ALIKE: Accurate and Lightweight Keypoint Detection and Descriptor Extraction

Xiaoming Zhao, Xingming Wu, Jinyu Miao, Weihai Chen*, *Member, IEEE*, Peter C. Y. Chen, and Zhengguo Li*, *Senior Member, IEEE*

***Abstract*—Existing methods detect the keypoints in a non-differentiable way, therefore they can not directly optimize the position of keypoints through back-propagation. To address this issue, we present a partially differentiable keypoint detection module, which outputs accurate sub-pixel keypoints. The reprojection loss is then proposed to directly optimize these sub-pixel keypoints, and the dispersity peak loss is presented for accurate keypoints regularization. We also extract the descriptors in a sub-pixel way, and they are trained with the stable neural reprojection error loss. Moreover, a lightweight network is designed for keypoint detection and descriptor extraction, which can run at 95 frames per second for 640×480 images on a commercial GPU. On homography estimation, camera pose estimation, and visual (re-)localization tasks, the proposed method achieves equivalent performance with the state-of-the-art approaches, while greatly reduces the inference time.**

***Index Terms*—Keypoint detection, keypoint descriptor, deep learning, local feature, image feature extraction, image matching**

I. Introduction

SPARSE keypoints and descriptors are compact representations for efficient image matching [1]. Hence they are widely used in real-time visual applications like simultaneous localization and mapping (SLAM) systems [2], [3], and high dynamic range imaging (HDRI) [4], [5].

For keypoint detection and descriptor extraction, early hand-crafted algorithms [6]–[8] were built upon limited human heuristics, which could lead to unstable keypoints and confusable descriptors in complex images. So neural networks are explored for this task in recent years. Early neural network based methods mainly focus on descriptor extraction from image patches [9]–[11]. Latter, many excellent methods address the keypoint detection and descriptor extraction with a single network [12]–[15]. Some of them [12], [13], [16]–[20] treat the keypoint detection as a score map estimation problem, where the score of a pixel indicates the probability that it is a keypoint. And then they train the intermediate score map with the synthetic ground truth score map and/or the similarity of score patches. Others [14], [15], [21], [22] define the score map based on the spatial and/or channel variation of dense feature map.

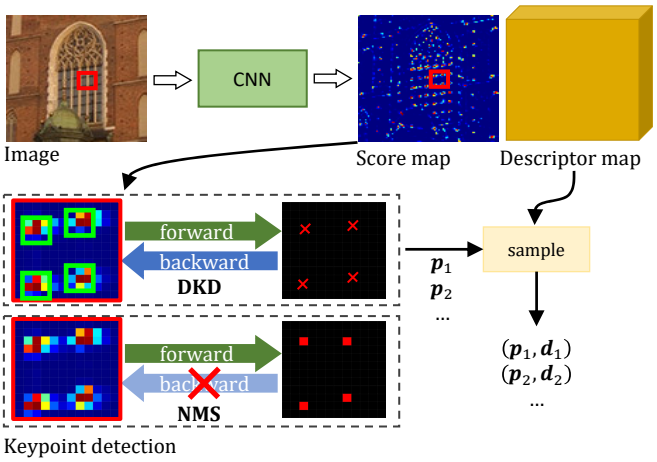


Fig. 1. We propose a Differentiable Keypoint Detection (DKD) for score map based keypoint detection and descriptor extraction. The keypoint detection on a score map patch (the red rectangle area) is illustrated in this figure. Compared with the Non-Maximum-Suppression (NMS) based methods, the DKD can back-propagate the gradients and produce sub-pixel keypoints. Thus we can directly optimize the position of detected keypoints to be accurate.

However, both of them have to detect the keypoints on score map with Non-Maximum-Suppression (NMS). The NMS simply chooses the pixel with maximum local score, it is non-differentiable and the gradients can not backward though (left bottom in Fig.1). So the position of detected keypoints can not be optimized directly. We observed that a keypoint is supported by its local score patch (the green areas in the middle score patch of Fig.1). The score distribution in the patch would influence the keypoint position, even if the position of maximum score remains unchanged. This allows us to accurately capture subtle changes of local score distribution. To this purpose, we adopt the score map based pipeline (Fig.1) for Differentiable Keypoint Detection (DKD). The DKD applies the `softargmax` [23]–[25] operation on local score patches. Thus the gradients can backward from keypoints to score map (middle row in Fig.1). We present the keypoint reprojection loss to train the sub-pixel keypoints detected from DKD, which directly minimizes the reprojection distance of detected keypoints between images. Inspired by the peak loss [13], [20], a dispersity peak loss is further introduced to avoid blob scores on score map, which forces the score map to be accurately "peaky" at the keypoint position.

Besides the sub-pixel keypoints from score map, the de-

This work was supported by the National Nature Science Foundation of China under Grant No. 61620106012. (*Corresponding authors: Weihai Chen; Zhengguo Li.*)

Xiaoming Zhao, Xingming Wu, Jinyu Miao, and Weihai Chen are with the School of Automation Science and Electrical Engineering, Beihang University, Beijing, 100191 (e-mail: xmzhao@buaa.edu.cn, wxmbuaa@163.com, mjy0519@buaa.edu.cn, and whchen@buaa.edu.cn).

Peter C. Y. Chen is with the Department of Mechanical Engineering, National University of Singapore, Singapore (email: mpechenp@nus.edu.sg).

Zhengguo Li is with the SRO department, Institute for Infocomm Research, 1 Fusionopolis Way, Singapore (email: ezgli@i2r.a-star.edu.sg).

scriptors are also sampled in a sub-pixel way from the dense descriptor map (Fig.1). The widely use method to train sparse descriptors is the triplet loss [12], [14], [15], [20], but our experiments indicate that the training of sub-pixel sparse descriptors with triplet loss is tricky and unstable, as they only cover the sampled keypoints of entire descriptor map. Inspired by sparse-to-dense matching methods [26]–[29], we adopt the recent proposed neural reprojection error (NRE) loss [29], which covers the entire descriptor map in training and thus provides more stable convergence.

On the other hand, estimation of score map and descriptor map must be very efficient, as the keypoint detection and descriptor extraction is a fundamental task for many real-time applications. However, existing methods pay more attention on the matching performance rather than running efficiency. To further improve the running efficiency for robots [30], a lightweight convolutional neural network (CNN) is designed by concatenating multi-level features [15], [20], [31], for both localization accuracy and representation capabilities. The experiments indicate that this lightweight network has comparable performance to existing methods while runs much faster.

To summarize, the main contributions of this paper are as follows:

- We present a differentiable keypoint detection module, as well as the reprojection and dispersity peak loss for accurate and repeatable keypoints training.
- We employ the NRE [29] loss to train the estimated dense descriptor map so that the model can converge more stably than using the triplet loss.
- A lightweight network aggregating hierarchical features is designed for efficient keypoint detection and descriptor extraction, which can run at 95 FPS (frame per second) on a commercial GPU while achieving comparable performances to state-of-the-art (SOTA) approaches.

The rest of this paper is organized as follows. Section II reviews the deep learning based methods. Section III introduces the lightweight network, proposes the differentiable keypoint detection module, and presents the training losses. In section IV, we first analyze each part of the proposed method, then conduct the evaluation and discussion on different tasks. Finally, a conclusion is given in section V.

II. Related Works

Deep learning based methods can be roughly divided into three categories: the patch-based, score map based and the description-and-detection methods.

A. Patch-based methods

The early patch-based methods only extract descriptors from image patches. The Matchnet [32] estimates similarity of descriptors, and trains them with cross entropy loss. Latter, TFeat [33] introduces the triplet loss for patch descriptors. And it is then widely used in latter patch-based methods [9]–[11], [16]. L2-Net [9] presents a progressive sampling strategy for triplet sampling. LIFT [16] mimics the SIFT [6]. HardNet [10] and SOSNet [11] introduce the hardest negative triplet and second order similarity of descriptors. However, the patch-based methods only focus on descriptors extraction, and their receptive field is limited in the image patch.

B. Score map based methods

These methods estimate a score map and a descriptor map, where the score map indicates keypoint probability. Tilde [18] first trains the score map on webcam dataset with SIFT [6] keypoints as ground truth. Quad-networks [19] trains the score map by ranking scores to eliminate the need of ground truth labeling. And KeyNet [34] estimates the score map with the handcrafted and learned features, it extracts keypoints from score map with `softargmax` [23]. Besides pure keypoint detection, recent methods estimate both the score map and descriptor map. LFNet [17] also uses `softargmax` [23], but it still trains on score map rather than keypoints. SuperPoint [12] first trains a MagicPoint model on synthetic dataset, and then bootstraps the score map on real images with homographic adaption strategy, and its descriptors are trained with triplet loss. This strategy is also adopted in MLIFeat [31] and SEKD [20]. R2D2 [13] identifies keypoints as reliable and repeatable positions in image and trains reliability through AP loss [35]. HDD-Net [36] weights the features with `softargmax` scores in grids to train the score map and feature map simultaneously. Furthermore, DISK [37] and reinforced SP [38] relax the keypoint detection and descriptor matching as probabilistic processes and train the network with reinforcement learning.

