MD-Net: Multi-Detector for Local Feature Extraction

Emanuele Santellani [*], Christian Sormann [*], Mattia Rossi [†], Andreas Kuhn [†], Friedrich Fraundorfer [*]
[*] Institute of Computer Graphics and Vision, Graz University of Technology, Austria
Email: {emanuele.santellani, christian.sormann, fraundorfer}@icg.tugraz.at
[†] R&D Center - Stuttgart Laboratory 1, Sony Europe B.V., Germany. Email: {mattia.rossi, andreas.kuhn}@sony.com

arXiv:2208.05350v1 [cs.CV] 10 Aug 2022

***Abstract*—Establishing a sparse set of keypoint correspondences between images is a fundamental task in many computer vision pipelines. Often, this translates into a computationally expensive nearest neighbor search, where every keypoint descriptor at one image must be compared with all the descriptors at the others. In order to lower the computational cost of the matching phase, we propose a deep feature extraction network capable of detecting a predefined number of complementary sets of keypoints at each image. Since only the descriptors within the same set need to be compared across the different images, the matching phase computational complexity decreases with the number of sets. We train our network to predict the keypoints and compute the corresponding descriptors jointly. In particular, in order to learn complementary sets of keypoints, we introduce a novel unsupervised loss which penalizes intersections among the different sets. Additionally, we propose a novel descriptor-based weighting scheme meant to penalize the detection of keypoints with non-discriminative descriptors. With extensive experiments we show that our feature extraction network, trained only on synthetically warped images and in a fully unsupervised manner, achieves competitive results on 3D reconstruction and re-localization tasks at a reduced matching complexity.**

I. Introduction

Being able to extract reliable sets of point correspondences between images is a fundamental requirement for a large variety of computer vision pipelines, such as Structure from Motion (SfM) [1], SLAM [2], Visual Localization [3], object detection [4], [5] and object tracking [6]. The problem has been historically divided into two sequential steps: local feature extraction and pairwise matching.

The feature extraction step starts with the detection of a sparse set of salient points, referred to as keypoints, in each image. Objects visible in multiple images should trigger the detection of the same set of keypoints, in order to permit the establishment of correspondences between the images. As a consequence, the detection process is required to be robust to some degree of image alteration, such as illumination and viewpoint changes or occlusions. Several algorithms have been proposed during the last decades, classified as either blob [7]–[9], corner [10], [11] or region detectors [12]. The feature extraction step continues with the assignment of a descriptor vector to each detected keypoint, whose purpose is to describe the keypoint neighborhood. Among the many proposed algorithms for descriptor extraction [7], [13]–[16], SIFT [7] and its improved versions [17], [18] are the most successful ones and still remain widely used nowadays. More recently, with the spread of data-driven approaches, a multitude of local feature extraction methods based on deep learning have emerged. Early methods relied on existing keypoint detectors and focused on designing networks meant to extract the corresponding descriptors [22], [23]. Later, motivated by the tight entanglement of keypoints and descriptors, the focus shifted towards the design of network architectures meant to predict keypoints and descriptors jointly [19], [20], [24]. Our method belongs to the latter group.

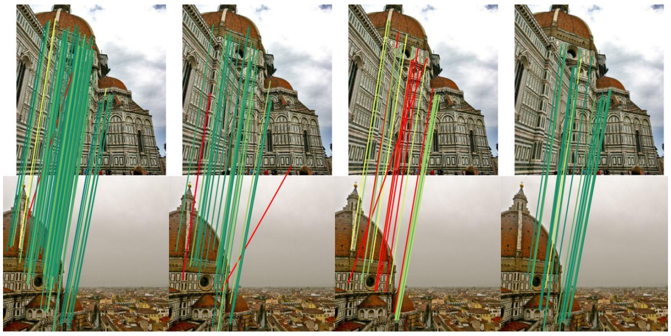

Fig. 1. Example pair from the Image Matching Benchmark [21] stereo task for the scene *Florence Cathedral Side*. The lines represent the RANSAC inlier matches, color coded according to the pixel error from green, no error, to red, error above 5px.

After the feature extraction step, in the pairwise matching, the descriptors at the different images are compared against each other in order to establish the correspondences. This last step is often responsible for a considerable part of the total computational cost of the sparse correspondence search. In order to reduce the matching complexity, we propose a deep feature extraction network capable of extracting multiple complementary keypoint sets. This permits to restrict the comparison between the descriptors at the different images only to the descriptors that belong to the same set, thus reducing the matching computational complexity. In order to train such a network, we propose a novel unsupervised loss that discourages overlaps between the different keypoint sets.

For 3D reconstruction tasks, it has been shown that the keypoint distribution has a strong influence on the quality of the recovered camera poses [25], which leads to worse results when large image portions are not covered. To encourage an even keypoint distribution, we employ the unsupervised loss formulation originally proposed by [19]. However, the use of this loss may lead to detections in non-discriminative regions as well. In order to mitigate this side effect, we propose a variance-based weighting scheme that dampens the loss in areas where the descriptors are less discriminative.

