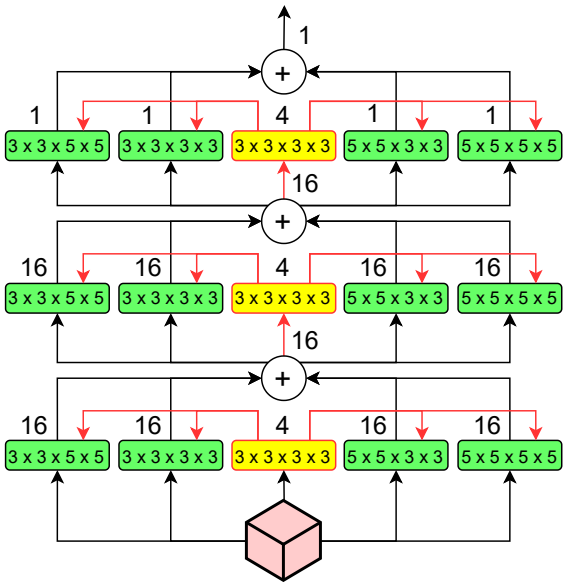


Fig. 4: An example of the dynamically adaptive NC module. The example consists of two hidden layers with 16 channels.

Given any coordinate (a, b) on $\widehat{\mathbf{F}}^A$, we can compute a correlation score w.r.t any location (k, l) on $\widehat{\mathbf{F}}^B$ using:

$$\widehat{\mathbf{S}}_{ab,kl} = \frac{\hat{\mathbf{f}}_{ab}^{A\top} \hat{\mathbf{f}}_{kl}^B}{\|\hat{\mathbf{f}}_{ab}^A\|_2 \|\hat{\mathbf{f}}_{kl}^B\|_2}, \tag{11}$$

where $\hat{\mathbf{f}}_{ab}^A$ and $\hat{\mathbf{f}}_{kl}^B$ are features at the spatial locations (a, b) and (k, l) on $\widehat{\mathbf{F}}^A$ and $\widehat{\mathbf{F}}^B$, respectively. After obtaining $\mathbf{S}_{ab,kl}$ for all (k, l), we have the fine-resolution correlation map $\widehat{\mathbf{S}}_{ab} \in \mathbb{R}^{H_b \times W_b}$. Next, the coordinate (a, b) is scaled down to the resolution of the coarse feature map, obtaining the corresponding coordinate (a', b'). Note that a', b' are float numbers rather than integers hence we use bilinear interpolation $\mathcal{B}(\cdot)$ to extract the coarse-resolution correlation map $\widetilde{\mathbf{S}}_{a'b'} \in \mathbb{R}^{h_b \times w_b}$ from coarse correlation tensor $\widetilde{\mathbf{C}}$:

$$\widetilde{\mathbf{S}}_{a'b',k'l'} = \mathcal{B}(\widetilde{\mathbf{C}}_{\lceil b' \rceil \lceil a' \rceil k'l'}, \widetilde{\mathbf{C}}_{\lceil b' \rceil \lfloor a' \rfloor k'l'}, \widetilde{\mathbf{C}}_{\lfloor b' \rfloor \lceil a' \rceil k'l'}, \widetilde{\mathbf{C}}_{\lfloor b' \rfloor \lfloor a' \rfloor k'l'}) \tag{12}$$

where $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ denote the ceiling and floor operation respectively. $\widetilde{\mathbf{S}}_{a'b'}$ is then upsampled to the same size as the high-resolution correlation map $\widehat{\mathbf{S}}_{ab}$ using the nearest neighbour upsampling method, obtaining the final correlation map $\mathbf{S}_{ab} \in \mathbb{R}^{H_b \times W_b}$:

$$\mathbf{S}_{ab} = \mathcal{U}(\widetilde{\mathbf{S}}_{a'b'}) \odot \widehat{\mathbf{S}}_{ab}, \tag{13}$$

where $\mathcal{U}(\cdot)$ denotes as nearest neighbour upsampling operation and $\odot$ represents the element-wise multiplication between two tensors. For each point (a, b) on $\widehat{\mathbf{F}}^A$, its correspondence (c, d) on $\widehat{\mathbf{F}}^B$ is then retrieved by:

$$(c, d) = \text{argmax}_{kl} \mathbf{S}_{ab,kl}. \tag{14}$$

This allows us to obtain dense correspondences from source to target or target to source.

At inference time, the model may be provided with a set of query points. If no query points is provided, we can use

the coarse-resolution feature maps to ignore the unreliable query points.

Inference with given queries. Let $\{(x_i, y_i)\}$ be the set of given queries in the image. They are firstly scaled down to the spatial resolution of fine-resolution features $\widehat{\mathbf{F}}^A$ and $\widehat{\mathbf{F}}^B$, denoted as $\{(\hat{x}_i, \hat{y}_i)\}$, which are float numbers. One can choose to round the coordinates to their nearest integer neighbours and use the nearest neighbours as the queries. However, this will induce undesirable error and reduce the accuracy of the estimated correspondences. Therefore, for a given query $(\hat{x}_i, \hat{y}_i)$ we instead find its four nearest integer neighbours $(\hat{x}_j^n, \hat{y}_j^n), \forall j = 1, ..., 4$ and their correspondences $(\hat{x}_j^{n*}, \hat{y}_j^{n*})$ by eq. (14). Finally, we use bilinear interpolation to obtain the correspondence of $(\hat{x}_i, \hat{y}_i)$ by:

$$(\hat{x}_i^*, \hat{y}_i^*) = \frac{\sum_{j=1}^4 q_j (\hat{x}_j^{n*}, \hat{y}_j^{n*})}{\sum_{j=1}^4 q_j}, \tag{15}$$

where $q_j = |\hat{x}_i - \hat{x}_j^n||\hat{y}_i - \hat{y}_j^n|$.