However, all these methods except KeyNet [34] train on intermediate score map rather than directly on keypoints, as they are extracted with non-differentiable NMS. Our DKD is most similar to KeyNet [34] and also utilizes `softargmax`. But our method dose not require handcrafted features and any pseudo keypoint annotations. KeyNet [34] detects keypoints on fixed patches and can not handle the keypoints on patch boundaries, whereas our keypoints are extracted from flexible potential positions and therefore no boundary issues. Besides keypoints training, previous works mainly adopt the triplet loss to train descriptors, which is proved to be unstable in our experiments for sub-pixel descriptors. Inspired by sparse-to-dense matching [27]–[29], we utilize the NRE loss [29] to train the sub-pixel descriptors.

C. Description-and-detection methods

Unlike score map based methods (detection-then-description), description-and-detection methods recognize keypoints as distinctive positions in an image and generate the score map by computing distinctiveness of the descriptor map or feature maps. D2Net [14] first proposes this concept, it applies channel-wise ratio-to-max and spatial-wise softmax on the descriptor map to compute score map. ASLFeat [15] improves it with channel-wise and spatial-wise peakiness on multi-layer feature maps. UR2KiD [22] computes the score map with the L2 response of features. And D2D [21] selects the absolute and relative salient points from feature map as keypoints. However, they also have to supervise on the computed score map rather than directly on the keypoints, as they also use the no-differentiable NMS to detect keypoints.

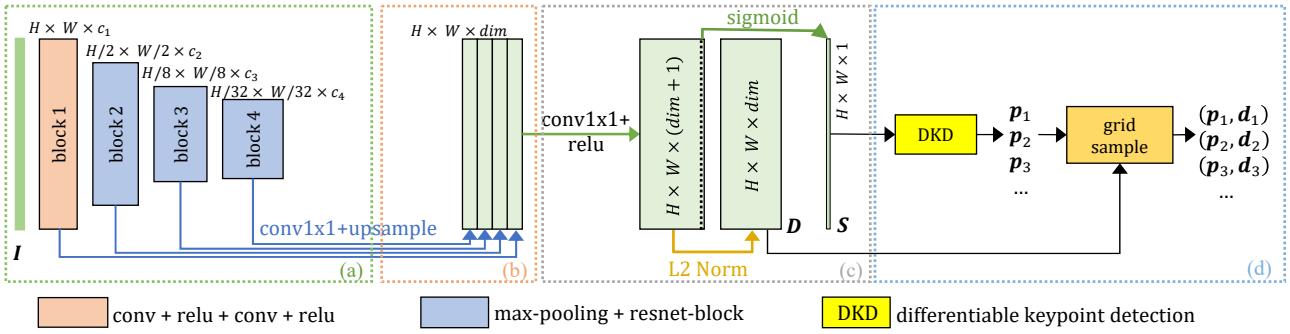


Fig. 2. The proposed network architecture. (a) image feature encoder, (b) feature aggregation, (c) feature extraction head, (d) differentiable keypoint detection and descriptor sampling. The $\boldsymbol{I}$ denotes the input image, $\boldsymbol{p}$ is the position of keypoint, and $\boldsymbol{d}$ represents the descriptor of a keypoint.

III. Methods

Following the score map based approaches [12], [13], for an input image $\boldsymbol{I} \in \mathbb{R}^{H\times W\times 3}$, the network first estimates a score map $\boldsymbol{S} \in \mathbb{R}^{H\times W}$ and a descriptor map $\boldsymbol{D} \in \mathbb{R}^{H\times W\times dim}$. Then sub-pixel keypoints $\{\boldsymbol{p} = [u, v]^T\}$ are detected with the DKD from the score map $\boldsymbol{S}$, and their corresponding descriptors $\{\boldsymbol{d} \in \mathbb{R}^{dim}\}$ are sampled from $\boldsymbol{D}$. In the following, we first introduce the network architecture and the DKD module. Then the training losses for accurate keypoints and discriminative descriptors are presented.

A. Network architecture

As illustrated in Fig.2, the network is designed to be as lightweight as possible to improve running efficiency [39]. It only has a basic encoder for feature extraction. Then the feature aggregation module assembles multi-level features [15], [20], [31] to retain both the localization and representation capabilities. For accurate localization performance, the feature extraction head estimates the score map $\boldsymbol{S}$ and descriptor map $\boldsymbol{D}$ under original image resolution. The details for each part are as follows:

(a) **The image feature encoder** encodes the input image $\boldsymbol{I} \in \mathbb{R}^{H\times W\times 3}$ to feature maps. It contains four blocks. The first block is a two-layer 3×3 convolution with "ReLU" activation [40], and the last three blocks contain a max-pooling layer and a 3×3 basic ResNet block [41]. The number of output features for i-th module is denote as c_i. The downsample rate of max-pooling in block 2 is $1/2$, and $1/4$ in block 3 and 4. Under this formulation, the maximum receptive field is 204×204 on the image.

(b) **The feature aggregation module** aggregates multi-level features from the encoder. An 1×1 convolution and bilinear up-sampling are first used to adapt the channels, and then they are simply concatenated together.

(c) **The feature extraction head** outputs an $H \times W \times (dim + 1)$ feature map in which the first dim channels are L2 normalized as the descriptor map $\boldsymbol{D}$ and the last channel is normalized by "Sigmoid" activation as score map $\boldsymbol{S}$.

(d) **The differentiable keypoint detection and descriptor sampling** first detects the sub-pixel keypoints from the score map $\boldsymbol{S}$ (section III-B), and then samples their descriptors from the dense descriptor map $\boldsymbol{D}$.

B. Differentiable keypoint detection module

To detect keypoints in score map $\boldsymbol{S}$, a widely used method is the NMS [12]–[14]. It finds the pixels with the maximum score within local windows. This operation is equivalent to the `argmax` in a local $N \times N$ window

$$[\hat{i}, \hat{j}]^T_{NMS} = \arg\max_{i,j} \{s(i, j) \mid 0 \le i, j < N\}, \tag{1}$$

where $s(i, j)$ denotes the score of the local position (i, j). However, in this formula, the output position is decoupled with the score map, thus it is a non-differentiable operation and can not embrace the power of deep learning.

To couple the keypoints with score map, we propose extracting differentiable keypoints from local windows with `softargmax`. Formally, the NMSed score map (Fig. 3(c)) is first obtained by suppressing the non-maximum scores $s = \boldsymbol{S}(u, v)$ in local $N \times N$ windows:

$$s = \begin{cases} s_{max} & s = s_{max} \\ 0 & others \end{cases}, \tag{2}$$

where $s_{max} = \max s(i, j)$ is the local maximum score. Then, a threshold th is applied on the NMSed score map to filter out low response scores (Fig. 3(d)). In NMS-based methods, keypoints $\{[u, v]^T_{NMS}\}$ are extracted in this step. We step further by peeking the scores in the local windows centered on NMS keypoints $\{[u, v]^T_{NMS}\}$ (Fig. 3(f)), and extracting local soft coordinates $\{[\hat{i}, \hat{j}]^T_{soft}\}$ with `softargmax`.

Considering a local $N \times N$ window, its scores are normalized with softmax:

$$s'(i, j) = \mathbf{softmax}\left(\frac{s(i, j) - s_{max}}{t_{det}}\right), \tag{3}$$

where t_{det} is the temperature which controls the "sharpness" of the normalization. And the $\mathbf{softmax}$ normalizes $\boldsymbol{x}$ as

$$\mathbf{softmax}(\boldsymbol{x}) = \frac{\exp(\boldsymbol{x})}{\sum \exp(\boldsymbol{x})}. \tag{4}$$

The $s'(i, j)$ indicates the probabilities of $[i, j]^T$ to be the keypoint. Thus the expectation position of keypoint in the local window can be given by integral regression [24], [25]

$$[\hat{i}, \hat{j}]^T_{soft} = \sum_{0 \le i,j < N} s'(i, j)[i, j]^T. \tag{5}$$

Now, the output subpixel-level keypoints is given as

$$\boldsymbol{p} = [u, v]^T_{soft} = [u, v]^T_{NMS} + [\hat{i}, \hat{j}]^T_{soft}. \tag{6}$$

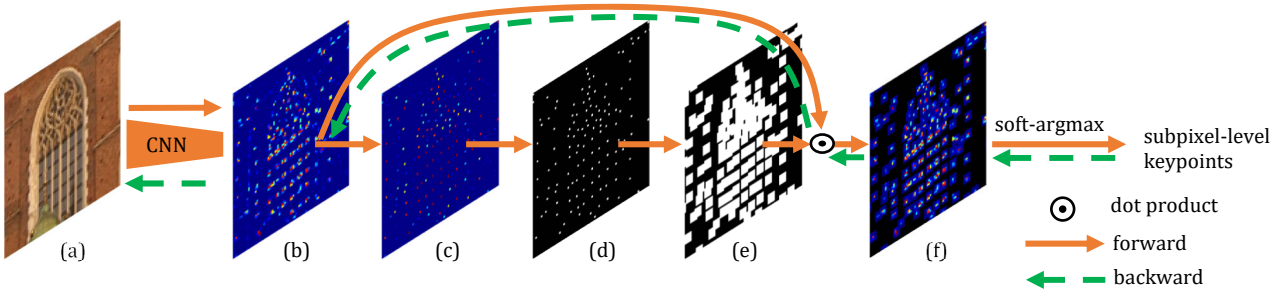


Fig. 3. The differentiable keypoint detection (DKD) module. (a) input image, (b) the estimated score map, (c) the NMSed score map, (d) the coordinates after threshold, (e) the local windows for differentiable keypoint detection, (f) the scores in local windows. It first extracts the pixel-level keypoints by using NMS and threshold, then the scores in local windows are used to extract the subpixel-level keypoints. The gradient flows in the backward process are marked as dash green arrows. And all the score maps are encoded with "jet" colormap. Best viewed in color.

