This CVPR workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Multi-scale Local Implicit Keypoint Descriptor for Keypoint Matching

JongMin Lee
Seoul National University
sdrjseka96@naver.com

Eunhyeok Park
POSTECH
canusglow@gmail.com

Sungjoo Yoo
Seoul National University
sungjoo.yoo@gmail.com

Abstract

We investigate the potential of multi-scale descriptors which has been under-explored in the existing literature. At the pixel level, we propose utilizing both coarse and fine-grained descriptors and present a scale-aware method of negative sampling, which trains descriptors at different scales in a complementary manner, thereby improving their discriminative power. For sub-pixel level descriptors, we also propose adopting coordinate-based implicit modeling and learning the non-linearity of local descriptors on continuous-domain coordinates. Our experiments show that the proposed method achieves state-of-the-art performance on various tasks, i.e., image matching, relative pose estimation, and visual localization.

1. Introduction

Image matching, a key building block of many vision tasks such as SfM [29], visual localization [26], and visual odometry [32], is an algorithm that recognizes structural characteristics with a similarity between two given images. Previously, image matching algorithms were implemented on top of hand-crafted features such as SIFT [15] and ORB [25]. Recent studies have achieved higher accuracy by re-interpreting image matching in terms of deep learning.

A deep learning-based image matching algorithm generally consists of the following steps. First, given two input images, a trained network is executed for each image to obtain the keypoint locations (i.e., keypoint detection), the corresponding repeatability or reliability scores, and the descriptors sampled from dense feature maps using each keypoint's coordinate. After selecting the keypoint locations (often at pixel-level resolution) having the top-K highest scores on each of the two input images and their associated descriptors, each descriptor is compared with the descriptors of the other image. The keypoint matching proceeds by iteratively selecting the keypoint pairs having the highest descriptor similarity.

For accurate matching, the detector should be able to find keypoints under various conditions (e.g. viewpoints or illumination change) [5, 7, 20, 23, 28, 33, 37] and should be reliable for the purpose of local feature matching when using descriptor [8, 18, 23]. Also, the descriptor from the dense feature map should reflect the uniqueness of keypoint by considering the surrounding image features. This means that the descriptor of a specific keypoint should be distinguishable from those of the other keypoints.

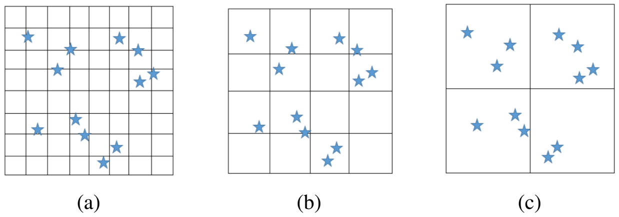


Figure 1. Keypoints are denoted with stars, and each grid represents the resolution of the feature map. As the resolution decreases, the number of keypoints per pixel increases, which makes the descriptors of adjacent keypoints similar to each other.

The descriptor feature map is obtained by convolution layers and the model architecture determines the size of feature map (1/N of input image size). For determining the size of this feature map, there is a trade-off between its discrimination ability and implementation cost. For instance, if a high resolution descriptor feature map is used, the descriptor from each pixel can have detailed information, which improves discrimination ability. However, it can incur high memory and computation costs due to high resolution [23, 36].

On the contrary, in the case of a low resolution descriptor feature map, due to a smaller number of feature vectors, the selection of positive and negative pairs becomes easier, which helps the training for enlarging a gap between positive and negative pairs. However, the descriptors of adjacent keypoints in the image tend to be similar to each other. It is because the descriptors of adjacent keypoints are affected by the same feature vector via bilinear interpolation (Figure 1). This can hurt the discrimination ability of the keypoint descriptor.

In many previous works [8,18,33], a single resolution of descriptor feature map (e.g., 1/4 of image size) is selected considering the above pros and cons of high and low resolutions, and the descriptor is obtained through a hand-crafted function, e.g., bilinear interpolation, at the pixel locations of the selected resolution.

We consider that the scale of the keypoint descriptor is under-explored and propose a multi-scale approach tackling both pixel and sub-pixel resolutions. In terms of pixel-level resolution, we propose a multi-scale descriptor in order to take advantage of both high and low resolutions in a U-net like architecture. Compared with the conventional single scale descriptor, our proposed multi-scale descriptor has a potential of offering better discrimination ability by making the best use of the CNN features with different receptive field sizes (e.g., large/small receptive fields for global/local features). In addition, we demonstrate the utility of sub-pixel resolution and propose a novel implicit model approach which can learn the embedding space of the descriptors, i.e., the non-linearity in the relationship between pixel-level descriptors and sub-pixel keypoint locations. Unlike conventional methods of bilinear interpolation for sub-pixel sampling, our proposed method is trainable thereby having a potential of giving more useful descriptors at a sub-pixel resolution.