Inference without given queries. If no query is provided, we treat all integer coordinates $\{(\hat{x}, \hat{y})\}, \forall \hat{x} = 0, ..., W_a - 1, \forall \hat{y} = 0, ..., H_a - 1$ on the fine-resolution feature map $\widehat{\mathbf{F}}^A$ of the query image as queries and find their correspondences $\{(\hat{x}^*, \hat{y}^*)\}$. In order to reduce the impact of incorrect matches to the downstream task, we impose a cyclic consistency on the matches to filter out incorrect pairs. We treat $\{(\hat{x}^*, \hat{y}^*)\}$ as queries and find their correspondences $\{(\hat{x}^{**}, \hat{y}^{**})\}$ by swapping the order of query and target images and obtain two sets of matching pairs $\big\{\big((\hat{x}, \hat{y}), (\hat{x}^*, \hat{y}^*)\big)\big\}$ and $\big\{\big((\hat{x}^{**}, \hat{y}^{**}), (\hat{x}^*, \hat{y}^*)\big)\big\}$. We pick the intersection between $\big\{\big((\hat{x}, \hat{y}), (\hat{x}^*, \hat{y}^*)\big)\big\}$ and $\big\{\big((\hat{x}^{**}, \hat{y}^{**}), (\hat{x}^*, \hat{y}^*)\big)\big\}$ as the result of the matching. However, querying the entire set of integer coordinates in the fine-resolution is time-consuming. Thanks to the subtle design of our dual-resolution structure, we can significantly accelerate the process. Firstly, we establish a set of matches for all integer coordinates in coarse-resolution feature map $\widetilde{\mathbf{F}}^A$, denoted as $\big\{\big((\tilde{x}, \tilde{y}), (\tilde{x}^*, \tilde{y}^*)\big)\big\}, \forall \tilde{x} = 0, ..., w_a - 1, \forall \tilde{y} = 0, ..., h_a - 1$, which can be directly retrieved from $\widetilde{\mathbf{C}}$ by $(\tilde{x}^*, \tilde{y}^*) = \text{argmax}_{kl} \widetilde{\mathbf{C}}_{\tilde{x}\tilde{y}kl}$. We then sort the correlation scores of these matches in descending order and select top 50% of them. Since each spatial location on $\widetilde{\mathbf{F}}^A$ corresponds to a $r \times r$ region on $\widehat{\mathbf{F}}^A$, where r is the spatial ratio between $\widehat{\mathbf{F}}^A$ and $\widetilde{\mathbf{F}}^A$, we treat all integer coordinates in $\widehat{\mathbf{F}}^A$ that lie in the corresponding local region of the selected top 50% in $\big\{\big((\tilde{x}, \tilde{y}), (\tilde{x}^*, \tilde{y}^*)\big)\big\}$ as queries and then follow the aforementioned matching procedure using the cyclic consistency. This effectively reduces the number of queries by half and increases inference speed.

3.4 Training loss

Due to the absence of dense keypoint annotations, existing methods normally use either image-level pairwise annotations [24], [26], [28] or sparse keypoint annotations [25] for training. We follow [25] to adopt the training loss based on sparse keypoint annotations.

Given a set of ground-truth correspondence pairs $\big\{\big((x_i, y_i), (x_i^*, y_i^*)\big)\big\}_{i=1}^N$ in the image resolution, we convert them to the scale of fine-resolution feature map. We then obtain the 2D correlation map for each query following eq. (13) and generate the corresponding ground-truth 2D correlation map. As the coordinates are float numbers in the fine-resolution, we dilute the one-hot value to its four nearest neighbours in the ground-truth 2D map and apply a Gaussian kernel to smooth the label. By stacking and flattening all the N correlation maps, we can form the predicted and ground-truth assignment matrices $\mathbf{M}^{AB}, \mathbf{M}_{gt}^{AB} \in \mathbb{R}^{N \times T}$ where $T = H_b \times W_b$. Each row in the two matrices is a flattened 2D correlation map for a given query point. Considering both $A \rightarrow B$ and $B \rightarrow A$ matching directions, the loss is defined as:

$$\mathcal{L}_k = \|\mathbf{M}^{AB} - \mathbf{M}_{gt}^{AB}\|_F + \|\mathbf{M}^{BA} - \mathbf{M}_{gt}^{BA}\|_F, \tag{16}$$

where $\|.\|_F$ denotes the Frobenius norm of matrix. Besides, we include the orthogonal loss which we proposed in [25] to enforces one-to-one matching constraint and use it as a regularization term. If $\mathbf{M}$ is a perfect assignment matrix, which means that each query is assigned to only one target and no double assignment, then $\mathbf{M}\mathbf{M}^\top$ would be an identity matrix. As such, the orthogonal loss is defined as:

$$\begin{aligned}\mathcal{L}_o = \|\mathbf{M}^{AB}\mathbf{M}^{AB^\top} - \mathbf{M}_{gt}^{AB}\mathbf{M}_{gt}^{AB^\top}\|_F + \\ \|\mathbf{M}^{BA}\mathbf{M}^{BA^\top} - \mathbf{M}_{gt}^{BA}\mathbf{M}_{gt}^{BA^\top}\|_F.\end{aligned} \tag{17}$$

As we dilute the ground-truth correlation map, $\mathbf{M}_{gt}^{AB}$ is not a perfect assignment matrix and $\mathbf{M}_{gt}^{AB}\mathbf{M}_{gt}^{AB^\top}$ is a diagonally dominant matrix rather than an identity matrix. The over all training loss can then be written as

$$\mathcal{L} = \mathcal{L}_k + \lambda \mathcal{L}_o, \tag{18}$$

where λ is a weight term.

4 Experiments

4.1 Implementation Details

Our model is implemented in Pytorch [50]. We use ResNet101 [38] pretrained on ImageNet [51] as the feature extractor and truncate the modules after layer `conv4_23`. The feature extractor is frozen during training. For dual-resolution feature map generation module, we select the output of layer `conv4_23` of the feature extractor as the coarse-resolution feature map, hence the coarse-resolution feature map is $1/16$ of the input image size. Since geometric matching requires a high matching resolution to achieve good accuracy, we select the output of the layer `conv3_4` ($1/8$ of the input image size) as the fine-resolution feature map for semantic matching but the output of the layer `conv2_3` ($1/4$ of the input image size) as the fine-resolution feature map for geometric matching. All feature maps are converted to 1024 channels by 1×1 convolution in the dual-resolution feature generation module.

Following [24], [26], the geometric and semantic tasks use two different configurations of the NC module. For geometric task, the DualRC model consists of two layers of 4D kernels $3 \times 3 \times 3 \times 3$ with channels of $\{16, 1\}$. The DualRC-L and DualRC-D follows the same configuration. For semantic task, the DualRC consists of three layers of 4D kernels $5 \times 5 \times 5 \times 5$ with channels of $\{16, 16, 1\}$. The DualRC-L and DualRC-D adopt this configuration as well. The different choices are mainly attributed to two facts. Firstly,

semantic matching requires the larger neighbourhood area and deeper structure to capture the good semantic understanding of the scene due to the significant intra-category appearance change. Secondly, geometric matching requires images with higher resolution for higher accuracy, hence the smaller kernel is favoured for its low memory requirement and faster speed. The k is set to 10 in the DualRC-L.

Our code can be found at https://code.active.vision.