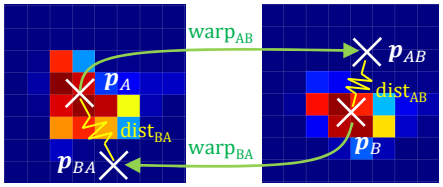


Fig. 4. The reprojection loss pulls the warped keypoint ($\boldsymbol{p}_{AB}$, $\boldsymbol{p}_{BA}$) and its corresponding keypoint ($\boldsymbol{p}_B$, $\boldsymbol{p}_A$) together. In this figure, dist_{AB} and dist_{BA} are the reprojection distance between actual keypoints and warped keypoints, $\mathbf{warp}_{AB}$ and $\mathbf{warp}_{BA}$ are the differentiable warp functions.

In this formulation, the first NMS term $[u, v]^T_{NMS}$ indicates the pixel-level keypoint position and is non-differentiable. While the second local soft coordinate term $[\hat{i}, \hat{j}]^T_{soft}$ represents an offset on the $[u, v]^T_{NMS}$ and is coupled with the scores in the local $N \times N$ window, making it differentiable in this window. Thus, the overall module is technically partially differentiable. In back-propagation, the gradient would flow to the scores in the local window through the second term, and thus optimizing the output keypoint position $[u, v]^T_{soft}$ is equivalent to optimizing the scores in the local window. Because there are many keypoints, the score map is sparsely optimized within individual local windows. This is similar to the sampling process in reinforced methods [37], [38], except that gradients can flow back into the scores in these windows.

C. Learning accurate keypoints

Accurate keypoints should be located precisely at repeatable locations (such as corners). For this purpose, we present the reprojection loss and dispersity peak loss. The reprojection loss directly optimizes the position of keypoints, and the dispersity peak loss ensures that the score is maximal at the keypoint position, resulting in accurate keypoints.

For two images $\boldsymbol{I}_A$ and $\boldsymbol{I}_B$, the network estimates their score maps $\boldsymbol{S}_A$ and $\boldsymbol{S}_B$, and the DKD module extracts the keypoints $\boldsymbol{p}_A$ and $\boldsymbol{p}_B$. Then $\boldsymbol{p}_A$ are warped to image $\boldsymbol{I}_B$ with

$$\boldsymbol{p}_{AB} = \mathbf{warp}_{AB}(\boldsymbol{p}_A), \tag{7}$$

where $\mathbf{warp}_{AB}$ can be any differentiable warp function projecting keypoints from image A to image B, such as the homography projection, 3D perspective projection, or even the optical flow. The homography projection is given as

$$[\boldsymbol{p}^T_{AB}, 1]^T = \boldsymbol{H}_{AB}[\boldsymbol{p}^T_A, 1]^T, \tag{8}$$

where $\boldsymbol{H}_{AB}$ denotes the 3×3 homography matrix. And the 3D perspective projection is

$$\boldsymbol{p}_{AB} = \pi(d_A \boldsymbol{R}_{AB} \pi^{-1}(\boldsymbol{p}_A) + \boldsymbol{t}_{AB}), \tag{9}$$

where $\pi(\boldsymbol{P}) = \boldsymbol{K}\boldsymbol{P}/Z$ projects a 3D point $\boldsymbol{P} = [X, Y, Z]^T$ in camera coordinate system to pixel coordinates on image plane with the camera intrinsics $\boldsymbol{K} \in \mathbb{R}^{2\times3}$; $\boldsymbol{R}_{AB}$ and $\boldsymbol{t}_{AB}$ are the rotation and translation of 3D points from image A to B, respectively; and d_A denotes the depth of keypoint $\boldsymbol{p}_A$.

1) Reprojection loss: For a warped keypoint $\boldsymbol{p}_{AB}$, we find its closest detected keypoint $\boldsymbol{p}_B$ within th_{gt} pixels distance as its corresponding keypoint. Then the reprojection distance of $\boldsymbol{p}_{AB}$ and $\boldsymbol{p}_B$ is defined as

$$\text{dist}_{AB} = \left\| \boldsymbol{p}_{AB} - \boldsymbol{p}_B \right\|_p, \tag{10}$$

where p is the norm factor. Inspired by symmetric epipolar distance, the reprojection loss is given in a symmetric way

$$\mathcal{L}_{rp} = \frac{1}{2}(\text{dist}_{AB} + \text{dist}_{BA}). \tag{11}$$

As shown in Fig. 4, the minimization of reprojection loss pulls the warped keypoint and its corresponding keypoints together by adjusting their position. Since the keypoints are extracted from the score map using the differentiable DKD, the keypoint position is adjusted in a single optimization step by optimizing the scores within the local $N \times N$ window corresponding to individual keypoint. It implicitly provides keypoint repeatability [42], because the keypoints in non-repeatable areas would result in large reprojection errors. Therefore, other similarity measurements such as score differences [34], the cosine similarity [13] or the Kullbackleibler divergence [20] are not necessary any more.

2) Dispersity peak loss: The minimization of reprojection loss optimizes the scores in the local window through the soft term $[\hat{i}, \hat{j}]^T_{soft}$ of equation (6). However, the gradient step to improve $[\hat{i}, \hat{j}]^T_{soft}$ might affect the $[u, v]^T_{NMS}$. To align their optimization directions, we regularize the scores in the local window to be "peaky": that is, it should have a high score at the keypoint and low scores around it in the local window. In such a case, even if the local window centered on $[u, v]^T_{NMS}$ is slightly shifted, it still contains the keypoint, and $[\hat{i}, \hat{j}]^T_{soft}$ will be regulated to a new soft offset *w.r.t.* the new local window center so that the keypoint position remains stable. In previous works [13], [20], this property is regularized by the difference

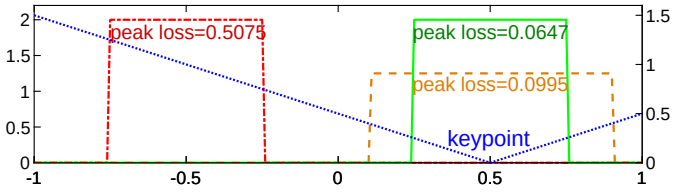


Fig. 5. A toy example of one-denominational dispersity peak loss. In this example, the window range is $[-1, 1]$, and the keypoint is at $x = 0.5$. Three example rectangular score distributions are shown by the red dot-dash, green solid, and brown dash curves in the window. Their range is represented by the y-axis on the left. The blue dot curve is the distance to keypoint. Its range is shown on the right y-axis. The statistical characteristics (max-mean) of the red dot-dash and green solid distributions are identical, but they have different dispersity peak losses because of different spatial distribution.

between the maximum and average scores. However, it is only the statistical properties of the score patch, neglecting the spatial distribution of the scores. To force the score patch to be "peaky" exactly at the keypoint, we propose the score dispersity peak loss (Fig.5). It takes the spatial distribution of scores into account, resulting in higher scores at the keypoint and lower scores further away.

Considering a $N \times N$ score patch, the distance of each pixel $[i, j]^T$ in the patch to the soft detected keypoint $[\hat{i}, \hat{j}]^T_{soft}$ is

$$d(i,j) = \left\{ \left\| [i,j] - [\hat{i},\hat{j}]_{soft} \right\|_p \mid 0 \le i,j < N \right\}. \tag{12}$$

The dispersity peak loss of this patch is then defined as

$$\mathcal{L}_{pk} = \frac{1}{N^2} \sum_{0 \le i,j < N} d(i,j) s'(i,j), \tag{13}$$

where s' is the softmax score in equation (4).

D. Learning discriminative descriptor

Descriptors of the same keypoints (in different images) should be identical, whereas descriptors of different keypoints should be distinct. This is known as descriptor discriminativeness, and it is trained using the triplet loss [10]–[12], [14], [15], [20], [33]. However, the triplet loss only optimizes the sparse descriptors sampled from keypoints, so the dense descriptor map cannot be fully constrained (For descriptor of $\boldsymbol{p}_A$ in Fig.6(a), the triplet loss only uses the descriptors of two points in Fig.6(b): the descriptors of the corresponding point $\boldsymbol{p}_{AB}$ and the hardest negative point.). To address this issue, we adopt the NRE loss [29] to train with the dense descriptor map. It minimizes the cross entropy difference between the dense reprojection probability map and the dense matching probability map (Fig. 6(c) and (d)), thereby providing a comprehensive constraint for the dense descriptor map as well as a stable training process (section IV-D2).

