DFM: A Performance Baseline for Deep Feature Matching

Ufuk Efe, Kutalmis Gokalp Ince, A. Aydin Alatan
Department of Electrical and Electronics Engineering, Center for Image Analysis (OGAM)
Middle East Technical University, Ankara, Turkey
ufuk.efe, kutalmis, alatan @ metu.edu.tr

Abstract

A novel image matching method is proposed that utilizes learned features extracted by an off-the-shelf deep neural network to obtain a promising performance. The proposed method uses pre-trained VGG architecture as a feature extractor and does not require any additional training specific to improve matching. Inspired by well-established concepts in the psychology area, such as the Mental Rotation paradigm, an initial warping is performed as a result of a preliminary geometric transformation estimate. These estimates are simply based on dense matching of nearest neighbors at the terminal layer of VGG network outputs of the images to be matched. After this initial alignment, the same approach is repeated again between reference and aligned images in a hierarchical manner to reach a good localization and matching performance. Our algorithm achieves 0.57 and 0.80 overall scores in terms of Mean Matching Accuracy (MMA) for 1 pixel and 2 pixels thresholds respectively on Hpatches dataset [4], which indicates a better performance than the state-of-the-art.

1. Introduction

Determining point correspondences between images is one of the vital and well-studied topics in computer vision. These correspondences are crucial for several applications, such as Simultaneous Localization and Mapping (SLAM), Structure-from-Motion (SfM), pose estimation, image retrieval, and image matching.

Classical image matching pipeline consists of *Feature Detection, Feature Description, Feature Matching,* and *Geometric Transformation Estimation* steps. Most of the popular algorithms before the deep learning era have focused on one or more phases in this pipeline. While many techniques [24, 61, 51, 35] concentrate on Feature Detection, some other approaches [18, 37, 13] handle Feature Description step locally. The most popular algorithms [33, 6, 52, 30, 1, 2] work on both Feature Detection and Description. Feature Matching is usually achieved by minimizing/maximizing some well-defined metrics, such as Sum of Squared Differences (SSD) or correlation, which is next accompanied by outlier rejection methods, such as ratio test of SIFT [33] or some outlier rejection algorithms, such as LPM [34]. In Geometric Transformation Estimation stage, either epipolar geometry or homography is estimated by DLT [25] or RANSAC [23] based algorithms [14, 65, 15, 16] commonly.

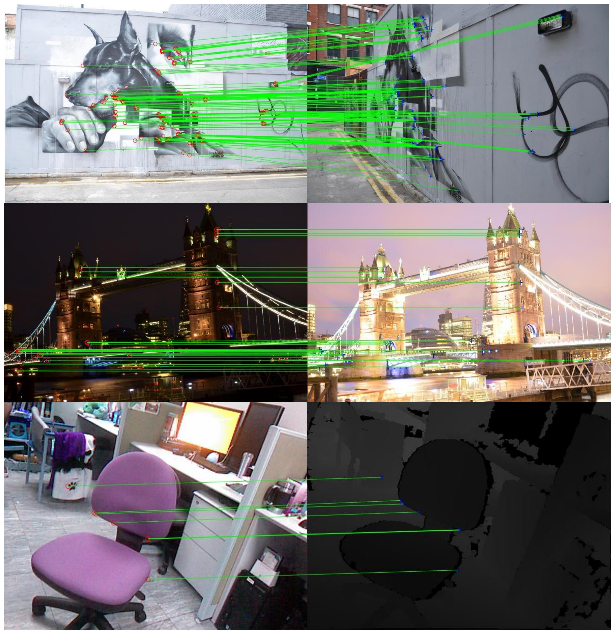

Figure 1: **Visual results of the proposed method.** Proposed DFM can find high-quality matches under severe viewpoint (1^{st} row) [4] and illumination changes (2^{nd} row) [4]. It is also successful on cross-domain problems, such as RGB to depth image matching (3^{rd} row) [68].

During recent years, image matching met deep learning, and a number of valuable efforts have been performed. However, most of these studies still rely on the classical pipeline. In this respect, while many methods

This research is funded by ROKETSAN.

[67, 29, 54, 20, 39, 5] try to improve Feature Detection with deep architectures, some others [71, 58, 3, 38, 64] reconstruct Feature Descriptors by utilizing learned representations. On the other hand, there are also techniques [21, 41, 56, 46, 22, 69] that tackle both problems together and jointly try to solve them. Some efforts, as in [70, 72], train the network to learn how to reject outliers in the presence of putative matches. SuperGlue [53] learns to make appropriate matches using *matching descriptors* which are obtained by benefiting both original descriptors and keypoint locations via multilayer perceptron. Another approach [7] uses reinforcement learning to optimize feature detection and description for high-level tasks like relative pose estimation. Some recent approaches [12, 10, 11] try to find differentiable alternatives to RANSAC in order to estimate geometric transformation better. Note that all the aforementioned efforts provide some improvements on some parts of the classical pipeline.

The performance of the overall conventional system is dictated by the weakest sub-block in this serial pipeline during image matching. Therefore, trying to enhance the performance of each stage would be sub-optimal as the output of each block is the input to another, and every single step is prone to error in a different scenario. Hence, it can be observed that some recent works in the literature have abandoned the classical pipeline and started to propose combining many stages together in their solutions to overcome this bottleneck. In this manner, there are methods [19, 40, 48] that handle image matching problem with a single architecture by directly estimating the geometric transformation between two images with regression. More recently, some efforts [50, 49, 73] find correspondences between two images by benefiting from deep features with a single network instead of separately detect, describe and match them.