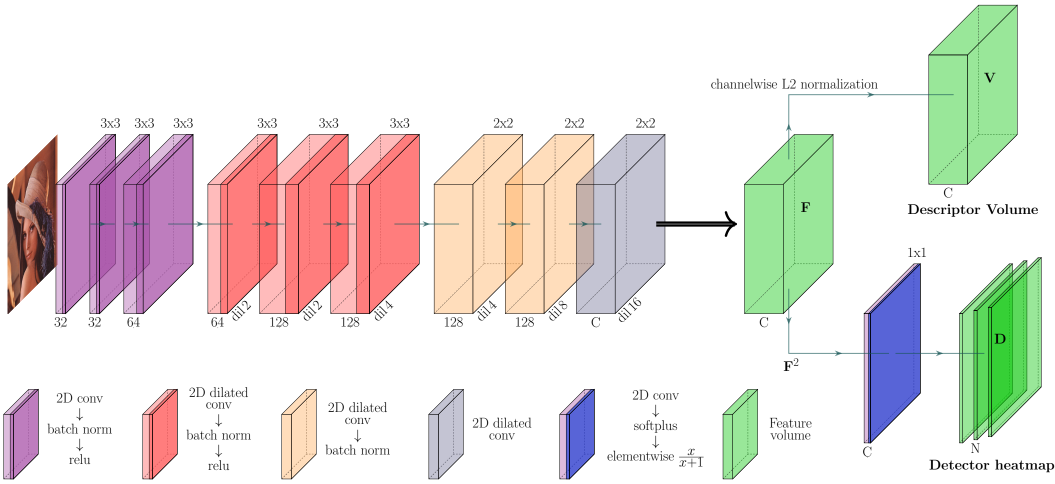
3x3
3x3
3x3
3x3
3x3
3x3
2x2
2x2
2x2
32
32
64
64
dil2
128
dil2
128
dil4
128
dil4
128
dil8
C
dil16
channelwise L2 normalization
V
C
Descriptor Volume
F
C
F^2
1x1
C
D
N
Detector heatmaps
2D conv ↓ batch norm ↓ relu
2D dilated conv ↓ batch norm ↓ relu
2D dilated conv ↓ batch norm
2D dilated conv
2D conv ↓ softplus ↓ elementwise $\frac{x}{x+1}$
Feature volume

Fig. 2. Network architecture. For each convolution, the kernel size and the number of output channels are indicated at the top and at the bottom, respectively

Differently from the classical methods, which rely on carefully handcrafted algorithms, deep methods require large amounts of data in order to generalize to unseen scenes. Moreover, depth maps and poses generated from 3D reconstructions, that are possibly inaccurate and incomplete, are often used for training [19], [20]. In an attempt to overcome these limitations, we train our feature extraction network exclusively on images warped using random homographies. Furthermore, we augment the data with photometric distortions.

Our contribution is threefold:

- We propose a deep architecture, named MD-Net, trained with a novel unsupervised loss formulation, which is capable of extracting multiple complementary sets of features. This reduces the computational complexity of the subsequent matching phase.
- A training loss re-weighting based on the local descriptor variance is introduced. This discourages the detection of keypoints with less discriminative descriptors.
- Our feature extraction network, which is trained exclusively on images warped using random homographies, generalizes well to 3D-related tasks as proven on two well known online benchmarks [21], [26].

II. Related works

In the last decades, a multitude of algorithms addressing the sparse correspondence problem have been designed: in-depth evaluations have been carried out in [27]–[29]. With the advent of deep learning, data-driven methods were proposed to address one or more steps of the existing feature extraction pipelines. Early methods were trained to either detect repeatable keypoints [30]–[33], or to distill compact descriptors from normalized patches, previously extracted by means of a classical method [22], [23], [34]. Later, deep methods were proposed to both detect keypoints and extract their descriptors [35]–[37], with a shift towards joint learning with D2-Net, R2D2 and ASLFeat [19], [20], [24]. Differently from the already listed approaches, [38] uses reinforcement learning to train a deep network for sparse feature extraction, obtaining good performance at the cost of a more expensive training procedure. Our method is most closely related to R2D2 [19], with which we share the core architecture and one of the unsupervised losses. Differently from R2D2, we employ a variance-based loss dampening, supported by a two-stage training scheme, to avoid detections in areas where the resulting descriptors are not locally discriminative. Additionally, our network is capable of detecting multiple complementary sets of keypoints. While in [39] a weight is predicted for each local features based on the relevance for the downstream image retrieval task, our loss re-weighting is based on a parameter-free local measure of discriminativeness.

Deep learning has been applied successfully to the matching task as well, with [40] completely replacing the traditional matching based on mutual nearest neighbors and other methods proposing learnt outlier filters [41], [42]. These methods lead to better matching results, but increase the matching computational complexity significantly.

Multiple strategies have been proposed in order to reduce the matching computational complexity for SfM pipelines. These are particularly useful when dealing with the reconstruction of a scene from an unordered set of images, potentially captured in different conditions. In fact, in this scenario, correspondences need to be established by matching all the possible image pairs, which results in a complexity growing quadratically in both the number of images and the number of extracted keypoints per image. One possible approach toward reducing this computational burden is to lower the number of image pairs by using strategies based on image similarities [43]. Alternatively, the number of matching operations can be reduced by using approximate nearest neighbor algorithms [44], [45]. However, the former approach introduces the risk of missing valid image pairs and the latter decreases the quality of the matches [21]. For those reasons, when high reconstruction quality is required, many 3D reconstruction pipelines still match all the possible image pairs and use the exact Mutual Nearest Neighbor (MNN) matching [1], [46]. With MD-Net, we propose a novel approach that reduces the