4.2 Geometric Matching

4.2.1 Training Data

We evaluate our model on three geometric datasets: HPatches, InLoc and Aachen Day-Night. There is no training or testing data split in these three datasets, hence we follow [8] and train our model on MegaDepth [52] dataset. It consists of a large number of internet images about 196 scenes with corresponding sparse 3D point clouds constructed by COLMAP [1], [2]. The camera intrinsics and extrinsics together with the depth maps are also included. We also follow the procedure in [8] to generate sparse ground truth labels. First, we compare the overlap in sparse SfM point cloud between all image pairs to select the pairs whose overlap is over 50%. Next, for all selected pairs, the second image with depth information is projected into the first image and occlusion is removed by depth check. Then, we randomly collect 128 correspondences from each image pair to train our model. We use the scenes with more than 500 valid image pairs for training and the rest scenes for validation. To avoid scene bias, 110 image pairs are randomly selected from each training scene to constitute our training set. In total, we obtain $15,070$ training pairs and $14,638$ validation pairs. All of three models are trained with Adam optimizer [53] for 15 epoch with initial learning rate of 0.001. The learning rate is halved for every 5 epochs. The coefficient to orthogonal loss is set to be 0.05. The images are resized to 400×400 during training.

4.2.2 HPatches

HPatches dataset [29] is designed to evaluate the accuracy of the correspondences under viewpoint and illumination changes. It contains 108 image sequences. 56 of them undergo viewpoint change and 52 of them undergo illumination change. Each sequence consists of one reference image and 5 query images. Each query image pairs with the reference image hence five image pairs are obtained for each sequence. Homography w.r.t. to the reference image is provided for each query image. We evaluate our method with two metrics for this dataset. The first one is Mean Matching Accuracy (MMA). This metric projects the keypoints $\mathbf{p}_i^A$ in the query image to the target image using the ground-truth homography $\mathcal{H}^*$, and calculate the percentage of predicted correspondences $\mathbf{p}_i^B$ in the target image whose distances to the ground-truth lies within a threshold t. Mathematically, it is defined as:

$$\text{MMA}(\{\mathbf{p}_i^A, \mathbf{p}_i^B\}_{i=1}^N; t) = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}(t - \|\mathcal{H}^*(\mathbf{p}_i^A) - \mathbf{p}_i^B\|_2), \tag{19}$$

where t is the threshold of 2D distance. $\mathbf{1}(\cdot)$ is a binary indicator function whose output is 1 for non-negative value and 0 otherwise. $\mathcal{H}^*(\cdot)$ denotes the warping by homography. MMA is calculated for each image pair and is averaged over the entire dataset. The second metric is the corner projection error (CPE) [17] to measure the accuracy of the correspondence by homography estimation. We use OpenCV to calculate homography $\mathcal{H}$ between two images based on the found correspondences. Then four corners $\{c_1, c_2, c_3, c_4\}$ of the query image are projected using both predicted homography $\mathcal{H}$ and ground-truth homography $\mathcal{H}^*$, obtaining $\{\hat{c}_1, \hat{c}_2, \hat{c}_3, \hat{c}_4\}$ and $\{\hat{c}_1^*, \hat{c}_2^*, \hat{c}_3^*, \hat{c}_4^*\}$. The CPE is defined as:

$$\text{CPE} = \frac{1}{4}\sum_{i=1}^{4} \mathbf{1}(\|\epsilon - \|\hat{c}_i - \hat{c}_i^*\|_2\|) \tag{20}$$

where ϵ is a threshold. CPE is calculated individually for each image pair and is then averaged over the entire dataset. The result is reported as the average CPE under thresholds ϵ = 3px, 5px, 7px, 10px.

During the evaluation, we interpolate the longer side of the input image to 1600 and keep the original aspect ratio before passing them through the network. The established correspondences are then rescaled to the original scale. For average MMA metric, we follow the evaluation procedure in [26] that 1000 matches are extracted by all methods for a fair comparison. We also include the results of top 2000 matches between methods related to the 4D neighbourhood consensus. For average CPE (AUC), the homographies are estimated by top 1000 matches for all methods. The results on two metrics are summarized in table 1 and table 2 respectively.

TABLE 1: Average MMA on HPatches Dataset. DualRC outperforms others for $t = \{5\text{px}, 10\text{px}\}$ and performs on par with Sparse-NCNet for $t = 3$px.

Method	Average MMA			
	$t = 3$px	$t = 5$px	$t = 10$px	# matches
D2-Net [8]	44.5	66.3	83.6	1k
R2D2 [9]	73.0	81.1	84.5	1k
S.P.+S.G. [17]	64.6	72, 5	78.2	1k
DELF [16]	47.7	54.8	70.5	1k
Sparse-NCNet [26]	**79.6**	88.0	91.8	1k
DualRC	78.8	**90.3**	**95.9**	1k
DualRC-L	75.9	87.4	93.3	1k
DualRC-D	77.5	89.5	95.7	1k
NCNet [24]	62.6	79.9	90.8	2k
Sparse-NCNet [26]	**78.5**	86.7	90.8	2k
DualRC	**78.5**	**90.1**	**95.7**	2k
DualRC-L	75.3	87.1	93.0	2k
DualRC-D	77.3	89.4	95.6	2k

Our method outperforms the existing works in both MMA and CPE metrics, especially over various neighbourhood consensus methods. The results indicate that the basic NC module and other two variants work very well in producing robust correspondences. The results of DualRC-L are slightly lower than the other two types in average MMA metric, this is not a surprising result since the sparse correlation tensor only stores the top 10 candidates for each feature, which limits the matching area. However, from results on average CPE metric, which focuses on the overall capability of extracted correspondences on the downstream

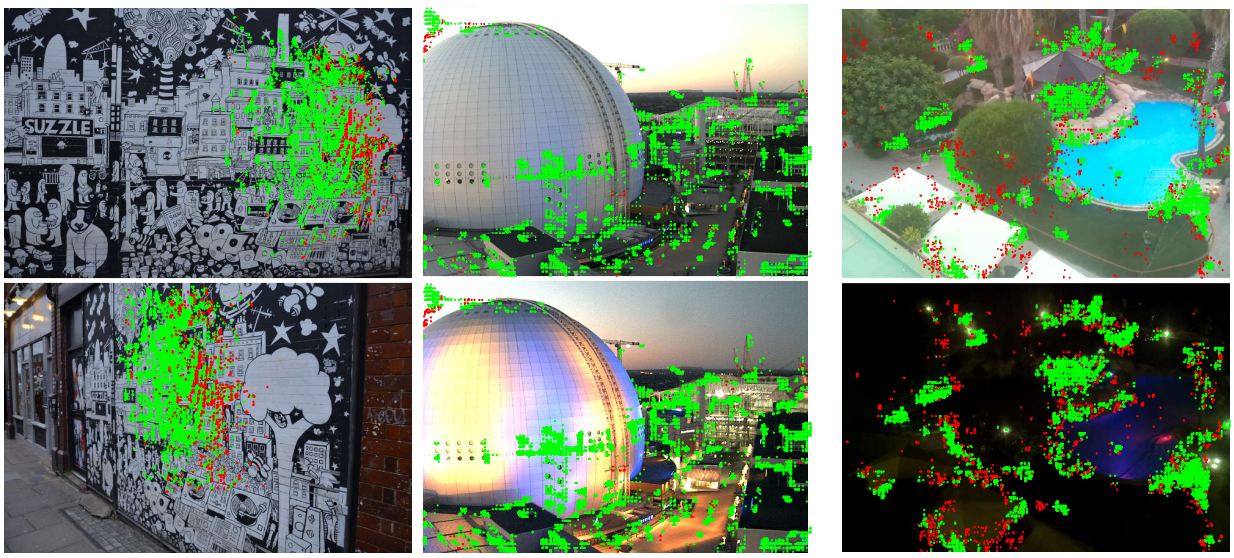


Fig. 5: Qualitative Results on HPatches Dataset. We select the top 6000 matches and set the projection error threshold as 3 pixel. Green parts are correctly localized matches while red parts are incorrect matches.