Our contributions are summarized as follows.

- We investigate the potential of a multi-scale approach in keypoint descriptor and present a U-net-like model architecture.

- We propose a multi-scale descriptor consisting of coarse/fine-grained descriptors, in order to make the best use of the advantages of each scale. We also present a reliability score and scale-aware negative sampling for the multi-scale descriptor learning.

- We propose a sub-pixel descriptor function which, based on the existing coordinate-based feature generator, aims at learning the non-linearity of sub-pixel descriptor on the continuous coordinate space.

- We demonstrate the utility of the proposed methods in image matching [1], relative pose estimation [6, 13], and visual localization [26].

2. Related works

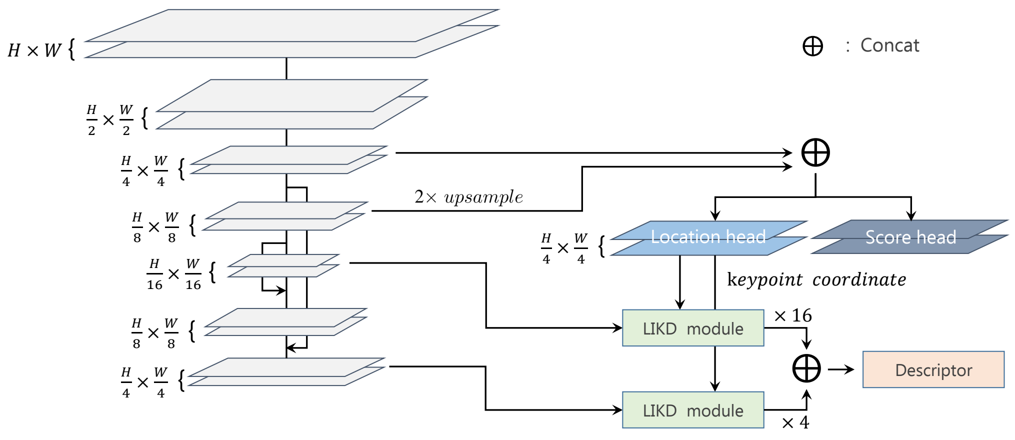
Multi-scale descriptor Recent methods such as [8,18,23] perform multi-scale estimation, where they resize the input image recursively and run the model multiple times to become invariant on scale changes. From multiple inference outputs, the keypoints and their descriptors are filtered by the keypoint scores. Another recent method [38] uses a coarse-to-fine architecture model, which is similar to our proposed one. It adopts a differentiable matching layer where a coarse feature map is used to alleviate computational cost in training to obtain keypoint locations. However, the method uses multi-descriptors differently in training time and inference time, which can lead to sub-optimal evaluation results. In our paper, we not only use a novel multi-scale descriptor, but also propose a new training pipeline to exploit the benefit of each scale for higher performance.

Learned Local Descriptors Recently, hand-crafted keypoint detectors and descriptors [2,15,24,25] have been outperformed by deep learning-based models by a large margin in several vision tasks. Early works in learned descriptor follow the classic vision descriptor method by running a CNN on a small patch of the image around the detected keypoint [9, 11, 16, 19, 20, 34, 35, 40]. To train the network, pairwise/triplet loss is often used with positive and negative patches [9, 16, 19, 20, 35, 40], and the average precision is used as loss function [11]. Subsequently, descriptors such as [5,7,8,14,18,21,23,33], which extract full image's feature map, have been suggested while still adopting the loss used in patch-based descriptors. More recently, [12,38] suggests a new loss that exploits the epipolar constraint using camera pose, and [36] suggests a pipeline where the detector and descriptor are learned via reinforcement learning. Usually in these works, the descriptor feature map is smaller than the image size, so the keypoint descriptors are obtained via bilinear interpolation on the dense descriptor feature map. In our method, rather than using bilinear interpolation, we use a coordinate-based implicit function to obtain local keypoint descriptors and learn the non-linearity of local descriptors on continuous-domain coordinates.