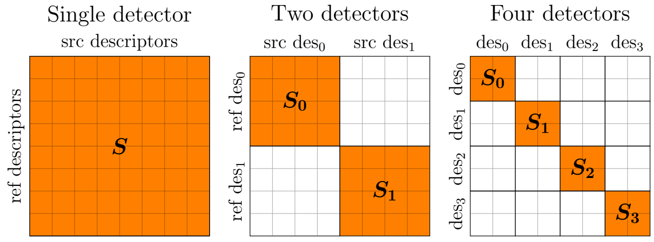

Fig. 3. Graphical representation of the computational complexity associated with the mutual nearest neighbors descriptor matching. While in the *single-detector* case the S matrix is computed for each possible descriptor combination, in the *multi-detector* cases it is sufficient to compute the smaller S_n between the different sets of descriptors des$_n$ with $n \in \{0, \ldots, N-1\}$. The computational cost, shown in orange, results halved in the case of *two detectors* and reduced by a factor of 4 in the case of *four detectors*.

matching complexity by extracting a predefined number of disjointed feature sets at each image, which permits to limit the matching to the sole features in the same set.

III. Model overview

A. Network architecture

The network architecture, depicted in Fig. 2, is a streamlined version of R2D2 [19] with the addition of our multi-detector branch. The backbone consists of a fully convolutional network where the commonly used convolution pyramid is replaced by a series of dilated convolutions, meant to increase the effective field-of-view of the network without lowering the output resolution. The backbone processes the input RGB image $\boldsymbol{I} \in \mathbb{R}^{3\times H\times W}$ and outputs the feature volume $\boldsymbol{F} \in \mathbb{R}^{C\times H\times W}$. The feature volume is then fed to two different branches: the descriptor branch and the multi-detector branch. In the descriptor branch the feature volume is L2-normalized along the channel dimension to produce the final descriptor volume $\boldsymbol{V} \in \mathbb{R}^{C\times H\times W}$. This associates a C-dimensional descriptor vector to each pixel of the input image. In the multi-detector branch, instead, the feature volume $\boldsymbol{F}$ is squared and a single 1x1 convolutional layer is used to generate a detection heatmap volume $\boldsymbol{D} \in \mathbb{R}^{N\times H\times W}$ where N is the desired number of keypoint sets. In fact, each channel of this volume, hereafter referred as $\boldsymbol{D}^n \in \mathbb{R}^{H\times W}$ with $n = 0, 1, \ldots, N-1$, will be used to extract one set of keypoints. The resulting Multi-Detector network, named MD-Net, is rather compact and counts less than half a million parameters.

B. Feature Extraction and Matching

At each heatmap $\boldsymbol{D}^n$, the candidate keypoints are detected as the pixel coordinates of the heatmap local maxima, after filtering out low values and applying a local Non Maxima Suppression (NMS). Given a budget of M keypoints, for each heatmap we select only the M/N candidate keypoints with the highest values in the heatmap. Finally, we obtain the local features by coupling each keypoint with its descriptor, sampled from the descriptor volume $\boldsymbol{V}$ at the keypoint pixel location.

For a pair of images with M features each, the Mutual Nearest Neighbor matching boils down to computing a distance matrix $\boldsymbol{S} \in \mathbb{R}^{M\times M}$ between the two image descriptor sets, which has a computational complexity $\mathcal{O}(M^2)$. Thanks to our network architecture instead, only descriptors associated with the same detector heatmap need to be matched, which reduces the distance matrix size to $M/N \times M/N$ and the corresponding computational complexity for each set to $\mathcal{O}(M^2/N^2)$. Repeating the matching for each one of the N feature sets results in an overall complexity reduction factor N, as follows:

$$\mathcal{O}_{\text{N-sets}} = N \cdot \mathcal{O}\left(\frac{M^2}{N^2}\right) = \mathcal{O}\left(\frac{M^2}{N}\right) \tag{1}$$

A visual intuition for the reduced computational complexity is provided in Fig. 3. The aggregated matches are obtained joining all the sets of matches.

IV. Loss formulation

The loss formulation can be split in two main components: the *descriptor loss* and *detector loss*, applied at the output of the corresponding branches, respectively.

A. Descriptor loss

The *descriptor loss* goal is to promote discriminative descriptors, that permit to recognize the correct correspondences between the keypoints of two images. Similarly to previous works [22], [23], we frame descriptor learning as a metric learning problem, where we promote that two corresponding keypoints have similar descriptors, while non corresponding keypoints should have dissimilar ones. To this purpose, we employ a simple hinged formulation of the Triplet Loss:

$$\mathcal{L}_{Triplet} = \underset{t\in\mathcal{T}}{\text{mean}} \left(\max(0, m - \boldsymbol{v}_{\boldsymbol{a}}^t \cdot \boldsymbol{v}_{\boldsymbol{p}}^t + \boldsymbol{v}_{\boldsymbol{a}}^t \cdot \boldsymbol{v}_{\boldsymbol{n}}^t) \right) \tag{2}$$

where $\cdot$ denotes the inner product, $\mathcal{T}$ is the set of all the sampled triplets, m is the hinge margin and $\boldsymbol{v}_{\boldsymbol{a}}^t$, $\boldsymbol{v}_{\boldsymbol{p}}^t$, $\boldsymbol{v}_{\boldsymbol{n}}^t \in \mathbb{R}^C$ refer to the *anchor* descriptor, the *positive* correspondence descriptor and one *negative* descriptor, respectively. While it is trivial to build the *(anchor, positive)* descriptor pair, if the geometric transformation $g : \mathbb{R}^2 \mapsto \mathbb{R}^2$ that relates the considered image pair is known, there is a virtually infinite number of possible *(anchor, negative)* candidates. As suggested in [22], we pick the *negative* following the *Hardest-in-Batch* strategy.