As Fig. 6, for a keypoint $\boldsymbol{p}_A$ in image A, and its reprojected keypoint $\boldsymbol{p}_{AB}$ in image B, the reprojection probability map (Fig. 6 (c)) is defined by $\boldsymbol{p}_{AB}$ with the bilinear interpolation:

$$\begin{aligned} q_r(\boldsymbol{p}_B|\mathbf{warp}_{AB}, \boldsymbol{p}_A) := {} & w_{00}[\![\boldsymbol{p}_B = \lfloor \boldsymbol{p}_{AB} \rfloor]\!] + \\ & w_{01}[\![\boldsymbol{p}_B = \lfloor \boldsymbol{p}_{AB} \rfloor + [0,1]^T]\!] + \\ & w_{10}[\![\boldsymbol{p}_B = \lfloor \boldsymbol{p}_{AB} \rfloor + [1,0]^T]\!] + \\ & w_{11}[\![\boldsymbol{p}_B = \lfloor \boldsymbol{p}_{AB} \rfloor + [1,1]^T]\!], \end{aligned} \tag{14}$$

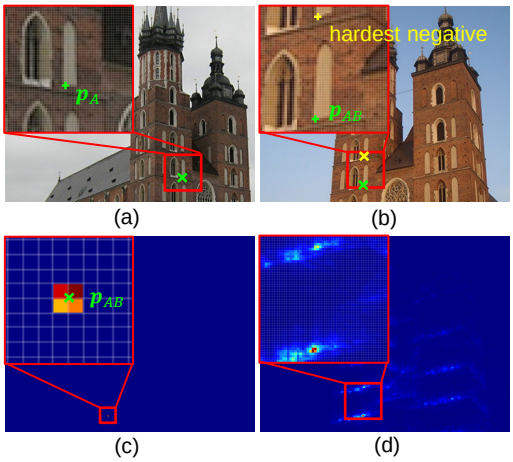


Fig. 6. The illustration of triplet loss and neural reprojection loss. (a) the source image A, (b) the target image B, (c) the reprojection prabability map, (d) the matching probability map. A corresponding keypoint pair $\boldsymbol{p}_A$ and $\boldsymbol{p}_{AB}$ in image A and B are marked as green crosses. The yellow cross in image B is the position of the hardest negative descriptor of $\boldsymbol{p}_A$. The triplet loss of $\boldsymbol{p}_A$ is only constrained by $\boldsymbol{p}_{AB}$ and its hardest negative. While the NRE loss [29] minimizes the difference between the reprojection probability map and the matching probability map, connecting $\boldsymbol{p}_A$ to the dense descriptor map $\boldsymbol{D}_B$ and providing a more comprehensive constraint.

where $[\![\cdot]\!]$ is the Iverson bracket ($[\![True]\!]$=1 and $[\![False]\!]$=0), $\lfloor \cdot \rfloor$ denotes the floor function, and w_{ij} are coefficients of bilinear interpolation. Following [29], the inconsistent warped keypoints, *e.g.* those outside the image, are marked as outlier ($\boldsymbol{out}$) with probability $q_r(\boldsymbol{p}_B|\mathbf{warp}_{AB}(\boldsymbol{p}_A) = \boldsymbol{out}) = 1$.

Besides, for the descriptor $\boldsymbol{d}_{\boldsymbol{p}_A} \in \mathbb{R}^{dim}$ of keypoint $\boldsymbol{p}_A$ and the dense descriptor map $\boldsymbol{D}_B \in \mathbb{R}^{(H \times W) \times dim}$, their similarity map $\boldsymbol{C}_{\boldsymbol{d}_{\boldsymbol{p}_A}, B} \in \mathbb{R}^{H \times W}$ is given as

$$\boldsymbol{C}_{\boldsymbol{d}_{\boldsymbol{p}_A}, \boldsymbol{D}_B} = \boldsymbol{D}_B \boldsymbol{d}_{\boldsymbol{p}_A}. \tag{15}$$

Again, an outlier category $\boldsymbol{out}$ is added to handle the case when the descriptor $\boldsymbol{d}_{\boldsymbol{p}_A}$ has no corresponding descriptor in the descriptor map $\boldsymbol{D}_B$. We denote the similarity map with outlier as $\overline{\boldsymbol{C}}_{\boldsymbol{d}_{\boldsymbol{p}_A}, \boldsymbol{D}_B} := \{\boldsymbol{C}_{\boldsymbol{d}_{\boldsymbol{p}_A}, \boldsymbol{D}_B}, \boldsymbol{out}\}$, thus $|\overline{\boldsymbol{C}}_{\boldsymbol{d}_{\boldsymbol{p}_A}, \boldsymbol{D}_B}| = H \times W + 1$. Now, the matching probability map (Fig. 6 (d)) is given as the softmax normalization of the similarity map

$$q_m(\boldsymbol{p}_B|\boldsymbol{d}_{\boldsymbol{p}_A}, \boldsymbol{D}_B) := \mathbf{softmax}\left(\frac{\overline{\boldsymbol{C}}_{\boldsymbol{d}_{\boldsymbol{p}_A}, \boldsymbol{D}_B} - 1}{t_{des}} \right), \tag{16}$$

where t_{des} controls the sharpness of matching probability map.

The NRE loss [29] minimizes the difference of the reprojection probability map and the matching probability map with cross-entropy (CE):

$$\begin{aligned} & NRE(\boldsymbol{p}_A, \mathbf{warp}_{AB}, \boldsymbol{d}_{\boldsymbol{p}_A}, \boldsymbol{D}_B) \\ & := CE\left(q_r(\boldsymbol{p}_B|\mathbf{warp}_{AB}, \boldsymbol{p}_A) \| q_m(\boldsymbol{p}_B|\boldsymbol{d}_{\boldsymbol{p}_A}, \boldsymbol{D}_B)\right) \\ & = -\sum_{\boldsymbol{p}_B \in \{\boldsymbol{I}_B, \boldsymbol{out}\}} q_r(\boldsymbol{p}_B|\boldsymbol{p}_{AB}) \ln\left(q_m(\boldsymbol{p}_B|\boldsymbol{d}_{\boldsymbol{p}_A}, \boldsymbol{D}_B)\right) \\ & = -\ln\left(q_m(\boldsymbol{p}_{AB}|\boldsymbol{d}_{\boldsymbol{p}_A}, \boldsymbol{D}_B)\right). \end{aligned} \tag{17}$$

Hence, we define the descriptor loss in a symmetric way as

$$\mathcal{L}_{de} = \frac{1}{N_A + N_B} * \\ \Big(\sum_{\boldsymbol{p}_A \in \boldsymbol{I}_A} NRE(\boldsymbol{p}_A, \mathbf{warp}_{AB}, \boldsymbol{d}_{\boldsymbol{p}_A}, \boldsymbol{D}_B) + \\ \sum_{\boldsymbol{p}_B \in \boldsymbol{I}_B} NRE(\boldsymbol{p}_B, \mathbf{warp}_{BA}, \boldsymbol{d}_{\boldsymbol{p}_B}, \boldsymbol{D}_A) \Big), \tag{18}$$

where N_A and N_B are the number of keypoints in image A and B, respectively.

E. Learning reliable keypoint

The reprojection and dispersity peak loss provide accurate and repeatable keypoints. However, the spatial properties of descriptor map are not taken into account, so the keypoints might be unreliable [13], *e.g.*, the keypoints could locate in non-discriminative low-texture areas. To address this issue, we introduce a reliability loss based on the matching probability map in the NRE loss [29].

First, the matching probability map is obtained by normalizing the similarity map $\boldsymbol{C}_{\boldsymbol{d}_{\boldsymbol{p}_A}, \boldsymbol{D}_B} \in \mathbb{R}^{H \times W}$ in equation (15)

$$\widetilde{\boldsymbol{C}}_{\boldsymbol{d}_{\boldsymbol{p}_A}, \boldsymbol{D}_B} = \exp\left(\frac{\boldsymbol{C}_{\boldsymbol{d}_{\boldsymbol{p}_A}, \boldsymbol{D}_B} - 1}{t_{rel}} \right), \tag{19}$$

where t_{rel} controls the sharpness. Then the reliability of keypoint $\boldsymbol{p}_A$ is defined as

$$r_{\boldsymbol{p}_A} = \mathbf{bisampling}\left(\widetilde{\boldsymbol{C}}_{\boldsymbol{d}_{\boldsymbol{p}_A}, \boldsymbol{D}_B}, \ \boldsymbol{p}_{AB} \right), \tag{20}$$

where $\mathbf{bisampling}(\boldsymbol{M}, \boldsymbol{p})$ is the bilinear sampling at position $\boldsymbol{p} \in \mathbb{R}^2$ on probability map $\boldsymbol{M} \in \mathbb{R}^{H \times W}$.

Intuitively, $r_{\boldsymbol{p}_A}$ assesses the matching quality of $\boldsymbol{p}_A$. If $\boldsymbol{p}_A$ is in unreliable low texture or repetitive region, the overall similarities in that region will be higher. As a result, the normalized similarity map $\widetilde{\boldsymbol{C}}_{\boldsymbol{d}_{\boldsymbol{p}_A}, \boldsymbol{D}_B}$ has lower values, and the sampled score $r_{\boldsymbol{p}_A}$ is small, indicating that $\boldsymbol{p}_A$ is unreliable.