Jointly learned detector through descriptor Learned detectors such as [5,20,33] are trained so that the keypoints with high scores on an image also have high scores in the other pair images. On the other hand, [8, 18] use triplet margin loss [19] where the multiplication of pair of keypoint scores is used for loss's weight. Likewise, in [23], the learned reliability score is utilized to weight average precision loss of [11] for descriptor training. Thus, the reliability score gets proportional to the matching probability of keypoint descriptor. Inspired by [23] where both repeatability and reliability scores are used for keypoint filtering, we devised a new relative reliability score loss to train keypoints considering both repeatability and reliability scores.

3. Preliminary: KeypointNet

We adopt two detector heads from KeypointNet [33], in which (i) the location head returns the estimated real value



$H \times W$
$\frac{H}{2} \times \frac{W}{2}$
$\frac{H}{4} \times \frac{W}{4}$
$\frac{H}{8} \times \frac{W}{8}$
$\frac{H}{16} \times \frac{W}{16}$
$\frac{H}{8} \times \frac{W}{8}$
$\frac{H}{4} \times \frac{W}{4}$
⊕ : Concat
2× upsample
$\frac{H}{4} \times \frac{W}{4}$
Location head
Score head
keypoint coordinate
LIKD module
× 16
Descriptor
LIKD module
× 4

Figure 2. Overview of the proposed model architecture.

(i.e., sub-pixel) coordinates of keypoints, and (ii) the score head returns the repeatability scores (between 0 and 1) of the corresponding keypoints. Each head's input is the feature map downsampled by the cell size and each keypoint's score and coordinate are obtained from each cell. Unlike the original cell size of 8x8 pixels, we change the cell size into 4x4 pixels in order to improve the detector performance.

The score head is learned so that those keypoints with high score tend to have a small distance of point-pair correspondence $d(p_i, \hat{p}_i)$ where p_i and $\hat{p}_i$ are two corresponding points from each image. $\hat{p}_i$ is defined as the location of the keypoint nearest to the reprojected (i.e., homography warped) point of p_i, p_i^*, on the target image. The original loss function for score head is as follows [33]:

$$L_{score} = \sum_i \left[\frac{(s_i + \hat{s}_i)}{2} \cdot (d(p_i, \hat{p}_i) - \bar{d}) \right] \qquad (1)$$

where s_i and $\hat{s}_i$ are the scores of the two corresponding points on the source and target images, respectively, and $\bar{d}$ is the average reprojection error of associated points in the current image, $\bar{d} = \sum_i^L \frac{d(p_i, \hat{p}_i)}{L}$, where L represents the total number of point pairs.

4. Methods

4.1. Overall Structure

Figure 2 shows the proposed model architecture. We use 5 VGG-style blocks to reduce the resolution of the image by 16 in the encoder. Our detector is based on KeypointNet [33], where it obtains the keypoint's location and score. The smaller feature maps are upsampled via pixelshuffle and concatenated with the larger feature maps and the intermediate feature maps of 1/4 and 1/8 of input image size are used for the score head and location head for the detector.

In the decoder part, we use up-sampling and skip connections to obtain coarse- and fine-level feature maps whose sizes are 1/16 and 1/4 of the original image size, respectively. As the figure shows, two Local Implicit Keypoint Descriptor (LIKD) modules sample the coarse- and fine-level feature maps to calculate a *multi-scale* descriptor at each location of keypoint with sub-pixel resolution.

4.2. Multi-scale Descriptor and Reliability Score

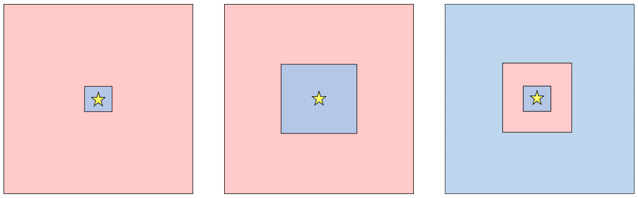
As Figure 2 shows, we first obtain two types of keypoint descriptor, coarse-grained (x16) and fine-grained (x4) ones from two LIKD modules, respectively. Then, we concatenate them for the final descriptor. We propose the following reliability score loss using the final descriptor.