B. Detector loss

The *detector loss* goal is twofold. First, promoting heatmaps with well localized maxima, as these will determine the detected keypoints. Second, promoting repeatable heatmaps: content appearing in two images should lead to similar heatmaps, such that keypoint correspondences can be established between the two images. We design our loss as the sum of three components: the *peakyness loss*, the *similarity loss* and the *dissimilarity loss*. While the first two losses are applied to all the detection heatmaps in $\boldsymbol{D}$ independently and then mean aggregated, the *dissimilarity* loss formulation considers each possible pair of detection heatmaps, in order to discourage any overlap between sets of keypoints selected by different detectors. For the sake of clarity, in the following we express each loss for the single pixel (i, j). The losses are then mean aggregated over the entire $H \times W$ image domain.

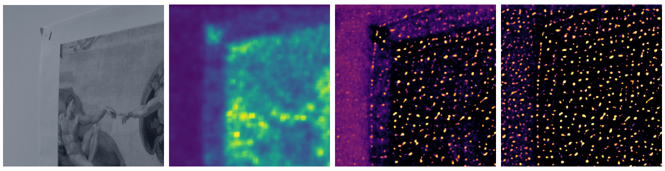

Fig. 4. Qualitative evaluation of the *peakyness* loss variance-based weighting. On the left, the source image and the computed local descriptors variance. On the right, the detection heatmaps obtained when training with and without our variance weighting in Eq. 3, respectively. Our variance-based weighting scheme smooths out the peaks in the non-discriminative region at the left border of the image, thus avoiding detections there.

1) Peakyness loss: In order to encourage the network to produce well distributed local peaks, while avoiding non-discriminative areas, we propose a modified version of the *peaky* loss formulated in [19]. The loss is defined as follows:

$$\mathcal{L}_{\text{peaky}}(\boldsymbol{D}^n)_{ij} = W_{ij}\left(1 - \left(\max_{kl\in\mathcal{P}_{ij}} \boldsymbol{D}^n_{kl} - \operatorname*{mean}_{kl\in\mathcal{P}_{ij}} \boldsymbol{D}^n_{kl}\right)\right) \quad (3)$$

where $\mathcal{P}_{ij}$ is a square patch centered at the pixel (i,j) and W_{ij} is a weight designed to avoid peaks in areas where local descriptors are not discriminative defined as follows:

$$W_{ij} = \operatorname*{mean}_{c\in\mathcal{C}}\left(\left(\operatorname*{mean}_{pq\in\mathcal{B}_{ij}} \boldsymbol{F}^2_{pq} - \left(\operatorname*{mean}_{pq\in\mathcal{B}_{ij}} \boldsymbol{F}_{pq}\right)^2\right)_c\right) \quad (4)$$

and it represents the local variance of the backbone output $\boldsymbol{F}$, computed over a patch $\mathcal{B}_{ij}$ centered at (i,j), averaged along the channel dimension. Additionally, the loss in Eq.(3) is computed on the detection heatmaps in $\overline{\boldsymbol{D}} \in \mathbb{R}^{N\times H\times W}$ obtained from the warped image $g(\boldsymbol{I}) \in \mathbb{R}^{H\times W}$. The two losses are averaged. An example of the effect of the pixelwise weighting is shown in Fig. 4, where the detection heatmap appears smoother in the less discriminative regions.

2) Similarity loss: In order to promote repeatable heatmaps, we adopt the following loss, that enforces consistency between the heatmaps produced by $\boldsymbol{I}$ and $g(\boldsymbol{I})$:

$$\mathcal{L}_{\text{sim}}(\boldsymbol{D}^n, \overline{\boldsymbol{D}}^n)_{ij} = \left(\boldsymbol{D}^n_{ij} - g^{-1}\left(\overline{\boldsymbol{D}}^n\right)_{ij}\right)^2 \quad (5)$$

where $g^{-1}(\cdot)$ denotes the inverse warping.

3) Dissimilarity loss: Finally, in order to promote that the N detection heatmaps in $\boldsymbol{D}$ lead to different sets of keypoints, we propose a novel loss that penalizes co-located peaks for each pair $(\boldsymbol{D}^m, \boldsymbol{D}^n)$. Our loss is formulated as follows:

$$\mathcal{L}_{\text{dissim}}(\boldsymbol{D}^0, \cdots, \boldsymbol{D}^N)_{ij} = \binom{N}{2}^{-1} \sum_{\substack{0\le n<N-1\\ n<m<N}} \boldsymbol{D}^m_{ij}\boldsymbol{D}^n_{ij} \quad (6)$$

where N is the number of detectors and the binomial $\binom{N}{2}$ is the number of possible detector heatmap combinations. Similarly to the *peakyness* loss, Eq. (6) is applied to the N detection heatmaps in $\overline{\boldsymbol{D}}$ as well. Fig. 5 provides an example of the keypoint sets obtained for $N = 2$.

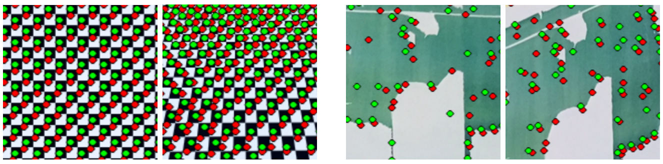

Fig. 5. Example images showing the complementarity of the keypoints sets detected by our architecture with 2 detectors. The two sets of keypoints are represented in red and green. On the left, a pair of checkerboard images: the red set keypoints are extracted always at the white squares, the green set ones at the black squares. It's worth noticing that the network has never seen checkerboard patterns during the training. On the right, another example pair.