Considering all valid keypoints in image A, we define their reliability loss similar to D2Net [14] and ASLFeat [15]

$$\mathcal{L}^A_{reliability} = \frac{1}{N_A} \sum_{\substack{\boldsymbol{p}_A \in \boldsymbol{I}_A, \\ \boldsymbol{p}_{AB} \in \boldsymbol{I}_B}} \frac{s_{\boldsymbol{p}_A} s_{\boldsymbol{p}_{AB}}}{\sum_{\substack{\boldsymbol{p}'_A \in \boldsymbol{I}_A, \\ \boldsymbol{p}'_{AB} \in \boldsymbol{I}_B}} s_{\boldsymbol{p}'_A} s_{\boldsymbol{p}'_{AB}}} (1 - r_{\boldsymbol{p}_A}), \tag{21}$$

where N_A is the number of keypoints in the image A. For each keypoint $\boldsymbol{p}_A$ in image A, $\boldsymbol{p}_{AB}$ is its corresponding projection keypoint in image B (equation (7)). And $s_{\boldsymbol{p}}$ denotes the score value of keypoint $\boldsymbol{p}$ in its corresponding image. Similarly, the reliability loss is also given in a symmetric way as

$$\mathcal{L}_{rl} = \frac{1}{2}(\mathcal{L}^A_{reliability} + \mathcal{L}^B_{reliability}). \tag{22}$$

IV. Experiments

In this section, we first introduce the datasets, training details, and the evaluation metrics. To analyze the proposed method, ablation studies are conducted on network architecture and loss terms. At last, we reported the comparison results with the state-of-the-art methods on homography estimation, camera pose estimation, and visual (re-)localization tasks.

TABLE I
THE NETWORK CONFIGURATIONS. THE "c_i" DENOTES THE CHANNEL NUMBERS OF THE i BLOCK, AND "N_{head}" IS THE NUMBER OF LAYERS IN FEATURE AGGREGATION HEAD. THE "MP" IS THE NUMBER OF PARAMETERS IN MILLIONS. AND THE "GFLOPS" DENOTES THE GIGA FLOATING-POINT OPERATIONS OF THE NETWORK FOR 640×480 IMAGE.

Models	c_1	c_2	c_3	c_4	dim	N_{head}	*MP*	*GFLOPs*
Tiny	8	16	32	64	64	1	0.080	2.109
Small	8	16	48	96	96	1	0.142	3.893
Normal	16	32	64	128	128	1	0.318	7.909
Large	32	64	128	128	128	2	0.653	19.685

A. Datasets

MegaDepth [43] dataset includes tourist photos on famous sites and 3D maps built by COLMAP [44]. It provides dense depth and camera pose for each image, which allow us to establish dense correspondences between images. We adopt the image pairs generated in DISK [37] to train our model. With co-visibility heuristics, it generates image pairs from the scenes except those overlapped with IMW2020 [45] validation and test sets, resulting 135 scenes with 63k images in total.

HPatches [46] dataset contains planar images of 57 illumination and 59 viewpoint scenes. Each scene has 5 image pairs with ground truth homography matrices. Following D2Net [14], eight unreliable scenes are excluded. We conducted the ablation studies and homography estimation on this dataset.

IMW2020 [45] is also built with tourist photos using COLMAP [44]. It provides a standard pipeline for camera pose estimation, which we used to compare the proposed method to existing methods.

Aachen Day-Night [47] dataset allows us to evaluate the effectiveness of descriptors on visual (re-)localization task. It tries to localize 98 query images captured in night-time based on a 3D model pre-built with day-time images.

B. Training details

1) Details of loss calculation: We used the DKD with a window size of $N = 5$ to detect 400 keypoints and randomly sampled another 400 keypoints on non-salient positions. In reprojection loss, the $th_{gt} = 5$ and the normal factor $p = 1$. The overall loss is

$$\mathcal{L} = w_{rp}\mathcal{L}_{rp} + w_{pk}\mathcal{L}_{pk} + w_{rl}\mathcal{L}_{rl} + w_{de}\mathcal{L}_{de}. \tag{23}$$

where $w_{rp} = 1$, $w_{pk} = 1$, $w_{rl} = 1$, and $w_{de} = 5$ in our experiments. And we set the normalization temperatures as $t_{det} = 0.1$, $t_{rel} = 1$, and $t_{des} = 0.02$.

2) Training setups: The images were cropped and resized to 480×480 in the training. The network was trained using the ADAM optimizer [48], with the learning rate starting at zero and warming up to $3e^{-3}$ in 500 steps before remaining at $3e^{-3}$. We set the batch size to one, but accumulated the gradient over 16 batches. Under these settings, the proposed model converges on NVIDIA Titan RTX in about two days.

C. Evaluation metrics

Assuming that image A and image B have N'_A and N'_B co-visible keypoints, respectively, the number of co-visible

TABLE II
ABLATION STUDIES OF NETWORK ARCHITECTURES.

Models	*Rep*	*MS*	*MMA@3*	*MHA@3*
Tiny	56.19%	32.54%	63.92%	70.56%
Small	58.25%	34.96%	67.61%	73.52%
Normal	54.93%	38.58%	70.78%	75.74%
Large	53.70%	39.80%	70.50%	76.85%

TABLE III
ABLATION STUDIES OF LOSSES. "RP", "PK", "RL", "DE", AND "TRI" DENOTE REPROJECTION LOSS, DISPERSITY PEAK LOSS, RELIABILITY LOSS, DESCRIPTOR LOSS, AND TRIPLET LOSS, RESPECTIVELY.

RP	*PK*	*RL*	*DE*	*Rep*	*MS*	*MMA@3*	*MHA@3*
		✓	✓	59.26%	34.55%	71.83%	67.22%
	✓	✓	✓	48.75%	30.88%	62.95%	72.59%
✓		✓	✓	**65.77%**	38.12%	**74.44%**	70.74%
✓	✓		✓	51.11%	33.77%	65.23%	74.44%
✓	✓	✓	✓	54.93%	**38.58%**	70.78%	**75.74%**
✓	✓	✓	*Tri*	57.91%	14.13%	39.38%	60.74%

keypoints is defined as $N_{cov} = (N'_A + N'_B)/2$. Among them, we get N_{gt} ground truth keypoint pairs with the reprojection distance less than three pixels. On the other hand, $N_{putative}$ matches are obtained with mutual matching of the descriptors. And N_{inlier} inlier matches are acquired by assessing the reprojection distance within different pixels threshold. Following previous works [12], [46], we adopt the following metrics:

1) **Repeatability** of keypoints is given as $Rep = N_{gt}/N_{cov}$.
2) **Matching Score** is given as $MS = N_{inlier}/N_{cov}$.
3) **Mean Matching Accuracy** (MMA) denotes the mean of matching accuracy of all image pairs, it is given as $N_{inlier}/N_{putative}$.
4) **Mean Homography Accuracy** (MHA) is the mean of homograpy accuracy of all image pairs, it is defined as the percentage of correct image corners with the estimated homography matrix.

D. Ablation Studies

We conduct ablation studies on HPatches dataset [46] with single-scale. The score threshold th in the DKD module is 0.2. Up to 5000 keypoints are detected. We evaluate the $MMA@3$ and $MHA@3$ without loss of generality.

1) Ablation studies on network architecture: Lightweight networks can improve running efficiency, but they have worse performance. To select a network that can run in real-time without significant performance degradation, we investigate four network variations (Table I). As in Table II, the matching score, matching accuracy, and homography accuracy of the "Normal" network increased by about 6% compared with the "Tiny" network. However, despite doubling in complexity, the "Large" network only improves about 1% compared with the "Normal" one, with diminishing marginal improvement. So, we use the "Normal" network to keep real-time performance.

2) Ablation studies on loss functions: To investigate the loss functions, we train the "Normal" network with different loss configurations (Table III).

The reprojection loss directly optimizes keypoints position and should produce accurate keypoints. In Fig. 7, the visual

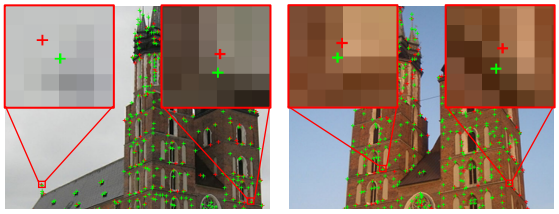


Fig. 7. The effect of reprojection loss. The sub-pixel keypoints from the network trained with and without reprojection loss are denoted by green and red crosses, respectively.

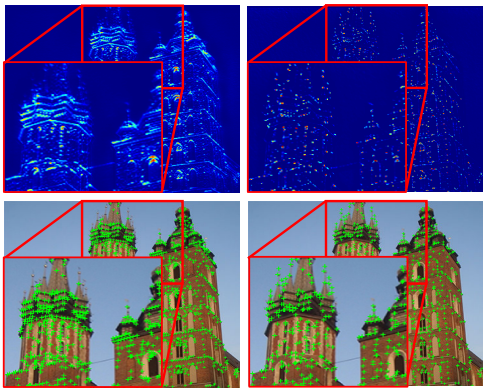


Fig. 8. The effect of dispersity peak loss. The score maps and keypoints detected from these score maps are shown in the first and second row.

comparisons of keypoints from networks trained with and without reprojection loss validates this anticipation: the keypoints of network trained with reprojection loss (green crosses) are more accurate than without reprojection loss (red crosses). Table III shows the quantitative comparisons: the reprojection loss increases all metrics of the fifth row compared to the second, and the third row compared to the first.