The original score loss in Eqn. 1 does not take into account the descriptor. Thus, in order to ensure the correlation between keypoint's score and descriptor's discriminativeness, we devise a new loss of reliability score which contains the triplet margin loss [19] as relative reliability as follows:

$$L_{rel\_score} = \sum_i \left[\frac{(s_i + \hat{s}_i)}{2} \cdot (rel(p_i, f_i) - \bar{rel}) \right], \qquad (2)$$

$$rel(p_i, f_i) = \sum_i max\{0, \|f_i, f^*_{(i,+)}\| - \|f_i, f^*_{(i,-)}\| + m)\}, \qquad (3)$$

where the anchor f_i is the descriptor of the keypoint at p_i in the source image, and $f^*_{(i,+)}$ is a positive example, i.e., the descriptor sampled at p_t^* (the reprojected location of p_i on the target image under homography) in the target image's descriptor feature map. $f^*_{(i,-)}$ is a negative descriptor that has the highest similarity with f_i in the target image's descriptors whose position is farther than a certain distance (called safe-radius) from the positive sample. We will describe the details of negative sampling on our multi-scale descriptor in Section 4.3.



(a) Conventional safe-radius setting for training descriptor.

(b) Safe-radius setting for training x16 descriptor.

(c) Safe-radius setting for training x4 descriptor.

Figure 3. Area (in red) of negative sampling. The star in the image represents a positive sample to be used for calculating loss, and the area nearby star is also considered as positive. Thus, the descriptors of the keypoints in the blue area are not used to calculate loss through negative similarity.

This loss helps the score head selecting keypoints which produce better descriptor matching. For instance, when $rel(p_i, f_i) < \bar{rel}$, the descriptor is relatively more discriminative, thus the model must learn to set the score high to minimize the loss. On the contrary, for $rel(p_i, f_i) > \bar{rel}$, the model learns to set the score low to minimize the loss. As a result, a keypoint with a high score can not only be detected in various circumstances(e.g., viewpoint or illumination change) by L_{score} but also well-matched when using the keypoint descriptor by L_{rel_score}. Note that this loss is used to train the score head. Thus, when we calculate $rel(p_i, f_i)$ for L_{rel_score}, we stop the propagation of the gradient to the descriptor feature map to make this loss affect only score head.

4.3. Negative Sampling for Multi-Scale Descriptor

Multi-scale descriptors like ours can offer new opportunities for better discriminative power. In order to make the best use of the advantages of each of the multiple scales (x16 and x4 in our experiments), we propose re-visiting negative sampling used for the training of descriptors. In our model architecture in Figure 2, we need negative sampling on the coarse (x16) and fine-grained (x4) feature maps of decoder.

In order to train the descriptors at multiple scales, basically we follow the pipeline similar to [7, 8, 18, 33] where it uses positive and negative samples from a pair of images. In the pipeline, negative samples are selected from points at more than a small distance (called safe-radius) away from the positive sample, as illustrated in Figure 3a. In the conventional negative sampling for single-scale descriptors, the safe-radius is fixed to a single value in [18], as illustrated in Figure 3a. However, in our multi-scale descriptor training, a single fixed value of safe-radius can result in poor descriptors. For instance, in case of a coarse-grained feature map as in Figure 1c, a small safe-radius may produce negative samples whose descriptors are similar to that of a positive one. On the contrary, a large safe-radius used on a fine-grained feature map, as in Figure 1a, would lose opportunities of selecting nearby negative samples for better discrimination ability.

We propose a scale-aware negative sampling method for multi-scale descriptor learning. The basic idea is to divide the area of negative sampling in a scale-aware manner. We use a large safe-radius (16 pixels in our experiments) in case of the coarse-grained feature map as illustrated in Figure 3b. The rationale behind this idea is that the descriptors of adjacent keypoints (at continuous-domain coordinate) on the coarse-grained feature map are likely to be similar to each other due to the aggregation, e.g., bilinear interpolation, of same feature vector at pixel locations. Thus, it would be better to find negative samples farther apart from the positive one for stable learning. Additionally, the adoption of a large safe-radius has the effect of enhancing discrimination ability for distant or global negative samples.

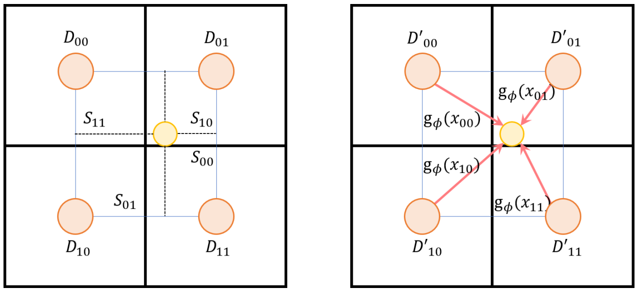
Regarding negative sampling on the fine-grained feature map, we propose a complementary solution where negative samples are selected in a band of area (between 4 and 16 pixels in our experiments) as illustrated in Figure 3c. In this case, we choose points near the keypoint as in the conventional method with a small safe-radius. However, we do not select points far from the keypoint. The rationale of this is that the negative sampling of farther points from the keypoint is covered by the coarse-grained scale as explained before. Thus, both the area of negative sampling inside of a band on the fine-grained feature map and that of large safe-radius on the coarse-grained feature map are complementary to each other, and, both combined correspond to the conventional area of negative sampling.