TABLE 2: Average CPE on HPatches Dataset. DualRC-L has a very closed performance to other two variants, indicating the lightweight NC module being able to achieve efficient filtering without compromising the performance.

Method	Average CPE			
	$\epsilon = 3$px	$\epsilon = 5$px	$\epsilon = 7$px	$\epsilon = 10$px
Sparse-NCNet [26]	49.4	57.2	62.7	68.3
DualRC	55.9	62.6	67.3	72.6
DualRC-L	**56.7**	63.3	68.3	73.4
DualRC-D	56.6	**63.8**	**68.9**	**74.4**

TABLE 3: Run-time Comparison. DualRC-L is almost ten times faster than DualRC but has comparable performance with DualRC. In contrast, Sparse-NCNet requires the input images to have a larger size, which in return slows down the speed of the model.

Method	Input Size px × px	Time second	MMA $t = 5$px
NCNet [24]	1600 × 1200	4.65	79.9
Sparse-NCNet [26]	3200 × 2400	1.64	87.6
Sparse-NCNet [26]	1600 × 1200	**0.37**	82.3
DualRC	1600 × 1200	3.28	**90.3**
DualRC-L	1600 × 1200	0.47	87.4

task, rather than individual accuracy of each correspondence, all of our three types have similar performance, notably outperforming Sparse-NCNet.

We also evaluate the run-time for different NC-based methods and report the results in table 3. All methods are tested on a Nvidia TITAN RTX GPU. We compare the run-time for one single forward pass through the network. The time for one forward pass is reduced from 3.28 seconds to 0.47 second, almost 10 times faster when switching from the basic NC module to lightweight NC module, significantly outperforming Sparse-NCNet with the same input resolution and marginally being outperformed by DualRC. To maintain good performance, Sparse-NCNet requires the input images to have a larger size, which in return slows down the speed of the model. For the DualRC-D model, the model is too large to be fit into a single GPU under the image resolution of 1600×1200, hence we evaluated it on CPU. Thus it is unfair to include it in this comparison.

4.2.3 InLoc

InLoc dataset [4] is designed to test the performance on long-term indoor relocalization. It consists of a large collection of images with depth acquired by the 3D scanner. Query images are taken by a smartphone camera a few months later. The dataset contains very challenging viewpoint and illumination changes. We follow the evaluation protocol in [4]. For each query image, the dataset provides a shortlist of 10 candidate database images. Our method is then used to establish the correspondence between the query image and the shortlisted image. Finally the PnP solver is used to estimate the pose of the query image. Similar to the evaluation on the HPatches dataset, we interpolate the longer side of the input images to 1600 and keep the same aspect ratio during the evaluation, and scale the correspondences back to original scale afterward. The accuracy of the relocalization is reported in the form of the percentage of the relocalized poses whose distance errors are within τ meters and angular errors are within 10°. The results are shown in table 4. We achieve the best results among methods related to 4D neighbourhood consensus, indicating the superiority of our architecture. Several qualitative results can be found in fig. 6.

4.2.4 Aachen Day Night

Aachen Day-Night dataset [30] is designed to evaluate performance on outdoor relocalization under day night illumination change. It contains 4479 day time images of the Aachen city, with 98 night query images. We use the official evaluation script provided by [30]. A shortlist of image pairs is provided and our method extracts correspondences

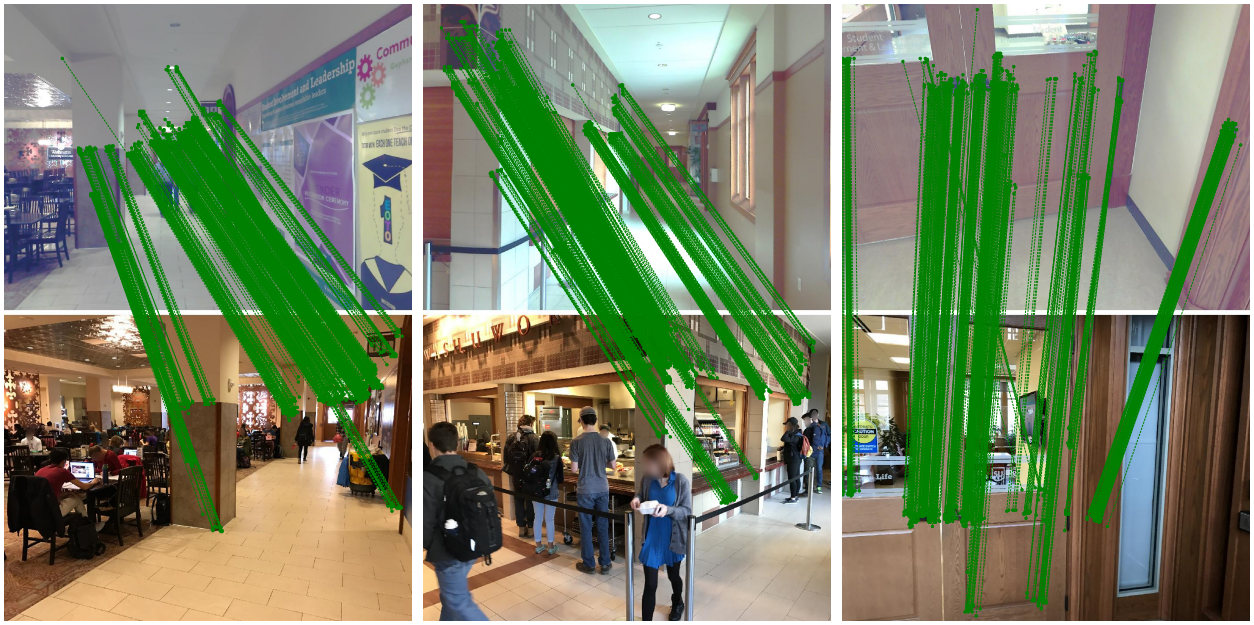


Fig. 6: Qualitative Results on InLoc Dataset. Top 500 matches are drawn.

TABLE 4: Results on InLoc Dataset. Our DualRC achieves the best results in distance error of 0.5m, 1m and 2m and is slightly outperformed by SparseNC-Net in 0.25.