The dispersity peak loss regularizes the shape of score patches. Table III illustrates an intriguing phenomenon: the dispersity peak loss improves homography accuracy but decreases repeatability and matching accuracy (the second row compared to the first, and the fifth row compared to the third). To investigate this phenomenon, Fig. 8 visualizes the score map and keypoints of network trained with and without the dispersity peak. With the dispersity peak loss, the score map becomes "peaky", which improves localization certainty and homography accuracy. While without the dispersity peak loss, the score map is less "peaky", keypoints are more likely to be crowded together. It increases the probability of finding a match in the crowd, so it is advantageous to repeatability and matching accuracy. As the accuracy of downstream tasks (homography estimation in this case) is more important, we retain the dispersity peak loss as the score map regularization.

The descriptor loss were studied by comparing the NRE loss [29] and triplet loss [33]. For triplet loss, negative descriptor is mined from the keypoints (Fig. 6(b)), and the triplet margin is 0.5. Fig. 9 shows the training process of triplet loss with sub-pixel keypoints. It falls into a local minimum where the loss approaches but dose not cross the triplet margin (all the descriptors become the same). So its performance are

TABLE IV
THE NUMBER OF NETWORK PARAMETERS, GFLOPS AND INFERENCE FPS ON 640 × 480 IMAGES; THE MMA AND MHA WITHIN ONE TO THREE PIXELS THRESHOLDS OF DIFFERENT METHODS ON HPATCHES [46] DATASET. THE TOP THREE RESULTS ARE MARKED WITH **RED**, **GREEN**, AND **BLUE**.

Models	*Params/M*	*GFLOPs*	*FPS*	*MMA@1*	*MMA@2*	*MMA@3*	*MHA@1*	*MHA@2*	*MHA@3*
D2-Net(MS) [14]	7.635	889.40	6.60	9.78%	23.52%	37.29%	5.19%	21.30%	38.33%
LF-Net(MS) [17]	2.642	**24.37**	29.88	19.94%	41.98%	55.60%	17.41%	42.41%	57.78%
SuperPoint [12]	1.301	26.11	**45.87**	34.27%	54.94%	65.37%	35.00%	58.33%	70.19%
R2D2(MS) [13]	**0.484**	464.55	8.70	33.31%	62.17%	**75.77%**	35.74%	59.44%	71.48%
ASLFeat(MS) [15]	0.823	44.24	8.96	39.16%	61.07%	72.44%	37.22%	61.67%	73.52%
DISK [37]	1.092	98.97	15.73	43.71%	**66.98%**	**77.59%**	34.07%	57.59%	70.56%
ALIKE-N	**0.318**	**7.91**	**95.19**	43.52%	63.14%	70.78%	42.04%	**62.78%**	75.74%
ALIKE-L	**0.653**	**19.68**	**68.18**	**43.90%**	63.11%	70.50%	**45.00%**	**65.93%**	**76.85%**
ALIKE-N(MS)	**0.318**	25.97	41.48	**44.06%**	**65.56%**	74.05%	**44.07%**	**65.37%**	**75.93%**
ALIKE-L(MS)	**0.653**	64.63	29.15	**44.97%**	**66.21%**	**74.51%**	**45.00%**	**65.37%**	**76.48%**

TABLE V
POSE ESTIMATION RESULTS ON IMW2020 [45] TEST SETS (UP TO 2048 KEYPOINTS PER IMAGE). THE GFLOPS AND PERFORMANCE PER COST (PPC) ARE REPORTED. FOR STEREO TASK, WE REPORT NUMBER OF FEATURES (NF), REP, MS, MAA(5°) AND MAA(10°). FOR MULTIVIEW TASK, THE NUMBER OF MATCHES (NM), NUMBER OF 3D LANDMARKS (NL), TRACK LENGTH OF LANDMARK (TL), MAA(5°) AND MAA(10°) ARE REPORTED. THE TOP THREE RESULTS ARE MARKED WITH **RED**, **GREEN**, AND **BLUE**.

Methods	*GFLOPs*	*Stereo*						*Multiview*					
		NF	*Rep*	*MS*	*mAA(5°)*	*mAA(10°)*	*PPC*	*NM*	*NL*	*TL*	*mAA(5°)*	*mAA(10°)*	*PPC*
D2-Net(MS) [14]	889.40	2046	16.80%	29.30%	6.06%	12.27%	0.0138	2045.60	1999.37	3.01	17.77%	28.30%	0.0318
LF-Net(MS)* [17]	**24.37**	—	—	—	—	23.44%	0.9620	196.70	1385.00	4.14	—	51.41%	**2.1099**
SuperPoint [12]	26.11	2048	36.40%	63.00%	19.71%	28.97%	**1.1093**	2048.00	1185.38	4.33	44.35%	54.66%	2.0935
R2D2(MS) [13]	464.55	2048	42.90%	74.60%	27.20%	39.02%	0.0840	2048.00	1225.85	4.28	53.13%	64.03%	0.1378
ASLFeat(MS) [15]	77.58	2043	**43.10%**	74.90%	22.62%	33.65%	0.4337	157.51	1106.59	4.42	45.28%	55.61%	0.7168
DISK [37]	98.97	2048	**44.80%**	**85.20%**	**38.72%**	**51.22%**	0.5175	526.35	2424.80	5.50	**63.25%**	**72.96%**	0.7372
ALike-N	**7.91**	1803	**43.30%**	**81.10%**	**35.12%**	**47.18%**	**5.9652**	276.48	1644.20	4.97	**59.18%**	**69.21%**	**8.7507**
ALike-L	**19.68**	1771	42.90%	**82.20%**	**37.24%**	**49.58%**	**2.5187**	298.30	1693.31	5.02	**60.30%**	**70.22%**	**3.5673**

* There are no public test results of LF-Net [17] on the benchmark website, and the test results came from DISK [37].

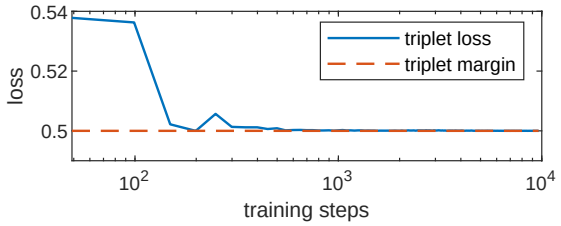


Fig. 9. The training process of the network with triplet loss.

extremely worse (the last row of Table III). We find it is tricky to tune the hyperpameters for triplet loss, so we adopt the NRE loss [29] as it has a stable convergence, although it requires more GPU memory and training time because of dense similarity map computation in equation (15).

The reliability loss ensures that the keypoints are located at areas where the descriptors are discriminative. More reliable keypoints would produce fewer false matches, resulting in a higher matching score and accuracy, as shown in the fourth and fifth row of Table III.

E. Comparisons with the state-of-the-arts

The proposed methods are compared to state-of-the-arts on homography estimation, camera pose estimation, and visual (re-)localization tasks. And we denote the proposed methods as "ALIKE-[T/S/N/L][(MS)]", where "[T/S/N/L]" is model size (Table I), and "MS" is multi-scale keypoint detection.

1) The network complexity: Table IV reports the complexity of different methods, including the number of parameters, GFLOPs (Giga FLoating-point OPerations), and the inference FPS. As can be seen, the ALIKE-N has the fewest parameters (318K), followed by R2D2 [13] (484K) and ALIKE-L (653K). However, a network with fewer parameters does not mean it has less computation due to non-parametric operations. For example, R2D2 [13] has 464.55 GFLOPs computational cost despite having only 484K parameters. While ALIKE-N (318K) and ALIKE-L (653K) only have it for 7.91 and 19.68 GFLOPs, respectively. For more intuitive comparisons, the inference FPS are also reported in Table IV. The ALIKE-N, ALIKE-L, and SuperPoint [12] are the fastest, can run at 95.19, 68.18, and 45.87 FPS, respectively. Considering the matching performance, the proposed methods have a short inference time while also provide accurate transformation estimation.