Regarding the concatenated descriptor which is used at test time, we apply the conventional setting of safe-radius for negative sampling as illustrated in Figure 3a. We set safe-radius to 12 in our experiments as ASLfeat [18]) using circle loss [31]. The detail for loss function to train the descriptor will be provided in Section 4.5.

4.4. LIKD: Local Implicit Keypoint Descriptor

In order to obtain the descriptor on the real value coordinate, previous works [5, 8, 18, 33] often perform bilinear sampling (interpolation) from dense feature maps to obtain the descriptors. However, bilinear interpolation has a limited expression capability since it fails to capture the nonlinearity of local descriptors on the continuous-domain coordinate. In order to resolve this problem, we propose a new module called local implicit keypoint descriptor (LIKD), which enables us to learn the embedding space of local features on the continuous-domain coordinate.

Our proposed LIKD module is motivated by the continuous feature mapping technique called LIIF [4] for super-resolution with arbitrary scaling. Compared with LIIF, our



(a) Bilinear interpolation in LIIF

(b) LIKD's implicit modeling

Figure 4. Left: Obtaining the descriptor of sub-pixel keypoint via simple bilinear interpolation of Eqn. 6. Right: Obtaining the descriptor of sub-pixel keypoint via LIKD in Eqn. 7. For simplicity, we remove the real value coordinate x in S and D.

proposed LIKD is different in that it does not adopt the continuous function(e.g., bilinear interpolation), but the function learned through training for weights used to aggregate feature vectors. It is because the keypoint descriptor function does not need to be continuous but discriminative, as will be explained later in this section.

First, we define a coordinate-based feature generator function f_θ as follows.

$$d'_{x_{ij}} = f_\theta(D'_{x_{ij}}, \Delta x_{ij}), \tag{4}$$

where $d'_{x_{ij}}$ is the output of descriptor generator f_θ at the pixel location x_{ij} near the real value coordinate x. Δx_{ij} represents the coordinate difference between x and a pixel location x_{ij}, $\Delta x_{ij} = x - x_{ij}$. $D'_{x_{ij}}$ is obtained by concatenating the vectors of descriptor head D. Specifically, $D'_{x_{ij}}$ is calculated by concatenating descriptor vectors in the 3x3 window centered at the pixel location x_{ij} as follows:

$$D'_{x_{ij}} = Concat(\{D_{x_{i+l,j+k}}\}_{l,k\in\{-1,0,1\}}). \tag{5}$$

Note that zero padding is applied to the border of the descriptor head output D.

If we adopted the original LIIF [4] to obtain the final descriptor of the keypoint located at a real value coordinate x, we could apply the followings:

$$d'_x = \sum_{m\in\{00,01,10,11\}} \frac{S_m}{S} \times f_\theta(D'_{x_m}, \Delta x_m), \tag{6}$$

where the index m ($m \in \{00, 01, 10, 11\}$) represents one of four neighbor pixel locations around the given coordinate x of the keypoint as shown in Figure 4a. S_m represents the area of (diagonally located) rectangle assigned to x_m as shown in Figure 4a.[1]

Unlike the super-resolution task where LIIF is applied, there is no need for the keypoint descriptor function to be continuous or smooth. Instead, it needs to be discriminative enough to support the case that some specific corners are more important to explain the keypoint's descriptor than other corners regardless of their relative position in the square.

Thus, unlike Eqn. 6 used in the original LIIF, we also propose learning the weights as follows:

$$d'_x = \sum_m g_\phi(D_{x_{all}}, \Delta x_{all}, m) \cdot f_\theta(D'_{x_m}, \Delta x_m) \tag{7}$$

where $D_{x_{all}}$ and Δx_{all} are the concatenated ones of all D_{x_m} and Δx_m ($m \in \{00, 01, 10, 11\}$), respectively. f_θ and g_ϕ are each composed of three fully connected layers where f_θ consists of hidden dimensions of 512, 256, and 128 with ReLU activation functions, and g_ϕ consists of hidden dimensions of 256, 64, and 4 with softmax at the final layer. g_ϕ includes softmax layer at the end to make $\sum_m g_\phi(D, \Delta x, m) = 1$.