Method	Distance Error			
	$\tau = 0.25$m	$\tau = 0.5$m	$\tau = 1$m	$\tau = 2$m
D2-Net [24]	43.2	61.1	74.2	-
NCNet [24]	44.1	63.8	76.0	78.4
Sparse-NCNet [26]	**45.4**	66.2	79.9	82.1
DualRC	44.1	**67.5**	**82.4**	**84.8**
DualRC-L	44.1	63.2	79.9	81.5

from the pairs. Then a sparse 3D pointcloud of the scene is constructed using COLMAP [1], [2]. The poses of the query images are estimated based on reconstructed pointcloud. The accuracy is measured in the percentage of relocalized pose within the distance error τ meters and the angular error $\theta°$. The results are presented in table 5. Our DualRC model achieves the best result in thresholds (0.5m, 2°) and (5m, 10°) among the existing works. Qualitative results are shown in fig. 7.

TABLE 5: Results on Aachen Day Night Dataset. Our DualRC achieves the best results in threshold (0.5m, 2°) and (5m, 10°).

Method	(Distance Error, Angular Error)		
	(0.25m, 2°)	(0.5m, 5°)	(5m, 10°)
SP+SG [54]	**79.6**	**90.8**	**100**
ASLFeat+OANet [55]	77.6	89.8	**100**
Sparse-NCNet [26]	76.5	84.7	98.0
DualRC	**79.6**	88.8	**100**
DualRC-L	77.6	87.8	**100**

4.3 Semantic Matching

We evaluate our model on three semantic matching datasets: PF-PASCAL, PF-WILLOW, and SPair-71k datasets.

PF-PASCAL and PF-WILLOW are parts of Proposal Flow dataset [31]. PF-PASCAL contains 1351 images from 20 categories, while PF-WILLOW has 100 images equally distributed into 10 categories. Both datasets have sparse keypoint annotation. We follow the standard evaluation procedure [18], [20], [24], [25], [56] and generates 2940 training pairs, 300 validation pairs and 300 testing pairs for PF-PASCAL, as well as 900 testing pairs for PF-WILLOW. Since PF-WILLOW does not have the training data, the evaluation on both datasets uses the model trained on the PF-PASCAL dataset. The model is trained with Adam optimizer for 20 epochs with learning rate of 0.001 and orthogonal loss weight is set to 0.005. The input image is resized to 256×256 for both training and testing on both PF-PASCAL and PF-WILLOW datasets.

SPair-71k is a far more challenging dataset. It consists of 53340 training image pairs, 5384 validation pairs and 12234 testing pairs, spanning 18 categories with provided keypoint annotations object bounding boxes. Compared with PF-PASCAL, SPair-71k contains much larger viewpoint and scale variations between images, making correspondence localization very hard. We evaluate two types of model on SPair-71k: one is fine-tuned on the training data of SPair-71k and the other is the transferred model trained on PF-PASCAL. The model is trained with Adam optimizer with the learning rate of 0.001 for 10 epochs with orthogonal loss weight is set as 0.005. The model is trained and tested with an image size of 400×400.

The evaluation metric adopted for both datasets is the standard percentage of corrected keypoints (PCK). Given a set of prediction and groundtruth keypoints $\mathcal{K} = \{(k_i, k_i^*)\}_{i=1}^N$, it measures the percentage of predictions whose distances to the ground-truth values lies within a certain threshold value. Mathematically, it is defined as:

$$PCK(\mathcal{K}) = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\big(\|k_i - k_i^*\|_2 \leq \alpha \cdot \max(w, h)\big) \quad (21)$$

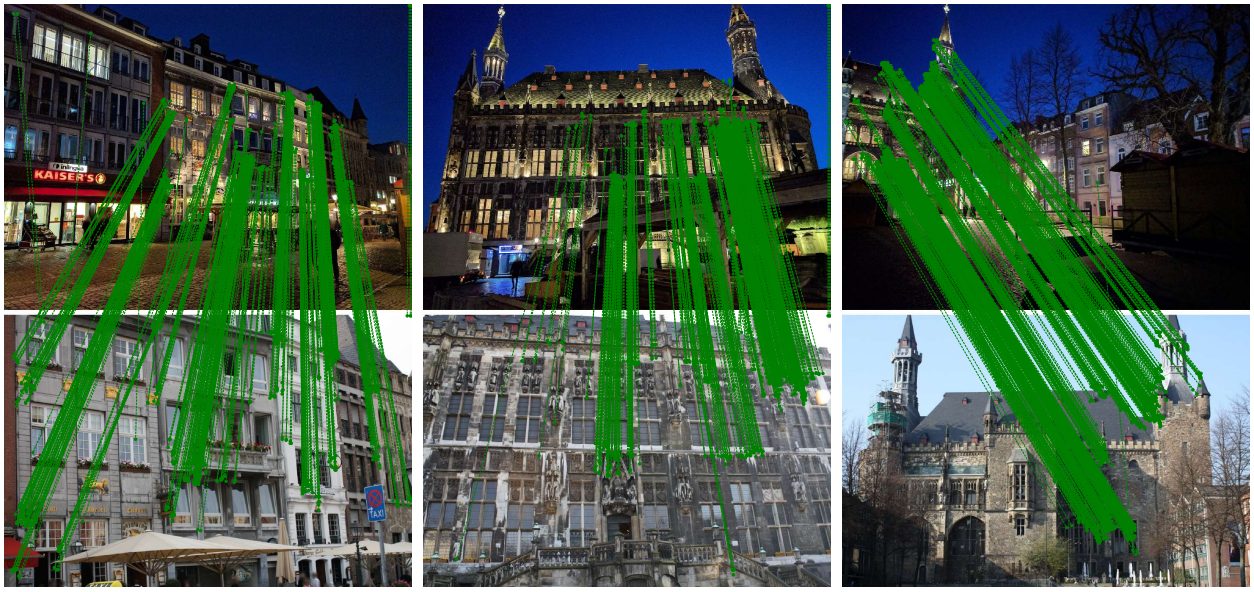


Fig. 7: Qualitative Results on Aachen Day Night Dataset. Top 500 matches are drawn. Note that our model can achieve good matching under significant illumination and viewpoint change.