2) Homography estimation: Following previous works [13]–[15], we conduct homography estimation on HPatches dataset [46]. Previous works focus on the matching accuracy (MMA), but we believe it is only an intermediate metric, as the goal of keypoints matching is downstream homography estimation. For better homography estimation accuracy, the MMA at stricter thresholds and the homography accuracy (MHA) is more important. Therefore, we report MMA and MHA at stricter thresholds in Table IV. Due to the DKD module and proposed losses, the proposed methods have much better MMA at stricter thresholds than previous works. This indicates that accurate keypoints are obtained with the proposed

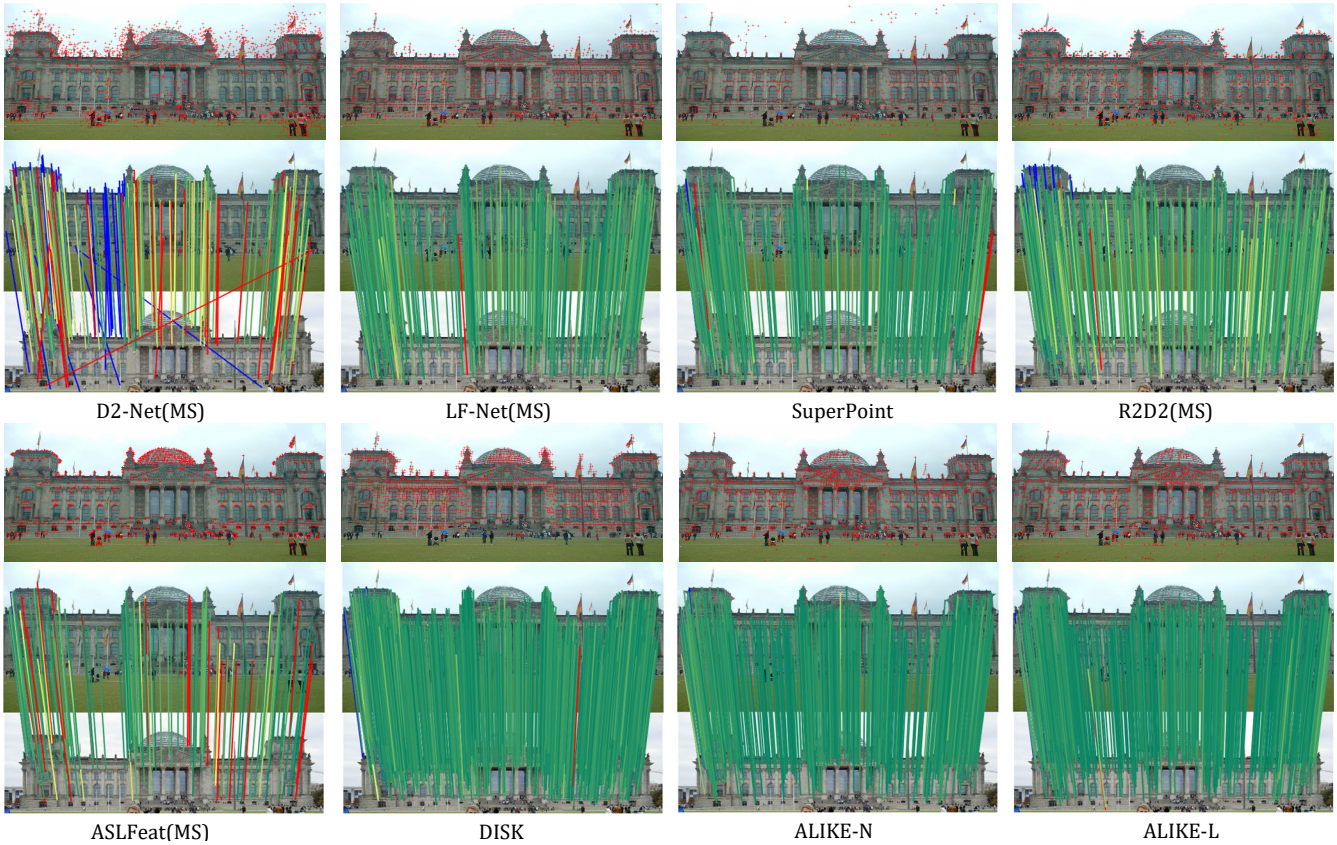


Fig. 10. The visualization of detected keypoints (the first and third row) and the matches (the second and fourth row) on IMW2020 [45] validation set. In the the second and fourth row, the inliers are plotted from green to yellow if they are correct (0 to 5 pixels in reprojection error), in red if they are incorrect (above 5 pixels), and in blue if ground truth depth is not available. Best viewed in color and zoom in 400%.

method. More importantly, for homography accuracy MHA, the proposed methods outperform SOTA methods. Compared to ASLFeat(MS) [15], the method with the highest MHA@3, the proposed ALIKE-N and ALIKE-L improve MHA@3 by 2.22% and 3.33%, respectively, while decreases computational complexity by about 5.6 and 2.2 times.

3) Camera pose estimation: The IMW2020 [1] [45] includes the stereo and multi-view tasks for pose estimation. For fair comparison, we used the best configuration for each method and detected up to 2048 key points using the built-in mutual nearest neighbor matching. The mean Average Accuracy (mAA) is obtained by integrating the translation and rotation vector errors to 5° and 10°. As real-time performance is also important for practical applications, the GFLOPs and performance per cost (PPC) [49], where $PPC = mAA(10°)/GFLOPs$, are also reported in Table V.

Table V illustrates the quantitative evaluation results. Without considering the PPC, the proposed ALIKE outperforms all previous methods except DISK [37]. However, DISK requires 98.97 GFLOPs of computational cost, whereas the ALIKE-L and ALIKE-N require only 7.91 and 19.68 GFLOPs, respectively. Considering the PPC, the ALIKE-N and ALIKE-L are approximately 10 and 5 times more efficient than DISK, respectively. In summary, the proposed ALIKE is both accurate and lightweight, making it ideal for real-time applications.

To examine each method, the detected keypoints and estimated matches are visualized in Fig. 10. The D2-Net [14] and ASLFeat [15] extract keypoints from low-resolution feature maps. So their keypoints are less accurate, resulting in some error matches (the red lines). For LF-Net [17], R2D2 [13], and DISK [37], many false keypoints are in the texture-less building boundary. Keypoints of R2D2 [13] tend to crammed together, whereas keypoints of DISK [37] are almost all on buildings. SuperPoint [12] generates the sparsest keypoints, and some of them are in unreliable sky. While the majority keypoints of the proposed methods are located at image corners and edges, and less incorrect matches are included.

4) Visual (re-)localization: The proposed method was tested on Aachen Day-Night visual (re-)localization benchmark[2] [47] with default configurations. It builds a 3D map based on image keypoints, registers query images to the map, and uses the percentage of correctly registered images in three error tolerances (*i.e.* $(0.25m, 2°)$/$(0.5m, 5°)$/$(5m, 10°)$) as evaluation metric. We report performance when using limited (up to 2048) and unlimited features in Table VI.

Except SEKD [20], most methods perform well when using unlimited number of keypoints. But practical applications typically use limited number of keypoints, D2-Net(MS) [14], ASLFeat [15], and DISK [37] degrade dramatically in such cases. ALIKE-L achieves superior performance with fewer

[1] https://www.cs.ubc.ca/research/image-matching-challenge/

[2] https://www.visuallocalization.net

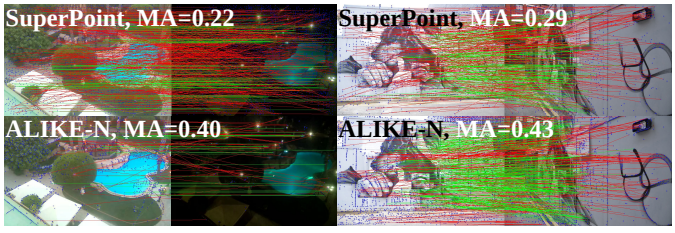
TABLE VI
VISUAL LOCALIZATION RESULTS ON AACHEN DAY-NIGHT [47] DATASET. PERCENTAGES OF LOCALIZED QUERIES WITHIN THREE TOLERANCES USING TOP-2048 AND UNLIMITED KEYPOINTS ARE REPORTED. THE TOP THREE RESULTS ARE MARKED AS **RED**, **GREEN**, AND **BLUE**.

Methods	2048 keypoints			unlimited keypoints		
	0.25m,2°	0.5m,5°	5m,10°	0.25m,2°	0.5m,5°	5m,10°
D2-Net(SS) [14]	**74.5**	**85.7**	**96.9**	**78.6**	**88.8**	**100.0**
D2-Net(MS) [14]	61.2	81.6	**94.9**	**79.6**	86.7	**100.0**
SEKD(SS) [17]	42.9	51.0	57.1	54.1	65.3	74.5
SEKD(MS) [17]	50.0	63.3	70.4	59.2	72.4	82.7
SuperPoint [12]	**72.4**	79.6	88.8	73.5	81.6	**93.9**
R2D2(MS) [13]	63.3	78.6	87.8	71.4	85.7	**99.0**
ASLFeat(SS) [15]	54.1	67.3	76.5	77.6	**88.8**	**100.0**
ASLFeat(MS) [15]	49.0	59.2	69.4	77.6	**90.8**	**100.0**
DISK [37]	**70.4**	82.7	**94.9**	**84.7**	**90.8**	**100.0**
ALIKE-N	68.4	**84.7**	**96.9**	77.6	**87.8**	**100.0**
ALIKE-L	**74.5**	**87.8**	**98.0**	**79.6**	86.7	**100.0**

keypoints even without heuristically fine-tuning configurations, demonstrating the required effectiveness in resource-constrained applications. Furthermore, the normal model (ALIKE-N) achieves comparable localization accuracy while using fewer computational resources.