Note that, in terms of test-time computation cost, the LIKD module incurs negligible additional cost compared with the overall encoder-decoder architecture. It is because the frequency of executing the generator function in Eqn. 7 is proportional to the number of keypoints under top K (depending on the image size in our experiments) selection.

4.5. Implementation

Training We train our model using the dataset in GL3D [17, 30, 39] and [22]. The training dataset consists of about 800K image pairs and we resize images into 480 × 480. Similar to [18], the gradients are calculated only if the pair has at least 128 matches which are confirmed by relative pose and depth. We also augment each input image with random photometric augmentation of brightness, contrast, saturation and hue. The Adam optimizer is used with a learning rate set to 10^{-3}.

Loss design For training the score head and location head in detector, we adopt a similar approach to [5, 33] where a known homography transformation is used. We apply random homography transformation which is generated in training time and obtain a target image I_t from source image I_s. From the pair of images, we use location loss:

$$L_{loc} = \sum_i ||p_i^* - \hat{p}_i||_2 \ , \tag{8}$$

where p_i^* is a homography-warped point on the target image, and $\hat{p}_i$ is its associated point (i.e., the closest keypoint) on the target image. The losses for score head are defined in Eqn. 1 (L_{score}) and 2 (L_{rel_score}).

As explained in Section 4.3, for training the multi-scale descriptor, we use three losses for the coarse-grained descriptor ($L_{desc_{\times 16}}$), the fine-grained descriptor ($L_{desc_{\times 4}}$)

[1] For simplicity, we use S_m instead of S_{x_m} in the figure.

and the final concatenated descriptor ($L_{desc_{concat}}$), respectively. For coarse and fine-grained descriptors, we use triplet margin loss [19] where we set the margin to 0.3 and safe-radius to 16 and 4 respectively as mentioned before. For the concatenated descriptor, we set circle loss's hyper-parameter m_{fuse} and γ to 0.1 and 512, respectively [31]. Thus, the final loss for training is as follows:

$$\begin{aligned} L_{total} &= \lambda_1 L_{loc} + \lambda_2 L_{score} + \lambda_3 L_{rel\_score} \\ &+ \lambda_4 L_{desc_{\times 16}} + \lambda_5 L_{desc_{\times 4}} + \lambda_6 L_{desc_{concat}} \end{aligned} \tag{9}$$

where we set each of scaling parameters λ_i as 1, 2, 2, 2, 2, and 1, respectively.

5. Experiments

We evaluate our method on various datasets with different tasks, and compare ours with SOTA methods. For a direct evaluation of keypoint detector and descriptor, we use sparse feature matching on the HPatches dataset [1]. We also evaluate our method on the downstream tasks, relative pose estimation, and visual localization. In these experiments, we use Megadpeth [13] for outdoor and ScanNet [6] for indoor for relative pose estimation, and Aachen day-night dataset [26] for visual localization. Also, to check whether each of our proposed methods contributes to higher performance, we evaluate each method one by one using the HPatches dataset.

5.1. Image Matching

Datasets From HPatches dataset [1], which includes 116 image sequences with ground-truth homography, we evaluate our method on the image sequences. Each sequence contains a reference image and 5 target images with varying illumination and viewpoint changes. When we evaluate each model, we exclude 8 high-resolution sequences following D2-Net [8].

Evaluation metrics We follow three metrics which are mainly used in the dataset. 1) Repeatability: the ratio of possible matches and the minimum number of keypoints in the shared view. 2) Matching score (M.score): the average of each ratio of successful matches and the minimum number of keypoints in the shared view from two images. 3) Mean matching accuracy (MMA): the ratio of correct matches and possible matches. For matching keypoints, we use the nearest neighborhood from one image to the other, and the mutual nearest neighborhood in M.score and MMA respectively. In each metric, the matches are considered as successful if the distance between true warped keypoints and matched keypoints is under 3 pixels. We measured repeatability and M.score following KeypointNet [33], and MMA following D2-Net [8]. We limit the maximum number of features of our methods to 5K.

Method	Repeat.	MMA@3	M.Score
KeypointNet [33] w/ 8x8 cells	0.654	0.740	0.521
KeypointNet w/ 4x4 cells	0.747	0.760	0.533
V1 - Multi-scale descriptor	0.749	0.765	0.547
V2 - V1 + Reliability loss	0.748	0.770	0.549
V3 - V2 + LIKD (area weight)	0.749	0.772	0.551
V4 - V2 + LIKD (learned weight)	**0.750**	0.776	0.552
V5 - V4 + Our negative sampling	0.748	**0.785**	**0.560**

Table 1. Ablation experiment for KeypointNet and our model with 5 configurations discussed in Section 4.2, 4.3 and 4.4.