The representations that are obtained by off-the-shelf feature extractors are shown to be quite effective for complex tasks, such as object detection [45, 9, 31] and semantic segmentation [26]; however, as far as we know, currently the only method employing off-the-shelf feature extractors is D2-Net [22]. In other words, the requirement of designing new deep networks for image matching is not clearly justified by demonstrating the insufficiency of off-the-shelf feature extractors as they are not examined in detail. On the contrary, it is firmly believed that the human-level performance of these feature extractors for image classification task [27] indicates their semantic abstraction capability. Moreover, their success in object detection and semantic segmentation tasks reveals their good localization capability. If these pre-trained extractors are utilized in a way for exploiting both their semantic abstraction and good localization capabilities, they might have a chance to achieve satisfactory performance for image matching task as well. If these pre-trained extractors are utilized in such a way that exploits both their semantic abstraction and good localization capabilities, they might have a chance to achieve satisfactory performance for image matching task as well.

Recognition-by-components theory [8] advocates that our way of perception of objects are separating them into parts, which supports that humans might accomplish image matching task using objects as they are the main components of images. We have also been inspired by the Mental Rotation [57] which states that the human brain does some initial rotations while looking for similarities between two objects.

Our main motivation in this paper is to reveal the performance of a pre-trained off-the-shelf feature extractor on image matching problem by applying simple, but effective vision techniques, such as hierarchical search, coarse-to-fine strategy or preliminary alignment, as human visual system possibly employs.

We propose a method that is able to find high-quality correspondences between images, which does not require any training. The proposed method only needs a pre-trained network (e.g. VGG [59] or ResNet [28]) to extract its features. We first leverage the invariances of the deepest layers to find matches between images at a more semantic level. Then we obtain a rough estimate of the geometric transformation between the images to apply the initial warping, as in Mental Rotation [57]. After warping one of the images, we perform hierarchical matching, also a well-known technique, from the deepest level of the feature map to the shallowest layer. With this coarse to fine strategy, we basically exploit the properties of all layers together, such as the powerful semantic property of the deep layers and precise localization of the shallow layers.

Hence, in this study our contribution is threefold;

i. We show that a simple dense nearest neighbor search of the deep features at the terminal layer of a pre-trained network (such as a frozen VGG) still yields sufficient matches for obtaining a geometric transformation- and illumination-invariant matching.

ii. We also demonstrate that after making a rough alignment through geometric transformation and warping in a preliminary step, the image matching performance of a subsequent block that mimics the previous stage in a hierarchical manner to refine positions of dense matches from the terminal layers gives state-of-the-art performance.

iii. Using off-the-shelf deep feature extractors with no training and only employing standard techniques, the proposed method forms a performance baseline for learning-based image matching methods.

2. Related Work

Classical Image Matching Pipeline consists of feature detection, feature description, feature matching, and geometric transformation estimation stages, as mentioned before. Most of the recently developed methods propose some improvements on individual stages of this pipeline. A prominent algorithm, namely SuperPoint [21], which focuses on feature detection and description, utilizes MagicPoint [20] detector, which is trained on a synthetic dataset and uses Homographic Adaptation as a self-supervision strategy to label MS-COCO [32] dataset. Next, this detector-descriptor network is trained jointly by partly sharing parameters. This kind of combined training certainly improves the overall robustness. However, SuperPoint has limitations under some challenges, such as in-plane rotations. The authors in [43] show that due to the trade-off between invariance and discrimination, we can not enforce the network to be more rotation and illumination invariant. Another promising solution, SuperGlue [53] gets the detected features and their descriptors as inputs, then uses a graph neural network to find out cross and self attentions between features. Afterwards, this method learns optimally matching the features by the help of the differentiable Sinkhorn algorithm [60, 17]. SuperGlue argues that it could remarkably improve the matching performances of extracted features. However, its performance depends on the quality of feature detectors and descriptors in principle. Moreover, when the detector and descriptor couple change, it should be retrained to get the best performance. D2-Net [22], on the other hand, uses feature maps of VGG-16 to find salient points and their descriptors. Although D2-Net can work with no training, the authors argue that with well-defined losses, it can be trained and might perform better. Since D2-Net uses only the specific layers of the feature extractor network, it can not utilize all the information encoded in the feature extractor. Moreover, it tends to poorly localize the features on account of using features at a low resolution.

Integrated Approaches are also applied to the image matching problem recently. In integrated approaches, the algorithms take two images as input and directly compute the correspondences between them without externally detect, describe, and match features. As a typical example, NCNet [50] takes advantage of dense CNN features extracted by ResNet-101. Next, NCNet generates the 4D correlation map, which will be processed later by a 4D CNN, namely neighbor consensus network. Finally, the algorithm outputs filtered matches between two images. NCNet algorithm is a novel idea and it works well specifically in semantic matching scenarios; however, the localization performance is not good, since NCNet works in low resolutions due to the large memory requirement. As an alternative integrated approach, Patch2Pix [73] uses NCNet as a baseline and, starting from local patches, it predicts pixel-level matches by using epipolar loss. The algorithm first extracts features by adapted ResNet-34 until the 4^{th} convolutional layer, detects match proposals using the last layer and finally, refines these proposals by using mid and fine-level regressors. Yet, this approach does not entirely benefit from hierarchical refinement and coarse warping stages, which are well-known procedures in vision research. Moreover, it does not use the very last layers of ResNet-34 in which more semantic information is encoded, and it requires training when the feature extractor backbone change. In their ablation study, Patch2Pix clearly shows that starting with the last layer and using the information of all layers before the last layer provides a better performance compared to using some previous layers or using all layers containing the last one. These experimental results also indicate the possible advantages of hierarchical refinement that is used in our proposed method.