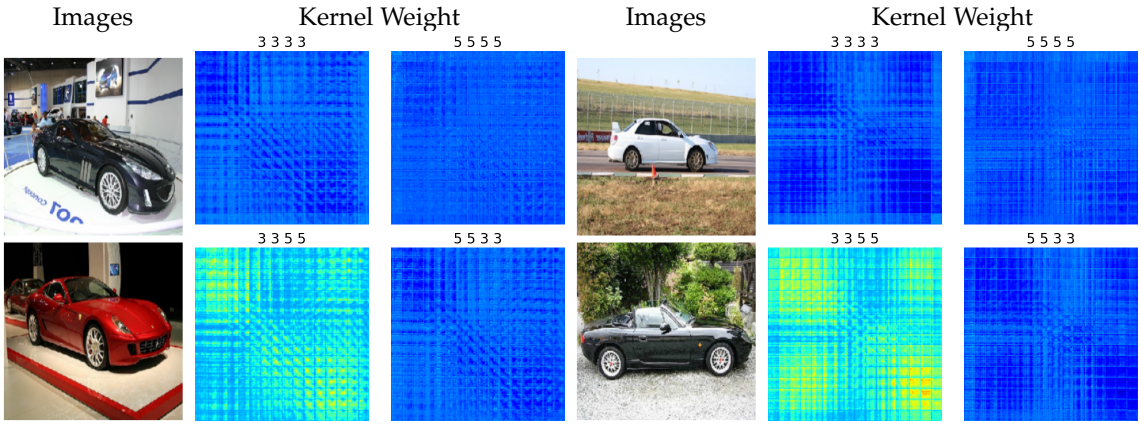


Fig. 8: Visualization of learned 4D Kernels. The weights of the input layer are drawn. It is clear that the non-isotropic kernel $3 \times 3 \times 5 \times 5$ have a greater response when scale variation between two images is greater.

where (w, h) are the width and height of either the input image or the bounding box of the object. In our case, PF-PASCAL and PF-WILLOW use the image and SPair-71k uses the bounding box.

We summarize the results in table 6. The DualRC-D model achieves the best results in SPair-71k datasets and the second best result in PF-PASCAL dataset, when compared with existing methods. We can see that there is a clear boost in accuracy across all datasets when switching from the basic NC module (DualRC) to the DyANC module, which demonstrates the effectiveness of DyANC module. We find that, for the same object category, the keypoints in PF-WILLOW are annotated slightly differently from the other two datasets. Meanwhile, PF-WILLOW only contains 3 categories, while PF-PASCAL and Spair-71K contain around 20 categories. Thus, there exist distribution shifts between PF-WILLOW and other datasets. Therefore, we observe that methods performing well on PF-WILLOW do not perform as well on PF-PASCAL and SPair-71k. Since PF-PASCAL and SPair-71k are much larger and more challenging than PF-WILLOW, the performance of PF-PASCAL and SPair-71k is a stronger indicator. We have also evaluated DualRC-L and Sparse-NCNet on the semantic tasks, however, the results are unsatisfactory. We believe it is due to the fact that the semantic matching requires a greater area of neighbourhood for high-level semantic understanding. The lightweight NC module in DualRC-L model only stores top 10 matching candidates for each feature and hence the neighbourhood is very sparse with little semantic information. This consequently leads to the poor performance in semantic tasks.

To verify whether our DyANC module is able to learn to choose proper neighbourhood sizes, we visualize weights of 4D kernel components and demonstrate them in fig. 8. In these two examples, the first pair have small scale variation and the second one have significant scale variation. From the weight learned by the input layer, it is clear that the non-isotropic kernel $3 \times 3 \times 5 \times 5$ of the pair with significant

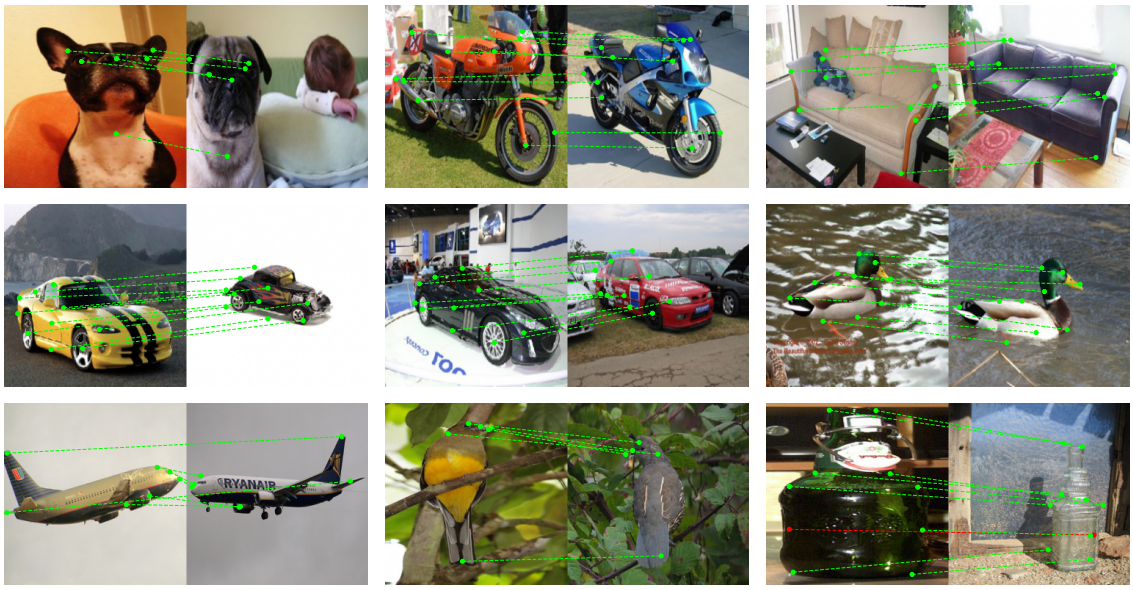


Fig. 9: Qualitative Results on Semantic Matching Datasets. From top to bottom: PF-PASCAL, PF-WILLOW (middle), SPair-71k.

scale variation has a greater response than that of the pair with less scale variation. This clearly indicates that DyANC module successfully captures the scale variation between objects and automatically select the suitable neighbourhood region. Qualitative results on three semantic datasets are presented in fig. 9.

TABLE 6: Results for Semantic Matching. Our DualRC-D model achieves the best results in both transferred and fine-tuned experiments on SPair-71k datasets and the second best result on PF-PASCAL dataset. Note that there is a clear boost in accuracy when switching from DualRC to DualRC-D, indicating the effectiveness of DyANC module.