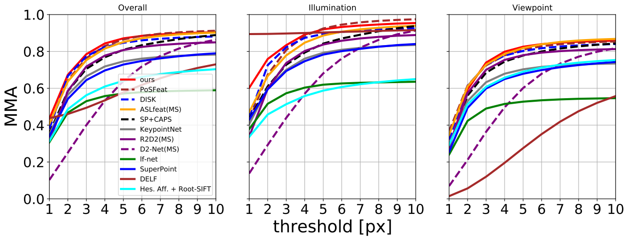


Figure 5. Comparison for Mean Matching Accuracy (MMA) on HPatches dataset [1]. We evaluate each method with varying pixel error thresholds. Our method achieves the best in overall performance. Table 2 compares top performing methods in detail.

Methods	MMA@1	MMA@3	MMA@5
R2D2 (MS)	0.363	0.728	0.807
SP + CAPS	0.356	0.705	0.809
ASLfeat (MS)	0.406	0.752	0.851
DISK	0.380	0.773	0.847
PoSFeat	0.396	0.765	0.865
Ours	**0.434**	**0.785**	**0.871**

Table 2. Comparison of top performing methods in Figure 5.

Ablation study We evaluate five different versions of our method. We define V1-V2 as (i) V1: From our model architecture in Figure 2, V1 is trained from loss L_{loc}, L_{score} and $L_{desc_{concat}}$; (ii) V2: The loss L_{rel_score} is also used when training. The keypoint descriptors of the two versions are sampled via bilinear interpolation from each feature map. Starting from V3, we apply LIKD module, (iii) V3: The weights for each corner's feature vector in LIKD module are determined by the area as Eqn. 6; (iv) V4: The weights for each corner's feature vector are learned as Eqn. 7; (v) V5: From V4, both $L_{desc_{\times 16}}$ and $L_{desc_{\times 4}}$ are used together with our proposed negative sampling.

The evaluation results are shown in Table 1. The comparison of the first two rows between KeypointNet with 8x8 and 4x4 cells shows that 4x4 cells can boost performance. We found that the detector of cell size 8 has early saturation of repeatability when increasing the number of keypoints.

Methods	Accuracy on ScanNet [%]			Accuracy on MegaDepth [%]		
	d_{frame} = 10	d_{frame} = 30	d_{frame} = 60	*easy*	*moderate*	*hard*
SIFT	91.0 / 14.1	65.1 / 15.6	41.4 / 11.9	58.9 / 20.2	26.9 / 11.8	13.6 / 9.6
SuperPoint	94.4 / 17.5	75.9 / 26.3	53.4 / 22.1	67.2 / 27.1	38.7 / 18.8	24.5 / 14.1
LF-Net	93.6 / 17.4	76.0 / 22.4	49.9 / 18.0	52.3 / 18.6	25.5 / 13.2	15.4 / 11.1
D2-Net (MS)	97.5 / 19.0	83.7 / 27.5	62.9 / 25.1	70.6 / 31.9	47.9 / 23.6	28.5 / 15.7
R2D2 (MS)	98.0 / 20.9	87.3 / 30.8	65.1 / 27.7	73.8 / 31.9	51.6 / 25.0	36.9 / 16.7
KeypointNet	94.6 / 19.0	79.0 / 26.4	54.0 / 20.7	68.0 / 29.6	43.1 / 22.1	36.6 / 12.5
GIFT w/ SuperPoint kp.	94.7 / 17.6	77.1 / 27.4	47.4 / 11.1	72.8 / 32.4	49.5 / 24.3	31.9 / 17.9
CAPS w/ SuperPoint kp.	96.1 / 17.1	79.5 / 27.2	59.3 / 26.1	72.9 / 30.5	53.5 / 27.9	38.1 / 19.2
ASLfeat (MS)	97.5 / 21.0	87.6 / 34.0	70.6 / 33.7	72.2 / 32.7	**57.2** / 29.3	40.3 / 19.9
DISK	95.3 / 19.1	77.5 / 22.7	53.2 / 21.5	74.1 / 32.5	56.6 / 29.6	**45.5** / 20.4
PoSFeat	96.3 / 18.1	77.1 / 22.7	54.7 / 21.1	**76.9 / 35.2**	**57.2 / 30.0**	43.7 / 20.2
Ours	**98.2 / 22.8**	**92.6 / 38.9**	**74.1 / 37.5**	74.1 / 32.7	55.9 / 27.9	41.3 / **20.7**

Table 3. Relative pose estimation accuracy on ScanNet [6] and MegaDepth [13]. Each cell shows the accuracy of estimated rotations / translations. Each accuracy value is defined as the percentage of pairs with relative pose error under a certain threshold. The detail of evaluation metric is described in Section 5.2.