3. Method

We propose a two-stage architecture to perform matching between two images, as shown in Figure 2. Our key idea is to extract features with a pre-trained network, aligning images with rough geometric transformation estimation then applying a coarse to fine strategy for better localization. Opposing to the classical feature matching pipeline, rather than constraining the set of feature points by a detection step, we exploit the feature points which are successfully matched like [50, 73]. In Stage-0, we perform *Dense Nearest Neighbor Search* (DNNS) in low spatial resolution to obtain a rough estimation of the geometric transformation between given images. Using this estimate, we warp the second image to form the image that we use next. In Stage-1, we first perform DNNS using the feature maps of the reference image and the warped image at the last layers. Starting with these coarse matches, we refine the matches hierarchically by moving to finer resolutions at each step.

Given an image pair, A and B, we first perform deep feature extraction for both images using a pre-trained VGG-19 network. Then, we continue with a) Dense Nearest Neighbor Search and b) Hierarchical Refinement, which will be explained next in detail. After explaining these two basic steps, we explain the two-stage architecture and how these basic steps are employed.

3.1. Dense Nearest Neighbor Search

Dense Nearest Neighbor Search (DNNS) searches for matches in dense feature maps using a mutual nearest neighbor search with a ratio test [33] as employed in [22]. Given feature patches F^A and F^B extracted from images A and B, DNNS searches F^B to find the best match for each element of F^A. Potential matches are defined as the nearest neighbors in terms of *l2* distance. For a point p^A in feature map F^A, if the ratio of the distance to best match p^B and the second-best match is below the given threshold, then the point p^A is matched to p^B. However, the pair is accepted only if the match is mutual, i.e., p^A and p^B are returned as a matched pair if p^B is matched to p^A as well.

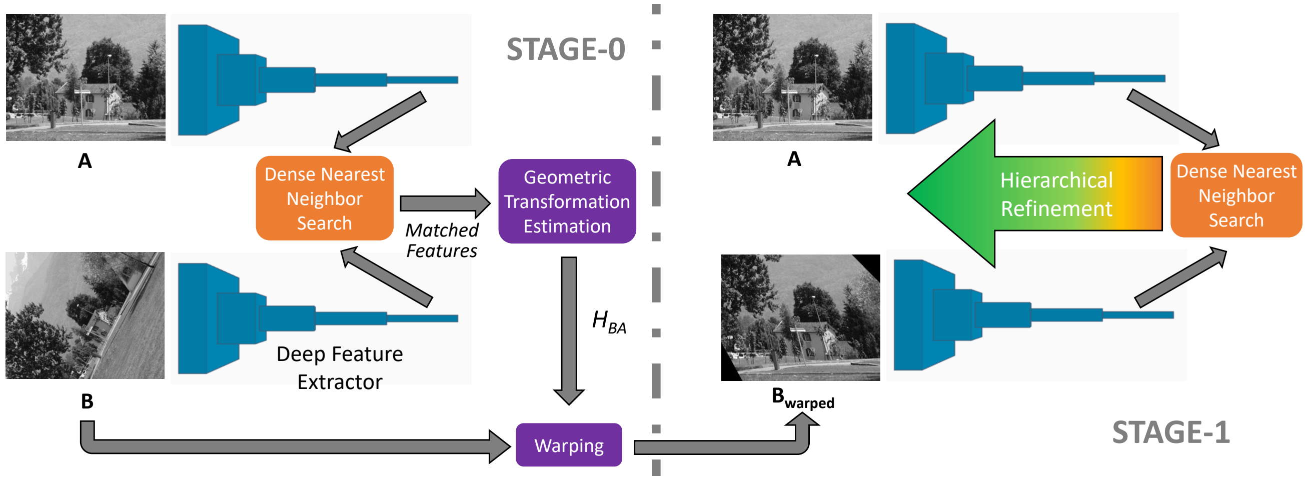

Figure 2: Architecture for Deep Feature Matching (DFM) Algorithm

We first employ DNNS to initialize matches at the terminal layer of the feature extractor. Note that, downsampled size of this layer makes them suitable for DNSS. For the VGG feature extractor, the size of the feature maps at the 5^{th} convolutional layer is $W/16 \times H/16$ where W and H are the original width and height of the image, which makes DNNS feasible. We also employ DNNS to refine the matches at finer spatial resolutions, as explained next.

3.2. Hierarchical Refinement

The outputs of DNNS at the deep layers are robust to geometric transformations and illumination changes, but they are poorly localized. Directly upsampling them to the original resolution would yield poor matching performance.