V. Model training

While the local variance computed over the input image can be helpful in discerning textured and flat areas, it does not directly relate to the local descriptor discriminativeness. For this reason, Eq. (3) employs the backbone output local variance instead, which is related to the descriptor volume directly and it is therefore more suitable to avoid keypoint detections in areas whose descriptors would not be particularly discriminative. However, this reasoning does not hold true at the beginning of training, when the network weights are randomly distributed and the predicted descriptor volume is not meaningful. Thus, we adopt a *two-stage training* procedure:

1) First, in the *descriptor volume priming*, we train only the backbone and the descriptor branch with the loss $\mathcal{L} = \mathcal{L}_{\text{triplet}}$.
2) Then, in the *joint training*, we train our overall architecture with $\mathcal{L} = \mathcal{L}_{triplet} + \alpha\mathcal{L}_{peaky} + \beta\mathcal{L}_{sim} + \gamma\mathcal{L}_{dissim}$ where α, β and γ balance the individual losses.

The descriptor volume priming represents the main training effort, while the joint training needs only few iterations. An added benefit of this training procedure is that changing the number of keypoint sets N requires us to repeat only the joint training stage. Finally, during the joint training, the local variance W is used purely as a weighting term, i.e., the weight gradients do not participate in the backpropagation.

VI. Experiments

A. Training details

We train MD-Net on 192 × 192 patches randomly drawn from the *Revisiting Oxford and Paris distractors* dataset [47]. We implement our model in PyTorch [48] and train it with the Adam optimizer [49] ($\beta_1 = 0.9$, $\beta_2 = 0.999$) and fixed lr $= 1e^{-4}$ on a single Nvidia GTX1080Ti. The descriptor volume priming consists of $70k$ iterations and takes 13 hours. Instead, the joint training consists only of $1k$ iterations and is completed in 12 minutes. Each iteration employs a batch of 10 patches. Overall, the training procedure consumes a total of 710k images. Concerning the descriptor loss in Eq. (2), we set the hinge margin $m = 1$, sample the *positive* and *negative* descriptors on a regular grid with step 10px and classify a descriptor as negative candidate when it is more than 5px away from the correct location. We adopt 128-dimensional descriptors. Concerning the peaky loss in Eqs. (3) and (4), we set $\mathcal{P}_{ij}$ and $\mathcal{B}_{ij}$ to be 17 × 17 and 9 × 9 patches, respectively.

TABLE I
ABLATION STUDIES ON HPATCHES - OVERALL

Method	MMA ↑			MS ↑		
	@1px	@2p	@3px	@1px	@2px	@3px
MD-1-Net No-Var	0.385	0.622	0.740	0.161	0.255	0.298
MD-1-Net	**0.398**	**0.638**	**0.757**	0.196	0.306	0.356
MD-2-Net	**0.398**	0.630	0.743	0.206	0.317	**0.369**
MD-4-Net	**0.398**	0.621	0.731	**0.210**	**0.319**	**0.369**
MD-8-Net	0.346	0.541	0.636	0.184	0.280	0.325

TABLE II
COMPARISON ON HPATCHES

	Method	MMA ↑			MS ↑		
		@1px	@2p	@3px	@1px	@2px	@3px
v	MD-2-Net (ours)	0.316	**0.600**	**0.722**	0.171	0.313	0.393
	R2D2 [19]	0.280	0.568	0.700	0.118	0.228	0.273
	ASLFeat [20]	**0.332**	0.565	0.675	**0.203**	**0.338**	**0.398**
	Upright-SIFT [7]	0.313	0.472	0.533	0.167	0.247	0.277
i	MD-2-Net (ours)	**0.480**	0.658	0.765	0.242	0.323	0.368
	R2D2 [19]	0.377	0.660	**0.797**	0.170	0.285	0.336
	ASLFeat [20]	0.469	**0.664**	0.774	**0.290**	**0.398**	**0.456**
	Upright-SIFT [7]	0.344	0.475	0.528	0.161	0.216	0.238
overall	MD-2-Net (ours)	**0.398**	**0.630**	0.743	0.206	0.317	0.369
	R2D2 [19]	0.326	0.612	**0.747**	0.143	0.255	0.304
	ASLFeat [20]	**0.398**	0.613	0.723	**0.245**	**0.367**	**0.426**
	Upright-SIFT [7]	0.327	0.473	0.531	0.164	0.232	0.258

TABLE III
AACHEN DAY-NIGHT VISUAL LOCALIZATION V1.1

	Method	successfully localized percentage ↑		
		0.25m, 2°	0.5m, 5°	5m, 10°
8k kpts	MD-2-net (ours)	**70.2**	83.2	96.3
	R2D2 [19]	66.0	82.2	94.8
	ASLFeat [20]	69.6	**84.8**	**97.4**
	Upright-SIFT [7]	54.5	69.6	79.1
20k kpts	MD-2-net (ours)	69.1	84.8	**97.9**
	R2D2 [19]	68.1	83.8	96.9
	ASLFeat [20]	**72.3**	**86.4**	**97.9**
	Upright-SIFT [7]	62.3	78.5	90.1

Finally, for the training scenario with $N = 2$ detectors, we use the loss weights $\alpha = 1.0$, $\beta = 4.0$ and $\gamma = 0.5$.