	PF-PASCAL PCK($\alpha = 0.1$)	PF-WILLOW PCK($\alpha = 0.1$)	SPair-71k ($\alpha = 0.1$) (Fine-tuned)	(Transferred)
UCN [56]	75.1	-	-	17.7
NCNet [24]	78.9	67.0	-	20.1
SCOT [19]	85.6	**76.0**	35.6	-
DCC-Net [28]	82.3	73.8	-	26.7
ANC-Net [25]	86.1	-	-	<u>28.7</u>
HPF [32]	84.8	<u>74.4</u>	28.2	-
DHPF [57]	**90.7**	71.0	37.3	27.4
DualRC	85.8	56.3	<u>42.0</u>	26.4
DualRC-D	<u>86.7</u>	60.0	**42.9**	**29.4**

4.4 More discussion and comparison

Correspondence estimation is a fundamental problem and the field advances very fast. During the preparation of this paper, a few new works appear and show better performance than our DualRC in either the geometric or semantic correspondence estimation task. Particularly, for the geometric task, LoFTR [46], a transformer based method sharing a similar dual-resolution design as our DualRC, achieves the current state-of-the-art performance. Meanwhile, for the semantic task, CHM [20], which introduces 6D convolution kernels to construct deep Hough matching networks, and CATs [58] and its followup [59], which introduce cost aggregation transformers for matching, achieve the state-of-the-art performance. Our DualRC is designed for general visual correspondence estimation including both geometric and semantic tasks, while the aforementioned methods are developed with a specific focus either on geometric or semantic task. Thus, to have a full picture of how these methods work on the general visual correspondence estimation problem, we retrain LoFTR on FP-PASCAL to measure its performance on the semantic task, and CHM and CATs on MegaDepth to measure their performance on the geometric task. The experimental details are provided in the supplementary material. We found that DualRC notably outperforms LoFTR on the semantic task (PCK w/ $\alpha = 0.1$ on PF-PASCAL: 85.8 vs 63.1). CHM and CATs struggle on the geometric task and are significantly outperformed by DualRC (table 2 in the supp.), partly due to their huge memory consumption, prohibiting the models to process images of a large resolution (e.g., 1600×1600), which is critical for geometric matching. Our method outperforms CHM and CATs on all resolutions that can be handled under the same hardware. For LoFTR, we further evaluate it on the PhotoTourism dataset of Image Matching Challenge [60] and our method achieves overall on-par performance with the LoFTR when both methods are trained under the same hardware resource (table 1 in the supp.). In contrast to these recent advances, which show state-of-the-art performance on either of the two tasks, our DualRC consistently achieves promising results on both tasks. Detailed comparison on performance and memory consumption can be found in the supplementary material.

5 Conclusion

In this paper, we propose a dense matching model which can be easily adapted to both semantic and geometric matching. We propose a model named DualRC that consists of a dual-resolution structure to establish correspondences in a coarse-to-fine manner. The coarse-resolution feature

maps form a coarse-correlation tensor which is filtered by the neighbourhood consensus (NC) module. Besides, we propose two NC variants to tackle the specific demands for geometric and semantic matching respectively. To accelerate the model in the presence of high-resolution input images in geometric matching, we present DualRC-L which has a lightweight NC module based on sparse correlation [26] and convolution to run efficient filtering. For semantic matching where scale variation is predominant, we propose DualRC-D with the novel DyANC module to select the best neighbourhood region during filtering. The output of the filtering is then used to regularize the matching using fine-resolution feature maps. Our model is evaluated on both semantic matching datasets as well as geometric datasets. In all cases, our method achieves promising results, outperforming all other NC based methods. Although dense matching avoid missing-detection problem, it is slow when applied to large image pairs since all pixels are needed to be treated as queries. Naively prepending a keypoint detector instead may results in the incompatibility between keypoint detector and dense matcher, which degrades the performance. Therefore, one possible future direction is to combining keypoint detection and matching module, leveraging the advantages from both sides.

ACKNOWLEDGMENTS

This work is partially supported by Hong Kong Research Grant Council - Early Career Scheme (Grant No. 27208022), National Natural Science Foundation of China (Grant No. 62306251), and HKU Seed Fund for Basic Research.

REFERENCES

[1] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of IEEE Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[2] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, "Pixelwise view selection for unstructured multi-view stereo," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[3] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *Proceedings of IEEE Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[4] H. Taira, M. Okutomi, T. Sattler, M. Cimpoi, M. Pollefeys, J. Sivic, T. Pajdla, and A. Torii, "InLoc: Indoor visual localization with dense matching and view synthesis," in *Proceedings of IEEE Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[5] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Intl. Journal of Computer Vision (IJCV)*, 2004.

[6] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2006.

[7] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *Proceedings of Intl. Conf. on Computer Vision (ICCV)*, 2011.

[8] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler, "D2-net: A trainable cnn for joint detection and description of local features," in *Proceedings of IEEE Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[9] J. Revaud, P. Weinzaepfel, C. De Souza, N. Pion, G. Csurka, Y. Cabon, and M. Humenberger, "R2D2: Repeatable and Reliable Detector and Descriptor," in *Proceedings of Conf. on Neural Information Processing Systems (NeurIPS)*, 2019.

[10] O. Wiles, S. Ehrhardt, and A. Zisserman, "Co-attention for conditioned image matching," in *Proceedings of IEEE Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[11] Y. Tian, B. Fan, and F. Wu, "L2-net: Deep learning of discriminative patch descriptor in euclidean space," in *Proceedings of IEEE Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[12] Y. Ono, E. Trulls, P. Fua, and K. M. Yi, "Lf-net: learning local features from images," in *Proceedings of Conf. on Neural Information Processing Systems (NeurIPS)*, 2018.

[13] Y. Tian, X. Yu, B. Fan, F. Wu, H. Heijnen, and V. Balntas, "Sosnet: Second order similarity regularization for local descriptor learning," in *Proceedings of IEEE Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[14] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas, "Working hard to know your neighbor's margins: Local descriptor learning loss," in *Proceedings of Conf. on Neural Information Processing Systems (NeurIPS)*, 2017.

[15] Z. Luo, L. Zhou, X. Bai, H. Chen, J. Zhang, Y. Yao, S. Li, T. Fang, and L. Quan, "Aslfeat: Learning local features of accurate shape and localization," in *Proceedings of IEEE Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[16] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han, "Large-scale image retrieval with attentive deep local features," in *Proceedings of Intl. Conf. on Computer Vision (ICCV)*, 2017.

[17] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proceedings of IEEE Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[18] K. Han, R. S. Rezende, B. Ham, K.-Y. K. Wong, M. Cho, C. Schmid, and J. Ponce, "Scnet: Learning semantic correspondence," in *Proceedings of Intl. Conf. on Computer Vision (ICCV)*, 2017.

[19] Y. Liu, L. Zhu, M. Yamada, and Y. Yang, "Semantic correspondence as an optimal transport problem," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4463–4472.

[20] J. Min and M. Cho, "Convolutional hough matching networks," in *Proceedings of IEEE Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[21] P. H. Seo, J. Lee, D. Jung, B. Han, and M. Cho, "Attentive semantic alignment with offset-aware correlation kernels," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 349–364.

[22] I. Rocco, R. Arandjelovic, and J. Sivic, "Convolutional neural network architecture for geometric matching," in *Proceedings of IEEE Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[23] I. Rocco, R. Arandjelović, and J. Sivic, "End-to-end weakly-supervised semantic alignment," in *Proceedings of IEEE Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[24] I. Rocco, M. Cimpoi, R. Arandjelović, A. Torii, T. Pajdla, and J. Sivic, "Neighbourhood consensus networks," in *Proceedings of Conf. on Neural Information Processing Systems (NeurIPS)*, 2018.