As an imperfect alternative, DNNS can also be employed in former convolutional layers without any constraint. However, finer spatial resolutions of these layers would increase the computational complexity exponentially as we move towards the original resolution. Moreover, the features at deeper layers tend to carry semantic information more, which makes them more discriminative, while the features in shallower layers have smaller receptive fields and therefore tend to be more repetitive. Hence, applying DNNS at these layers brings the risk of mismatches and erroneous elimination of feature points. Patch2Pix [73] solves the latter problem by appending the features of different layers in the original resolution at the expense of computational complexity, which worsens the first problem more. However, coarse-to-fine strategy (or hierarchical search/refinement), which is a well-known technique in computer vision and practiced in many works [44, 62], can be employed to solve both problems at the same time. A similar hierarchical matching strategy is also used in [47].

Given the set of matched pairs, $P_n^{A,B}$ at the layer n and feature maps F_{n-1}^A and F_{n-1}^B at the layer $n-1$, Hierarchical Refinement Algorithm (HRA) finds the matched points at the current layer by employing DNNS in a limited area. For each matched pair, p^A and p^B, HRA first constructs point sets Ω_A and Ω_B which are the receptive fields of the p^A and p^B at layer $n-1$. For VGG-19, receptive fields can be easily obtained by upsampling the input points by 2 and including the one-pixel right and bottom neighborhood for each upsampled point. By restricting the search area as such, DNNS can be applied with none of the aforementioned problems. Therefore, given a set of matched pairs in layer n, for each pair, we feed the patches of feature maps F^A and F^B to DNNS and get the matched pairs in layer $n-1$. HRA is presented in Algorithm 1.

Algorithm 1: Hierarchical Refinement (HRA)

```
input  : $F_{n-1}^A$, $F_{n-1}^B$ →feature maps at layer $n-1$
         $P_n^{A,B}$ →set of matched pairs at layer $n$
output: $P_{n-1}^{A,B}$ →set of matched pairs at layer $n-1$
Function HRef($F_{n-1}^A$, $F_{n-1}^B$, $P_n^{A,B}$):
    for $p^A$,$p^B$ in $P_n^{A,B}$ do
        (1)Get the receptive fields at layer $n-1$ for
          feature points defined at layer $n$
          $\Omega^A$ = receptive($p^A$)
          $\Omega^B$ = receptive($p^B$)
        (2)Perform Dense Nearest Neighbor Search
          $M^{A,B}$ = DNNS($F_{n-1}^A(\Omega^A)$,$F_{n-1}^B(\Omega^B)$)
        (3)Record the matched pair at layer $n-1$
          $P_{n-1}^{A,B}.append(transform(M^{A,B}))$
    end
    return $P_{n-1}^{A,B}$
end
```

HRA changes the number of feature points in each refinement step by selecting the correct matches and rejecting the outliers. In other words, correct and robust matches are expected to survive and might even populate through hierarchical refinement, while the outliers are expected to be rejected, as illustrated in Figure 3. As seen in this figure, there are some poorly localized matches in $Layer-5$. In $Layer-4$, the number of matches is increased, but still, their localization is inferior. Moving forward to $Layer-3$, matched points are localized much better, but there are still some erroneous matches. (Note the 2^{nd} and 3^{rd} matches from the top.) Notice that these outliers are also rejected in $Layer-2$. Finally, we ended up with better localized and more accurate matches in $Layer-1$.

3.3. Two Stage Approach

Given an image pair, one can extract the features by using a pre-trained feature extractor, perform DNNS at the terminal layer to initialize the matched pairs, and refine them up to the original resolution using HRA. The matching performance of this single-step solution will be presented in Section 4 as DFM(s1). However, we realized that this one-step solution fails during hierarchical refinement in case of severe geometric transformations. This result is due to the fact that features at shallower layers are not as robust as deep layers against geometric transformations. We also observed that even if the method is able to generate correct matches at deeper layers, these matches are eliminated while moving towards shallower layers. To overcome this issue, we first predict a homography matrix H_{BA} using the matched set of points at the terminal layer and warp B using H_{BA} to obtain warped image B_w. Next, we apply the solution mentioned above by using the reference image A and B_w. With such an approach, we are effectively first aligning the two images, then looking for possible matches, as proposed in [48]. Recently, this approach is successfully applied in [55, 42, 66].

When VGG-19 is employed as the feature extractor, we first obtain a set of matched points by feeding P_5^A and P_5^B to DNNS. Using this set of matched pairs, we predict a homography matrix H_{BA} and obtain warped image B_{warped}. We extract features one more time for the warped image. Then, we initialize a set of matched points utilizing the terminal layers' features of the warped and the reference image with DNNS. For VGG-19, each point of the pair is the parent of 4 points in the former layer. To refine the pairs, HRA applies DNNS on 2x2 feature patches iteratively until the first layer. Note that each level number of feature points might change; in other words, P_1^A may have more points than P_5^A, or vice versa.