B. Benchmarks

We test MD-Net on three popular benchmarks: *HPatches* [50], *Aachen* day-night [51] and the *Image Matching Benchmark* [21]. In all the experiments we employ MD-Net with $N = 2$ detectors, denoted MD-2-Net. The filtering threshold and the NMS radius introduced for the keypoint extraction in Sec. III-B are set to 0.7 and 3px, respectively. We run MD-2-Net on a multi-scale image pyramid obtained by down scaling the input image by a factor $\sqrt{2}$ until the shortest image dimension drops below 256px. Finally, for each detector, we select the M/N keypoints with the highest scores across the multiple scales. The main metrics in the experiments are the following, involving a pair of images that have to be matched:

- **MMA**: The *Mean Matching Accuracy* is the mean ratio between the number of correct matches and the total number of proposed matches [24].
- **MS**: The *Matching Score* is the mean ratio between the number of correct matches and the number of keypoints extracted at one image in the area shared with the other. The metric is computed for both the images and the results are averaged [36].
- **mAA**: The *mean Average Accuracy* is the area under the curve of the fraction of correctly estimated relative poses as a function of the pose error [21].

MMA and MS are evaluated at a given pixel error threshold. In all the tables we represent the best result in bold and we underline the second best. We compare MD-2-Net with two state-of-the-art deep feature extraction networks: R2D2 [19] and ASLFeat [20]. We employ their official implementations and adopt either their default parameters or those specified by the authors for each benchmark, when provided. In addition, we consider also Upright-SIFT [7], the baseline method in the Image Matching Benchmark [21], employing their implementation. For the purpose of a fair comparison, we do not compare with methods employing deep matchers, such as [40].

1) HPatches [50]: This benchmark considers both indoor and outdoor scenes divided in two sets: *v* contains images of mostly planar scenes captured from different angles in the same lighting conditions, while *i* contains images captured from a fixed camera in different lighting conditions. Each scene, 56 for the v set and 52 for i, contains 6 images and the ground truth homographies linking the first image to all the others. We evaluate following the methodology of D2-net [24] with a maximum budget of 5k keypoints per image and evaluation on the sets *v*, *i* and their union, denoted *overall*. The performance at error thresholds greater than a few pixels are of little interest in real world applications, such as in 3D reconstruction, due to the tight geometric filters employed. For this reason, in Tab. II we report only the numerical values of MMA and MS up to 3px error. MD-2-Net obtains competitive MMA results on all the three image sets, at all the error thresholds. In particular, it is the best performing method in the *overall* set at both 1px and 2px, while following R2D2 closely at 3px. Additionally, MD-2-Net provides good MS results, following the best performing method ASLFeat.

2) Aachen Day-Night [26], [51]: This online benchmark is part of the long-term visual localization benchmark [52]. It consists of two sets of images of the German city Aachen. The first set is captured during daytime and the corresponding ground truth camera intrinsics and poses are provided. The second set is captured at night instead and the benchmark target is to re-localize these query images using the first set. The online benchmark has been recently updated to v1.1 with more precise ground truth poses and additional query images. For a fair comparison, we run MD-2-Net, R2D2, ASLFeat and UprightSIFT using the same re-localization pipeline based on COLMAP [1], available at [53]. The results are reported in Tab III, where MD-2-Net achieves the highest percentages of successfully localized images at the (0.25m, 2°) error threshold when considering a budget of 8k keypoints and it follows the other deep methods closely at the higher error thresholds.

TABLE IV
IMAGE MATCHING BENCHMARK - RESTRICTED KEYPOINTS 2048

		Stereo					Multiview					Avg
	Method	NF	NI ↑	Rep@3px ↑	MS@3px ↑	mAA@10° ↑	NM ↑	NL ↑	TL ↑	ATE ↓	mAA@10° ↑	mAA@10° ↑
Phototourism	MD-2-net (ours)	2047.5	**233.0**	0.396	**0.792**	**0.455**	238.6	**1391.5**	**4.604**	**0.411**	**0.708**	**0.581**
	R2D2 [19]	2048.0	201.5	0.429	0.746	0.390	**294.3**	1225.9	4.280	0.478	0.640	0.515
	ASLfeat [20]	2042.6	126.0	**0.431**	0.749	0.337	157.5	1106.6	4.415	0.533	0.556	0.446
	Upright-SIFT [7]	1892.8	98.6	0.333	0.788	0.383	148.0	1165.7	4.118	0.524	0.555	0.469
PragueParks	MD-2-net (ours)	2048.0	**175.5**	0.039	0.027	**0.542**	236.3	**605.8**	**3.197**	6.753	**0.451**	**0.497**
	R2D2 [19]	2048.0	167.0	0.032	0.025	0.539	**338.9**	526.0	3.170	6.837	0.444	0.491
	ASLfeat [20]	2048.0	110.5	0.059	**0.029**	0.401	217.1	574.4	3.036	6.414	0.400	0.403
	Upright-SIFT [7]	2048.0	119.8	**0.060**	0.027	0.414	157.3	433.3	2.989	**5.666**	0.361	0.387

In contrast to R2D2 [19] and ASLFeat [20], our network MD-2-Net is trained exclusively using synthetic homographies and neither on day-night pairs nor on 3D data.