[25] S. Li, K. Han, T. W. Costain, H. Howard-Jenkins, and V. Prisacariu, "Correspondence networks with adaptive neighbourhood consensus," in *Proceedings of IEEE Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[26] I. Rocco, R. Arandjelović, and J. Sivic, "Efficient neighbourhood consensus networks via submanifold sparse convolutions," in *arXiv preprint*, 2020.

[27] P. Truong, M. Danelljan, and R. Timofte, "GLU-Net: Global-local universal network for dense flow and correspondences," in *Proceedings of IEEE Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[28] S. Huang, Q. Wang, S. Zhang, S. Yan, and X. He, "Dynamic context correspondence network for semantic alignment," in *Proceedings of Intl. Conf. on Computer Vision (ICCV)*, 2019.

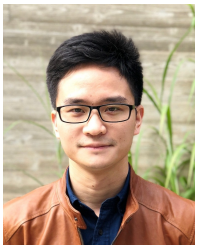
[29] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk, "Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors," in *Proceedings of IEEE Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[30] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, F. Kahl, and T. Pajdla, "Benchmarking 6dof outdoor visual localization in changing conditions," in *Proceedings of IEEE Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[31] B. Ham, M. Cho, C. Schmid, and J. Ponce, "Proposal flow: Semantic correspondences from object proposals," *IEEE Trans. Pattern Anal. Machine Intell. (PAMI)*, 2017.

[32] J. Min, J. Lee, J. Ponce, and M. Cho, "Hyperpixel flow: Semantic correspondence with multi-layer neural features," in *Proceedings of Intl. Conf. on Computer Vision (ICCV)*, 2019.
[33] X. Li, K. Han, S. Li, and V. Prisacariu, "Dual-resolution correspondence networks," in *Proceedings of Conf. on Neural Information Processing Systems (NeurIPS)*, 2020.
[34] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Proceedings of Intl. Conf. on Computer Vision (ICCV)*, 2011.
[35] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010.
[36] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer, "Discriminative learning of deep convolutional feature point descriptors," in *Proceedings of Intl. Conf. on Computer Vision (ICCV)*, 2015.
[37] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of Intl. Conf. on Learning Representations (ICLR)*, 2015.
[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of IEEE Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.
[39] H. Germain, G. Bourmaud, and V. Lepetit, "S2dnet: Learning accurate correspondences for sparse-to-dense feature matching," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.
[40] Q. Wang, X. Zhou, B. Hariharan, and N. Snavely, "Learning feature descriptors using camera pose supervision," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.
[41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of Conf. on Neural Information Processing Systems (NeurIPS)*, 2017.
[42] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *ICLR*, 2021.
[43] H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, S. Ma, C. Xu, C. Xu, and W. Gao, "Pre-trained image processing transformer," in *Proceedings of IEEE Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.
[44] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.
[45] W. Jiang, E. Trulls, J. Hosang, A. Tagliasacchi, and K. M. Yi, "COTR: Correspondence Transformer for Matching Across Images," in *Proceedings of Intl. Conf. on Computer Vision (ICCV)*, 2021.
[46] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, "LoFTR: Detector-free local feature matching with transformers," *Proceedings of IEEE Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.
[47] G. Tinchev, S. Li, K. Han, D. Mitchell, and R. Kouskouridas, "𝕏resolution correspondence networks," in *British Machine Vision Conference (BMVC)*, 2021.
[48] T.-y. Lin, P. Doll, R. Girshick, K. He, B. Hariharan, S. Belongie, and F. Ai, "Feature pyramid networks for object detection," in *Proceedings of IEEE Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.
[49] C. Choy, J. Gwak, and S. Savarese, "4d spatio-temporal convnets: Minkowski convolutional neural networks," in *Proceedings of IEEE Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.
[50] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Proceedings of Conf. on Neural Information Processing Systems (NeurIPS)*, 2019.
[51] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of IEEE Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2009.
[52] Z. Li and N. Snavely, "Megadepth: Learning single-view depth prediction from internet photos," in *Proceedings of IEEE Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.
[53] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of Intl. Conf. on Learning Representations (ICLR)*, 2015.
[54] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *Proceedings of IEEE Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.
[55] J. Zhang, D. Sun, Z. Luo, A. Yao, L. Zhou, T. Shen, Y. Chen, H. Liao, and L. Quan, "Learning two-view correspondences and geometry using order-aware network," in *Proceedings of Intl. Conf. on Computer Vision (ICCV)*, 2019.
[56] C. B. Choy, J. Gwak, S. Savarese, and M. Chandraker, "Universal correspondence network," *arXiv preprint arXiv:1606.03558*, 2016.
[57] J. Min, J. Lee, J. Ponce, and M. Cho, "Learning to compose hypercolumns for visual correspondence," in *ECCV*, 2020.
[58] S. Cho, S. Hong, S. Jeon, Y. Lee, K. Sohn, and S. Kim, "Cats: Cost aggregation transformers for visual correspondence," in *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.
[59] S. Hong, S. Cho, J. Nam, and S. Kim, "Cost aggregation is all you need for few-shot segmentation," *arXiv preprint arXiv:2112.11685*, 2021.
[60] "Image matching challenge," https://www.cs.ubc.ca/research/image-matching-challenge/.

Xinghui Li received Master of Engineering degree (a 4-year program) with first class honour from the University of Oxford in 2019. He is currently a Ph.D. student at Active Vision Lab (AVL) of the University of Oxford under the supervision of Professor Victor Prisacariu. His research interests are image matching, visual localization and multi-modal learning.



Kai Han received his Ph.D. in Computer Science from The University of Hong Kong in 2018. He has since held positions as a Postdoctoral Researcher at the Visual Geometry Group (VGG) at the University of Oxford, a Lecturer in Computer Vision at the University of Bristol, and a Visiting Faculty Researcher at Google Research. He is currently an Assistant Professor at the Department of Statistics and Actuarial Science at The University of Hong Kong.

Shuda Li is currently a Research Scientist at Common Sense Machine Inc. He received his Ph.D. in Computer Science from the University of Bristol, UK, in 2016 and worked as a Postdoctoral Research Assistant in the Active Vision Lab (AVL) at the University of Oxford between 2017 and 2020. His research interests include SLAM, Lie and Multi-view Geometry, Real2Sim2Real, and Spatial AI.

Victor Prisacariu received the Graduate degree (with first class hons.) in computer engineering from Gheorghe Asachi Technical University, Iasi, Romania, in 2008, and the Ph.D. degree in engineering science from the University of Oxford in 2012. He is currently an Associate Professor in Information Engineering with the Department of Engineering Science at the University of Oxford, leading the Active Vision Lab (AVL). He is also the Chief Scientist of Niantic Inc.