4. Experiments and Evaluation

We have evaluated the proposed DFM technique on

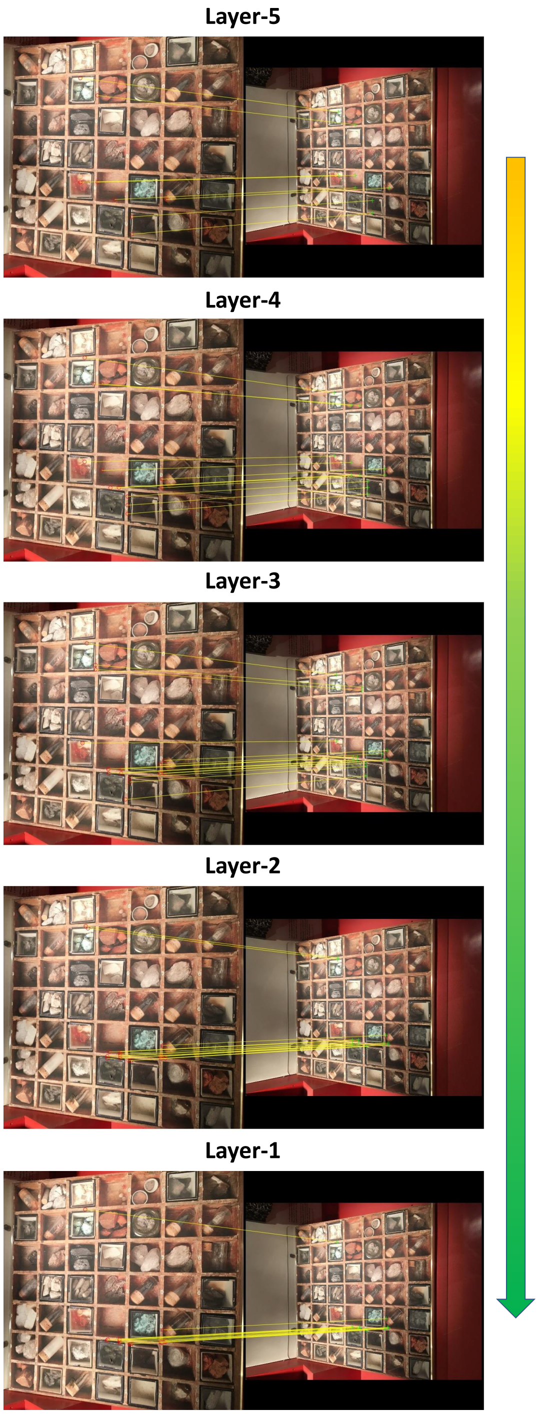

Figure 3: Visualization of Hierarchical Refinement Algorithm. Proposed method starts with Dense Nearest Neighbor Search at the $5^{th} layer$ of the VGG-19, then applies Hierarchical Refinement Algorithm until the $1^{st} layer$. For clarity, a limited number of matched pairs are shown.

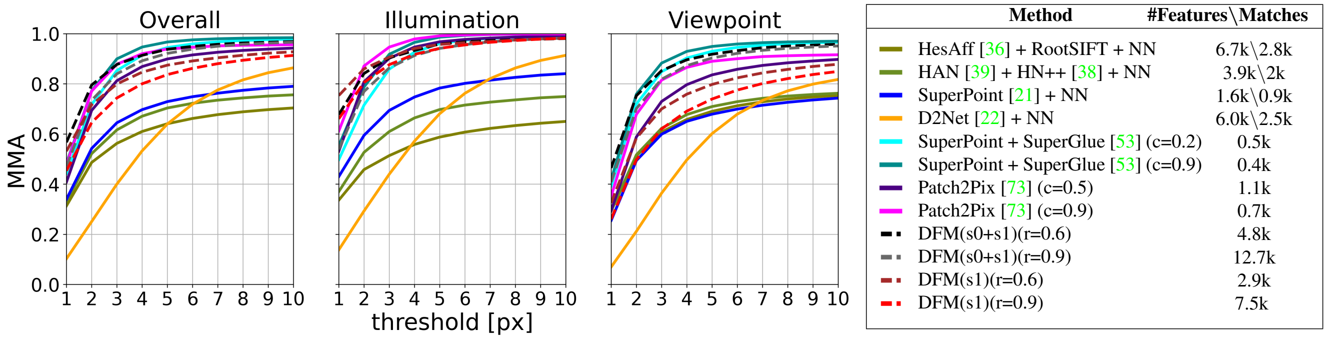

Figure 4: Feature Matching Evaluation HPatches in terms of MMA. Variants of DFM are demonstrated as dashed lines.

HPatches [4] dataset, which consists of 116 sequences in total. Each sequence has 6 images of the same scene, each captured from a different viewpoint or under different illumination. For each sequence, the dataset provides ground-truth homographies between the first image and the remaining five images. These 116 sequences are divided into two subsets; 57 sequences have significant illumination changes with the almost same viewpoint, while 59 sequences have significant viewpoint changes under similar illumination.

We have performed feature matching and homography estimation on the HPatches dataset and reported the results in the following two subsections.

4.1. Feature Matching

The performance of the feature matching task is measured by the pixel-wise distances between the matched features and their ground-truth projections on the pair images, as in [22]. Then, these values have been averaged for the whole dataset, and the percentage of the matches with matching error less than the selected threshold is reported for thresholds from 1 pixel to 10 pixels. This metric is known as mean matching accuracy (MMA) [22].