3) Image Matching Benchmark [21]: This is a recent online benchmark proposed to evaluate the performance of local features [21] in the context of *stereo pose recovery* and *multiview reconstruction* on two sets of sequences, namely Phototourism and PragueParks. It considers multiple intermediate metrics (Number of Features (NF), Number of Inlier matches (NI), Repeatability (Rep), Matching Score (MS), Number of inlier Matches filtered by COLMAP [1] (NM), Number of triangulated Landmarks (NL), Track Length (TL), Absolute Trajectory Error (ATL)) as well as the resulting mean Average Accuracy (mAA) up to 10°. For a more detailed description of the metrics, we refer to the benchmark documentation [21]. We evaluate MD-2-Net on the restricted keypoint category: maximum 2048 keypoints per image. The benchmark results are reported in Tab. IV. MD-2-Net achieves competitive results in all the metrics. In particular, it provides the best mAA on both the sets, for both the stereo and multiview tasks. A qualitative comparison between the considered methods is provided in Fig. 1 for the stereo pose recovery task.

C. Ablation studies

In order to test the performance of our method with a varying number of detectors, we train different instances of MD-Net with $N = 1, 2, 4$ and 8 detectors, using the same primed backbone, and test them on the HPatches dataset [50]. It is important to note that γ, the weight of the dissimilarity loss $\mathcal{L}_{\text{dissim}}$ plays a crucial role: the smaller its value, the higher the chances for multiple detectors to find very similar keypoints, and vice versa. To this purpose, we introduce the *Separability* metric at n pixels, denoted *Sep@$n$px*. This measures the overlap between all the detected keypoints as one minus the ratio between the number of keypoints selected by one detector that are closer than n pixels to any other keypoint detected by the other detectors and the total number of detected keypoints. The higher the separability, the lower the chances of observing keypoints from different detectors falling withing n pixels from each other. As an example, in our test with $N = 4$, $\gamma = 2.0$ leads to *Sep@3px* $= 0.971$ while setting $\gamma = 1.5$ leads to the lower *Sep@3px* $= 0.81$. In order to ensure that *Sep@3px* is higher than 0.95, we empirically set $\gamma = 0.5, 2.0$ and 18.0 for the cases $N = 2, 4$ and 8, respectively. The results of our test are reported in Tab. I (refer to Sec. VI for more details about the dataset and metrics) and show that the model trained with two detectors, denoted *MD-2-Net*, offers the best trade-off between the single, the four and the eight detector versions, in terms of metrics and matching complexity.

When comparing runtimes, matching all the possible pairs between 300 images with 8000 keypoints each takes 288s with the single detector, 161s using two, 87s using four and only 51s when using eight, with an average of 6.4ms, 3.6ms, 1.9ms and 1.1ms per pair, respectively. Tests are carried out on a single Nvidia GTX1080Ti and the matching time considers the scores computation, the mutual nearest neighbor search and the match aggregation.

Finally, in order to assess the effectiveness of our peakyness loss weighting scheme, we train our model without the weighting term W_{ij} in Eq. 3. The resulting network, denoted *MD-1-Net No-Var* in Tab. I, performs considerably worse than *MD-1-Net*, which employs our weighting scheme instead.

VII. CONCLUSION AND FUTURE WORKS

We introduced MD-Net, a novel deep feature extraction network capable of extracting multiple disjointed sets of local features: these can be matched independently, thus reducing the computational complexity of the matching phase. The high *separability* values obtained in our analysis, with varying number of detectors, confirm the effectiveness of the novel unsupervised *dissimilarity loss* at the basis of MD-Net. Additionally, we proposed a variance-based loss dampening scheme that, together with the two-stage training, avoids the detection of keypoints associated with non-discriminative descriptors.

Our experiments show that the network, trained unsupervised, achieves competitive results on different 3D-related tasks at a reduced matching complexity, despite being trained exclusively on images warped with random homographies.

In the future, we will consider different strategies to select the keypoints at each heatmap, and couple the proposed multi-detector paradigm with a deep matcher architecture, such as [40], in order to benefit from additional learnt geometric consistency while keeping the matching cost manageable.

Acknowledgement: This work has been supported by the FFG, Contract No. 881844: "Pro^{2}Future".

References

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

[2] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[3] L. Svärm, O. Enqvist, F. Kahl, and M. Oskarsson, "City-scale localization for cameras with known vertical direction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 7, pp. 1455–1461, 2017.

[4] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[5] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on statistical learning in computer vision, ECCV*, vol. 1, no. 1-22. Prague, 2004, pp. 1–2.

[6] H. Zhou, Y. Yuan, and C. Shi, "Object tracking using sift features and mean shift," *Computer Vision and Image Understanding*, vol. 113, no. 3, pp. 345–352, 2009, special Issue on Video Analysis.

[7] D. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, 1999, pp. 1150–1157 vol.2.

[8] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *International journal of computer vision*, vol. 60, no. 1, pp. 63–86, 2004.

[9] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[10] C. Harris, M. Stephens *et al.*, "A combined corner and edge detector," in *Alvey vision conference*, vol. 15, no. 50. Citeseer, 1988, pp. 10–5244.

[11] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European conference on computer vision*. Springer, 2006, pp. 430–443.

[12] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and vision computing*, vol. 22, no. 10, pp. 761–767, 2004.

[13] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International Conference on Computer Vision*, 2011, pp. 2564–2571.

[14] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Computer Vision – ECCV 2006*, A. Leonardis, H. Bischof, and A. Pinz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 404–417.