F. Limitations of proposed method

We observed two kinds of failure cases in image matching using the proposed keypoint and descriptor: images with extreme illumination changes and large viewpoint differences (as shown in Fig. 11), which are also the two most difficult challenges in image matching task. Technically, our lightweight network utilizes shallow architecture (Table I) and will be limited to extracting more low-level descriptors, which sacrifices some presentation capabilities. As a consequence, matching performance in extreme scenarios will be challenging. However, in the randomly selected challenging cases, the matching accuracy (MA, number of correct matches / number of putative matches) of ALIKE-N is twice as high as SuperPoint [12], which has the highest FPS and PPC except ours but three times the GFLOPs of ALIKE-N (Fig. 11).



(a) (b)

Fig. 11. The visualization of matching results of SuperPoint [12] and ALIKE-N on (a) challenging illumination and (b) viewpoint image pairs. The keypoints are represented by blue crosses, while correct matches (with a reprojection error < 3 pixels) and incorrect matches are represented by green and red lines, respectively. In this figure, only the top-500 putative matches with the highest similarity are shown for clarity.

V. CONCLUSIONS

In this paper, we present the ALIKE, an end-to-end accurate and lightweight keypoint detection and descriptors extraction network. It uses the differentiable keypoint detection module to detect accurate sub-pixel keypoints. And the keypoints are then trained with proposed reprojection loss and dispersity peak loss. Besides the keypoints, the NRE loss are used to train discriminative descriptors and the reliability loss are presented to force reliable keypoints. Compared to state-of-the-art approaches, the proposed method achieves comparable or superior results on homography estimation, camera pose estimation, and visual (re-)localization tasks while taking significantly less time to run. The ALIKE-N and ALIKE-L can run at 95 FPS and 68 FPS, respectively, for 640×480 images on NVIDIA TITAN X (Pascal). Our future works include enhancing the reliability and accuracy of detected keypoints using high-level semantic information or reinforcement learning, as well as integrating the proposed model into practical SLAM and HDR applications.

REFERENCES

[1] C.-R. Huang, H.-P. Lee, and C.-S. Chen, "Shot change detection via local keypoint matching," *IEEE Transactions on Multimedia*, vol. 10, no. 6, pp. 1097–1108, 2008.
[2] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE transactions on robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.
[3] X. Yang, Z. Yuan, D. Zhu, C. Chi, K. Li, and C. Liao, "Robust and efficient rgb-d slam in dynamic environments," *IEEE Transactions on Multimedia*, pp. 1–1, 2020.
[4] F. Kou, Z. Wei, W. Chen, X. Wu, C. Wen, and Z. Li, "Intelligent detail enhancement for exposure fusion," *IEEE Transactions on Multimedia*, vol. 20, no. 2, pp. 484–495, 2017.
[5] J. Zheng, Z. Li, Z. Zhu, S. Wu, and S. Rahardja, "Hybrid patching for a sequence of differently exposed images with moving objects," *IEEE transactions on image processing*, vol. 22, no. 12, pp. 5190–5201, 2013.
[6] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
[7] 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.
[8] M. Karpushin, G. Valenzise, and F. Dufaux, "Keypoint detection in rgbd images based on an anisotropic scale space," *IEEE Transactions on Multimedia*, vol. 18, no. 9, pp. 1762–1771, 2016.
[9] Y. Tian, B. Fan, and F. Wu, "L2-Net: Deep Learning of Discriminative Patch Descriptor in Euclidean Space," in *2017 IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, Jul. 2017, pp. 6128–6136.
[10] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas, "Working hard to know your neighbor's margins: Local descriptor learning loss," in *Advances in Neural Information Processing Systems*, Jan. 2018.
[11] Y. Tian, X. Yu, B. Fan, F. Wu, H. Heijnen, and V. Balntas, "SOSNet: Second Order Similarity Regularization for Local Descriptor Learning," in *Conference on Computer Vision and Pattern Recognition*, Dec. 2019.
[12] 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.
[13] J. Revaud, P. Weinzaepfel, C. D. Souza, N. Pion, G. Csurka, Y. Cabon, and M. Humenberger, "R2D2: Repeatable and Reliable Detector and Descriptor," in *NeurIPS*, 2019, p. 12.
[14] 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 *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, Jun. 2019, pp. 8084–8093.
[15] Z. Luo, L. Zhou, X. Bai, H. Chen, J. Zhang, Y. Yao, S. Li, T. Fang, and L. Quan, "ASLFeat: Learning Local Features of Accurate Shape and Localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Apr. 2020.
[16] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "LIFT: Learned Invariant Feature Transform," in *European Conference on Computer Vision*, vol. 9910. Cham: Springer, 2016, pp. 467–483.

[17] Y. Ono, E. Trulls, P. Fua, and K. M. Yi, "LF-Net: Learning Local Features from Images," in *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc., 2018, pp. 6234–6244.

[18] 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.

[19] 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.

[20] Y. Song, L. Cai, J. Li, Y. Tian, and M. Li, "SEKD: Self-Evolving Keypoint Detection and Description," *arXiv:2006.05077 [cs]*, Jun. 2020.

[21] Y. Tian, V. Balntas, T. Ng, A. Barroso-Laguna, Y. Demiris, and K. Mikolajczyk, "D2D: Keypoint Extraction with Describe to Detect Approach," in *Proceedings of the Asian Conference on Computer Vision*, 2020.

[22] T.-Y. Yang, D.-K. Nguyen, H. Heijnen, and V. Balntas, "UR2KiD: Unifying Retrieval, Keypoint Detection, and Keypoint Description without Local Correspondence Supervision," *arXiv:2001.07252 [cs]*, Jan. 2020.

[23] O. Chapelle and M. Wu, "Gradient descent optimization of smoothed information retrieval metrics," *Information retrieval*, vol. 13, no. 3, pp. 216–235, 2010.

[24] X. Sun, B. Xiao, F. Wei, S. Liang, and Y. Wei, "Integral human pose regression," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 529–545.

[25] K. Gu, L. Yang, and A. Yao, "Removing the bias of integral pose regression," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 11 067–11 076.

[26] H. Germain, G. Bourmaud, and V. Lepetit, "Sparse-to-dense hypercolumn matching for long-term visual localization," in *2019 International Conference on 3D Vision (3DV)*. IEEE, 2019, pp. 513–523.

[27] H. Germain, G. Bourmaud, and V. Lepetit, "S2DNet: Learning Accurate Correspondences for Sparse-to-Dense Feature Matching," in *European Conference on Computer Vision*, Apr. 2020.

[28] Q. Wang, X. Zhou, B. Hariharan, and N. Snavely, "Learning feature descriptors using camera pose supervision," in *European Conference on Computer Vision*. Springer, 2020, pp. 757–774.

[29] H. Germain, V. Lepetit, and G. Bourmaud, "Neural reprojection error: Merging feature learning and camera pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2021, pp. 414–423.

[30] G. Droge, "Dual-mode dynamic window approach to robot navigation with convergence guarantees," *Journal of Control and Decision*, vol. 8, no. 2, pp. 77–88, 2021.

[31] Y. Zhang, J. Wang, S. Xu, X. Liu, and X. Zhang, "MLIFeat: Multi-level information fusion based deep local features," in *Proceedings of the Asian Conference on Computer Vision*, 2020.

[32] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg, "Matchnet: Unifying feature and metric learning for patch-based matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3279–3286.

[33] V. Balntas, E. Riba, D. Ponsa, and K. Mikolajczyk, "Learning local feature descriptors with triplets and shallow convolutional neural networks." in *BMVC*, vol. 1, 2016, p. 3.

[34] 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 International Conference on Computer Vision*, 2019, pp. 5836–5844.

[35] K. He, Y. Lu, and S. Sclaroff, "Local Descriptors Optimized for Average Precision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 596–605.

[36] A. Barroso-Laguna, Y. Verdie, B. Busam, and K. Mikolajczyk, "Hdd-net: Hybrid detector descriptor with mutual interactive learning," in *Proceedings of the Asian Conference on Computer Vision*, November 2020.

[37] M. J. Tyszkiewicz, P. Fua, and E. Trulls, "DISK: Learning local features with policy gradient," in *Neural IPS*, Jun. 2020.

[38] A. Bhowmik, S. Gumhold, C. Rother, and E. Brachmann, "Reinforced Feature Points: Optimizing Feature Detection and Description for a High-Level Task," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, Jun. 2020, pp. 4947–4956.

[39] A. Rana, G. Valenzise, and F. Dufaux, "Learning-based tone mapping operator for efficient image matching," *IEEE Transactions on Multimedia*, vol. 21, no. 1, pp. 256–268, 2019.

[40] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Icml*, 2010.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[42] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005.

[43] Z. Li and N. Snavely, "Megadepth: Learning single-view depth prediction from internet photos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2041–2050.

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

[45] 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*, vol. 129, no. 2, pp. 517–547, 2021.

[46] 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.

[47] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic *et al.*, "Benchmarking 6dof outdoor visual localization in changing conditions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8601–8610.

[48] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *ICLR*, 2015.

[49] Z. Mubariz, G. Sourav, M. Michael, K. Julian, F. David, M.-M. Klaus, and E. Shoaib, "VPR-Bench: An open-source visual place recognition evaluation framework with quantifiable viewpoint and appearance change," *International Journal of Computer Vision*, vol. 129, pp. 2136–2174, 2021.