The results of various methods are illustrated in Figure 4 with the number of matched features. We reported results for four different variants of the proposed DFM. We have denoted the variants that employ two-stage approach as (s0+s1) to abbreviate Stage-0 and Stage-1. Note that Stage-1 is independent of Stage-0 and can be used solely. We have represented these one-stage variants as (s1). There are two more variants according to diversely conducted ratio tests. We abbreviate the set of ratio test thresholds $\{0.6, 0.6, 0.8, 0.9, 0.95\}$ employed in hierarchical refinement algorithm as (r=0.6) to denote the ratio threshold at the shallowest layer. For (r=0.9) the threshold set is $\{0.9, 0.9, 0.9, 0.9, 0.95\}$. For the two-stage (s0+s1) variants, we have not applied any ratio test on Stage-0. In both configurations, we have utilized conv5_3 layers of VGG-19 on Stage-0, and we have utilized conv5_2, conv4_2, conv3_2, conv2_2, conv1_2 layers on Stage-1. Our implementation will be available at https://github.com/ufukefe/DFM.

As shown in Figure 4, DFM(s0+s1)(r=0.6) outperforms all other methods for the thresholds less than 3 pixels on overall results. Moreover, all variants of the DFM compete with the state-of-the-art performance as the best second or third method depending on the case even without particular training for feature matching task. In viewpoint results, the significant effect of Stage-0 can be observed clearly.

Apart from the promising matching results, DFM has much more matches than any other algorithms. This means DFM returns denser matches, claiming that *there are at least this many matches giving this MMA results.*

4.2. Homography Estimation

Carrying out the Feature Matching evaluation in terms of MMA is not sufficient for understanding the performance of the compared algorithms. If an algorithm finds a single correct match for each pair in the dataset, then its MMA score will be 100% which is misleading. Researchers try to examine such a case by also reporting the average number of matches, but algorithms might still fail for specific pairs. In other words, an algorithm may have a sufficient number of matches, but all these matches may concentrate on the specific regions of the image pairs, preventing to achieve a high-quality homography estimation. For measuring this characteristic, following [21, 73], we also evaluate our method in terms of homography estimation accuracy for 1, 3, and 5 pixels. For this evaluation, four corners of the reference image is projected onto the pair image with estimated and ground-truth homographies, and the average pixel distance between estimated and ground-truth projections of the four corners is obtained. Then, this average distance is thresholded to conclude whether homography estimate for a pair is correct or not, and the rate of the correctly estimated homographies are reported in Table 1.

As various setups had been used for this evaluation

[21, 73, 53, 63], it is relatively difficult to make firm conclusions from Table 1. Even if almost in every setup RANSAC is employed for homography estimation, independent setups utilized different implementations (OpenCV and pydegensac). The parameter selections of RANSAC (the number of iterations, confidence and distance thresholds) are not clearly specified in all these previous setups. Moreover, the stochastic nature of the RANSAC is completely ignored. Besides, every method has performed this test by using the dataset images at different resolutions. For example, there is a remarkable amount of discrepancy between the reported results for the SuperPoint algorithm in [21, 73] as shown in Table 1.

In order to report the homography estimation performance of the proposed approach, we used M-estimator SAmple Consensus (MSAC) [65] algorithm by employing *estimateGeometricTransform* function of MATLAB. Parameters of the function are *'Confidence'*-trust of detecting the highest number of inliers-, *'MaxNumTrials'*-limit for random trials to find inliers- and *'MaxDistance'*-the maximum pixel-wise distance between a point and back projection of its pair for inliers-. These parameters were selected as *99.99, 5000, 3* respectively, and matched features from full-resolution images were used in our experiments. We reported the mean and variance of correctly estimated homographies to take MSAC's stochastic nature into account. As we achieve a small standard deviation in 10 repetitions, it can be concluded that the number of repetitions is enough to obtain statistically meaningful results for the overall dataset. We have also reported the maximum and minimum possible results by taking the best outcome for each pair (boe) and the worst outcome for each pair (woe) among 10 repeated tests. As shown in Table 1, *boe* and *woe* results differ significantly, which indicates either our MSAC configuration is not repetitive enough for individual pairs or our putative match sets is not distributed evenly over the image pairs. This approach might seem like cheating, in fact it is, and hence they are not used in comparisons; however, as the number of repeated tests is very limited, the results cannot be explained by randomness only. These results indicate that there are enough correct pairs to achieve *boe* homography estimation accuracy.

Reported in	Method	Homography Estimation Accuracy		
		≤1px	≤3px	≤5px
[21]	SuperPoint [21] + NN	0.31	0.68	0.83
[73]	SuperPoint [21] + NN	0.46	0.78	0.85
	D2Net [22] + NN	0.38	0.72	0.81
	R2D2 [46] + NN	0.47	0.78	0.83
	SuperPoint + SuperGlue [53]	0.51	0.83	0.89
	SparseNCNet [49]	0.36	0.66	0.76
	Patch2Pix [73]	0.51	0.79	0.86
	DFM(s0+s1)(r=0.6) (boe)	0.53	0.84	0.90
	DFM(s0+s1)(r=0.6) ($\mu \pm \sigma$)	0.43±.01	0.73±.01	0.83±.01
	DFM(s0+s1)(r=0.6) (woe)	0.34	0.62	0.73
	DFM(s0+s1)(r=0.9) (boe)	0.58	0.89	0.93
	DFM(s0+s1)(r=0.9) ($\mu \pm \sigma$)	0.43±.01	0.76±.01	0.87±.01
	DFM(s0+s1)(r=0.9) (woe)	0.27	0.60	0.76

Table 1: Homography Estimation results on HPatches [4]. DFM results are presented in terms of average accuracy and its standard deviation over 10 runs. Selecting the best outcome for each image pair (boe) and the worst outcome for each image pair (woe) result in significant deviation. r is the used ratio test threshold.