[15] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *Computer Vision – ECCV 2010*, K. Daniilidis, P. Maragos, and N. Paragios, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 778–792.

[16] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *2011 International Conference on Computer Vision*, 2011, pp. 2548–2555.

[17] Y. Ke and R. Sukthankar, "Pca-sift: a more distinctive representation for local image descriptors," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 2, 2004, pp. II–II.

[18] R. Arandjelović and A. Zisserman, "Three things everyone should know to improve object retrieval," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 2911–2918.

[19] J. Revaud, P. Weinzaepfel, C. R. de Souza, and M. Humenberger, "R2D2: repeatable and reliable detector and descriptor," in *NeurIPS*, 2019.

[20] 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 the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 6589–6598.

[21] Y. Jin, D. Mishkin, A. Mishchuk, J. Matas, P. Fua, K. M. Yi, and E. Trulls, "Image Matching across Wide Baselines: From Paper to Practice," *International Journal of Computer Vision*, 2020.

[22] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas, "Working hard to know your neighbor's margins: Local descriptor learning loss," *arXiv preprint arXiv:1705.10872*, 2017.

[23] Y. Tian, B. Fan, and F. Wu, "L2-net: Deep learning of discriminative patch descriptor in euclidean space," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 661–669.

[24] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. 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*, 2019, pp. 8092–8101.

[25] A. Kuhn, T. Price, J.-M. Frahm, and H. Mayer, "Down to earth: Using semantics for robust hypothesis selection for the five-point algorithm," in *Pattern Recognition*, V. Roth and T. Vetter, Eds. Cham: Springer International Publishing, 2017, pp. 389–400.

[26] T. Sattler, T. Weyand, B. Leibe, and L. Kobbelt, "Image retrieval for image-based localization revisited." in *BMVC*, vol. 1, no. 2, 2012, p. 4.

[27] T. Tuytelaars and K. Mikolajczyk, *Local invariant feature detectors: a survey*. Now Publishers Inc, 2008.

[28] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk, "Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5173–5182.

[29] J. L. Schonberger, H. Hardmeier, T. Sattler, and M. Pollefeys, "Comparative evaluation of hand-crafted and learned local features," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1482–1491.

[30] Y. Verdie, K. Yi, P. Fua, and V. Lepetit, "Tilde: A temporally invariant learned detector," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 5279–5288.

[31] C. Strecha, A. Lindner, K. Ali, and P. Fua, "Training for task specific keypoint detection," in *Joint pattern recognition symposium*. Springer, 2009, pp. 151–160.

[32] N. Savinov, A. Seki, L. Ladicky, T. Sattler, and M. Pollefeys, "Quad-networks: unsupervised learning to rank for interest point detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1822–1830.

[33] A. Barroso-Laguna, E. Riba, D. Ponsa, and K. Mikolajczyk, "Key. net: Keypoint detection by handcrafted and learned cnn filters," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5836–5844.

[34] 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 the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 016–11 025.

[35] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "Lift: Learned invariant feature transform," in *European conference on computer vision*. Springer, 2016, pp. 467–483.

[36] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 224–236.

[37] Y. Ono, E. Trulls, P. Fua, and K. M. Yi, "Lf-net: Learning local features from images," *arXiv preprint arXiv:1805.09662*, 2018.

[38] M. J. Tyszkiewicz, P. Fua, and E. Trulls, "Disk: Learning local features with policy gradient," *arXiv preprint arXiv:2006.13566*, 2020.

[39] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han, "Large-scale image retrieval with attentive deep local features," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3456–3465.

[40] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 4938–4947.

[41] J. Zhang, D. Sun, Z. Luo, A. Yao, L. Zhou, T. Shen, Y. Chen, L. Quan, and H. Liao, "Learning two-view correspondences and geometry using order-aware network," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5845–5854.

[42] W. Sun, W. Jiang, E. Trulls, A. Tagliasacchi, and K. M. Yi, "Acne: Attentive context normalization for robust permutation-equivariant learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 286–11 295.

[43] J. L. Schönberger, T. Price, T. Sattler, J.-M. Frahm, and M. Pollefeys, "A vote-and-verify strategy for fast spatial verification in image retrieval," in *Asian Conference on Computer Vision (ACCV)*, 2016.

[44] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu, "An optimal algorithm for approximate nearest neighbor searching fixed

dimensions," *Journal of the ACM (JACM)*, vol. 45, no. 6, pp. 891–923, 1998.

[45] M. Muja and D. Lowe, "Flann-fast library for approximate nearest neighbors user manual," *Computer Science Department, University of British Columbia, Vancouver, BC, Canada*, vol. 5, 2009.

[46] "Opensfm." [Online]. Available: https://opensfm.org/

[47] F. Radenović, A. Iscen, G. Tolias, Y. Avrithis, and O. Chum, "Revisiting oxford and paris: Large-scale image retrieval benchmarking," in *CVPR*, 2018.

[48] 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 *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035.

[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[50] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk, "Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5173–5182.

[51] Z. Zhang, T. Sattler, and D. Scaramuzza, "Reference pose generation for visual localization via learned features and view synthesis," *arXiv preprint arXiv:2005.05179*, vol. 5, no. 7, p. 9, 2020.

[52] "Long-term visual localization." [Online]. Available: https://www.visuallocalization.net/

[53] "Visual localization benchmark - local features." [Online]. Available: https://github.com/tsattler/visuallocalizationbenchmark/blob/master/local_feature_evaluation