As shown in Table 1, DFM(s0+s1)(r=0.9) variant is the second-best algorithm for the 5-pixel threshold in terms of average accuracy. Nevertheless, even we believe the necessity of this evaluation, we avoid making further comments due to the inconsistencies.

5. Conclusion

In this paper, we have proposed a method for finding point correspondences between two images by using the features extracted by deep neural networks. We show that benefiting from basic vision techniques and using the features extracted by off-the-shelf deep neural networks, state-of-the-art performance can be achieved without any special training for feature matching. We demonstrate that the features extracted by VGG network at its deepest layers are quite discriminative that a simple nearest neighbor search can be applied densely to initiate matches, decreasing the necessity of a learning-based dense feature matcher, like NCNet [50]. We also present our results that these initial matches can be refined hierarchically through shallower layers.

Nonetheless, as a limitation, for initial warping we assume that the matched scenes are planar. Hence, our two-stage solution tends to perform well in planar scenes as demonstrated on HPatches dataset [4] and may be useful for aerial image matching while it may have some problems in the presence of non-planar image pairs. For such scenarios, the proposed method should be either modified to solve appropriate geometric transformation instead of homography, or should be utilized with one-stage version. For the one-stage case, weakness of features at shallow layers limit the performance by preventing the method from finding matches at those layers, which we observed in the Hpatches dataset, in case of severe geometric transformations.

References

[1] Pablo Fernández Alcantarilla, Adrien Bartoli, and Andrew J Davison. Kaze features. In *European conference on computer vision*, pages 214–227. Springer, 2012. 1

[2] Pablo F Alcantarilla and T Solutions. Fast explicit diffusion for accelerated features in nonlinear scale spaces. *IEEE Trans. Patt. Anal. Mach. Intell*, 34(7):1281–1298, 2011. 1

[3] Vassileios Balntas, Edward Johns, Lilian Tang, and Krystian Mikolajczyk. Pn-net: Conjoined triple deep network for learning local image descriptors. *arXiv preprint arXiv:1601.05030*, 2016. 2

[4] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5173–5182, 2017. 1, 6, 7

[5] Axel Barroso-Laguna, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Key. net: Keypoint detection by handcrafted and learned cnn filters. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5836–5844, 2019. 2

[6] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006. 1

[7] Aritra Bhowmik, Stefan Gumhold, Carsten Rother, and Eric Brachmann. Reinforced feature points: Optimizing feature detection and description for a high-level task. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4948–4957, 2020. 2

[8] Irving Biederman. Recognition-by-components: a theory of human image understanding. *Psychological review*, 94(2):115, 1987. 2

[9] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. 2

[10] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. Dsac-differentiable ransac for camera localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6684–6692, 2017. 2

[11] Eric Brachmann and Carsten Rother. Expert sample consensus applied to camera re-localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7525–7534, 2019. 2

[12] Eric Brachmann and Carsten Rother. Neural-guided ransac: Learning where to sample model hypotheses. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4322–4331, 2019. 2

[13] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In *European conference on computer vision*, pages 778–792. Springer, 2010. 1

[14] Ondrej Chum and Jiri Matas. Matching with prosac-progressive sample consensus. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 220–226. IEEE, 2005. 1

[15] Ondřej Chum, Jiří Matas, and Josef Kittler. Locally optimized ransac. In *Joint Pattern Recognition Symposium*, pages 236–243. Springer, 2003. 1

[16] Ondrej Chum, Tomas Werner, and Jiri Matas. Two-view geometry estimation unaffected by a dominant plane. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 772–779. IEEE, 2005. 1

[17] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26:2292–2300, 2013. 3

[18] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005. 1

[19] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Deep image homography estimation. *arXiv preprint arXiv:1606.03798*, 2016. 2

[20] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Toward geometric deep slam. *arXiv preprint arXiv:1707.07410*, 2017. 2, 3

[21] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018. 2, 3, 6, 7

[22] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable cnn for joint description and detection of local features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8092–8101, 2019. 2, 3, 6, 7

[23] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 1

[24] Christopher G Harris, Mike Stephens, et al. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988. 1

[25] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, USA, 2 edition, 2003. 1

[26] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. 2

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[29] Karel Lenc and Andrea Vedaldi. Learning covariant feature detectors. In *European conference on computer vision*, pages 100–117. Springer, 2016. 2

[30] Stefan Leutenegger, Margarita Chli, and Roland Y Siegwart. Brisk: Binary robust invariant scalable keypoints. In *2011 International conference on computer vision*, pages 2548–2555. Ieee, 2011. 1

[31] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 2

[32] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 3

[33] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 1, 3

[34] Jiayi Ma, Ji Zhao, Junjun Jiang, Huabing Zhou, and Xiaojie Guo. Locality preserving matching. *International Journal of Computer Vision*, 127(5):512–531, 2019. 1

[35] Jiri Matas, Ondrej Chum, Martin Urban, and Tomás Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 22(10):761–767, 2004. 1

[36] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *International journal of computer vision*, 60(1):63–86, 2004. 6

[37] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE transactions on pattern analysis and machine intelligence*, 27(10):1615–1630, 2005. 1

[38] Anastasiya Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. *arXiv preprint arXiv:1705.10872*, 2017. 2, 6

[39] Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Repeatability is not enough: Learning affine regions via discriminability. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 284–300, 2018. 2, 6

[40] Ty Nguyen, Steven W Chen, Shreyas S Shivakumar, Camillo Jose Taylor, and Vijay Kumar. Unsupervised deep homography: A fast and robust homography estimation model. *IEEE Robotics and Automation Letters*, 3(3):2346–2353, 2018. 2

[41] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. Lf-net: Learning local features from images. *arXiv preprint arXiv:1805.09662*, 2018. 2

[42] Udit Singh Parihar, Aniket Gujarathi, K. Mehta, Satyajit Tourani, Sourav Garg, Michael Milford, and K. Krishna. Rord: Rotation-robust descriptors and orthographic views for local feature matching. 2021. 5

[43] Rémi Pautrat, Viktor Larsson, Martin R Oswald, and Marc Pollefeys. Online invariance selection for local feature descriptors. In *European Conference on Computer Vision*, pages 707–724. Springer, 2020. 3

[44] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4161–4170, 2017. 4

[45] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015. 2

[46] Jerome Revaud, Philippe Weinzaepfel, César De Souza, Noe Pion, Gabriela Csurka, Yohann Cabon, and Martin Humenberger. R2d2: repeatable and reliable detector and descriptor. *arXiv preprint arXiv:1906.06195*, 2019. 2, 7

[47] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Deepmatching: Hierarchical deformable dense matching. *International Journal of Computer Vision*, 120(3):300–323, 2016. 4

[48] Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. Convolutional neural network architecture for geometric matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6148–6157, 2017. 2, 5

[49] Ignacio Rocco, Relja Arandjelović, and Josef Sivic. Efficient neighbourhood consensus networks via submanifold sparse convolutions. In *European Conference on Computer Vision*, pages 605–621. Springer, 2020. 2, 7

[50] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelovic, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Ncnet: Neighbourhood consensus networks for estimating image correspondences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 2, 3, 7

[51] Edward Rosten and Tom Drummond. Fusing points and lines for high performance tracking. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1508–1515. Ieee, 2005. 1

[52] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011. 1

[53] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020. 2, 3, 6, 7

[54] Nikolay Savinov, Akihito Seki, Lubor Ladicky, Torsten Sattler, and Marc Pollefeys. Quad-networks: unsupervised learning to rank for interest point detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1822–1830, 2017. 2

[55] Xi Shen, François Darmon, Alexei A Efros, and Mathieu Aubry. Ransac-flow: generic two-stage image alignment. *arXiv preprint arXiv:2004.01526*, 2020. 5

[56] Xuelun Shen, Cheng Wang, Xin Li, Zenglei Yu, Jonathan Li, Chenglu Wen, Ming Cheng, and Zijian He. Rf-net: An end-to-end image matching network based on receptive field. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8132–8140, 2019. 2

[57] Roger N Shepard and Jacqueline Metzler. Mental rotation of three-dimensional objects. *Science*, 171(3972):701–703, 1971. 2

[58] Edgar Simo-Serra, Eduard Trulls, Luis Ferraz, Iasonas Kokkinos, Pascal Fua, and Francesc Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 118–126, 2015. 2

[59] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[60] Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967. 3

[61] Stephen M Smith and J Michael Brady. Susan—a new approach to low level image processing. *International journal of computer vision*, 23(1):45–78, 1997. 1

[62] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8934–8943, 2018. 4

[63] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. *CVPR*, 2021. 7

[64] Yurun Tian, Xin Yu, Bin Fan, Fuchao Wu, Huub Heijnen, and Vassileios Balntas. Sosnet: Second order similarity regularization for local descriptor learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11016–11025, 2019. 2

[65] Philip HS Torr and Andrew Zisserman. Mlesac: A new robust estimator with application to estimating image geometry. *Computer vision and image understanding*, 78(1):138–156, 2000. 1, 7

[66] Prune Truong, Martin Danelljan, and Radu Timofte. Glunet: Global-local universal network for dense flow and correspondences. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6258–6268, 2020. 5

[67] Yannick Verdie, Kwang Yi, Pascal Fua, and Vincent Lepetit. Tilde: A temporally invariant learned detector. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5279–5288, 2015. 2

[68] Qi Zhang Jiaya Jia Xiaoyong Shen, Li Xu. Multi-modal and multi-spectral registration for natural images. In *ECCV*, 2014. 1

[69] Tsun-Yi Yang, Duy-Kien Nguyen, Huub Heijnen, and Vassileios Balntas. Ur2kid: Unifying retrieval, keypoint detection, and keypoint description without local correspondence supervision. *arXiv preprint arXiv:2001.07252*, 2020. 2

[70] Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to find good correspondences. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2666–2674, 2018. 2

[71] Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4353–4361, 2015. 2

[72] Jiahui Zhang, Dawei Sun, Zixin Luo, Anbang Yao, Lei Zhou, Tianwei Shen, Yurong Chen, Long Quan, and Hongen Liao. Learning two-view correspondences and geometry using order-aware network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5845–5854, 2019. 2

[73] Qunjie Zhou, Torsten Sattler, and Laura Leal-Taixe. Patch2pix: Epipolar-guided pixel-level correspondences. In *CVPR*, 2021. 2, 3, 4, 6, 7