However, the detector with cell size 4 can detect more keypoints within the same area so that more keypoints, which might be missed with cell size 8, can be detected thereby yielding better matching performance.

Compared with KeypointNet with 4x4 cells, our proposed baseline V1, which is also based on 4x4 cells, noticeably improves the matching accuracy. It is mainly because V1 adopts multi-scale descriptors while KeypointNet uses single-scale descriptors. The table shows V2 improves V1 by training keypoint detector with our reliability loss in Eqn. 2. Note that the reliability loss aims at increasing the correlation between the keypoint score and the discrimination ability of descriptors, which contributes to the increase in MMA@3 and M.Score without hurting the repeatability of detector.

The table also shows that our proposed LIKD can further improve the metrics utilizing the coordinate-based feature generation with area-proportional (V3) or non-linear (V4) modeling. Finally, our scale-aware negative sampling (V5), which is applied to $L_{desc_{\times 16}}$ and $L_{desc_{\times 4}}$ to exploit the benefit of each of two different scales of the feature map, boosts the performance, especially in MMA and M.Score.

Results and comparison Figure 5 shows that our method outperforms the other methods, especially on MMA across all the thresholds in HPatches dataset. Table 2 gives a detailed comparison of the figure in terms of MMA@1, @3 and @5. The table shows that our method offers large margins from the other SOTA methods.

5.2. Relative Pose Estimation

Datasets We use MegaDepth dataset [13], an outdoor dataset which provides a pair of images with a difference in illumination and viewpoint with large time changes. The image pairs for evaluation are provided by CAPS [38], where there are three subsets with 1,000 images each according to relative rotation angle: easy ([0°, 15°]), moderate ([15°, 30°]) and hard ([30°, 60°]). For the indoor dataset, we use ScanNet dataset [6], where image pairs are randomly sampled at three different frame intervals, 10, 30, and 60. We follow the sampled image pairs from LF-net [20] and CAPS [38] and each subset consists of about 1,000 image pairs.

Evaluation metrics To estimate relative pose, we first derive the essential matrix from mutual nearest neighbor matches and OPENCV [3]'s *findEssentialMat* with RANSAC [10]. We can decompose it into relative rotation and translation through OPENCV's *recoverPose*. We consider a rotation or translation to be correct if the angular deviation is less than a threshold, and report the average accuracy for that threshold as CAPS [38]. We set a threshold of 5° for ScanNet, and 10° for MegaDepth because of the large variation between images in terms of viewpoint or illumination on Megadepth. We limit the maximum number of features to 5K and 10K on MegaDepth and ScanNet respectively for all the methods.

Results and comparison Table 3 shows that our method outperforms all other methods in ScanNet dataset, and has competitive performance with other methods on

Methods	Avg. #Features	Dim	0.25m, 2°	0.5m, 5°	5m, 10°	0.5m, 2° (V1.1)	1m, 5° (V1.1)	5m, 10° (V1.1)
SuperPoint	4K	256	73.5	80.6	91.8	69.6	85.9	95.3
SuperPoint KP + CAPS	4K	256	80.6	88.8	99.0	71.2	**88.0**	97.9
SuperPoint KP + Our descriptors	4K	256	**81.6**	89.8	**100.0**	73.3	86.4	97.9
D2Net (MS)	14K	512	79.6	87.8	100.0	67.5	86.4	97.4
D2Net KP (MS) + Our descriptors	14K	256	79.6	89.8	100.0	70.7	87.4	**98.4**
ASLfeat (MS)	9K	128	80.6	87.8	99.0	70.2	84.8	97.4
ASLfeat KP (MS) + Our descriptors	9K	256	78.6	87.8	**100.0**	71.2	86.9	97.9
R2D2 (MS)	10K	128	76.5	88.8	**100.0**	71.7	85.9	96.9
R2D2 KP (MS) + Our descriptors	10K	256	78.6	**90.8**	**100.0**	72.8	**88.0**	97.4
DISK	10K	128	**81.6**	89.8	**100.0**	**74.3**	86.9	97.4
DISK KP + Our descriptors	10K	256	80.6	89.8	**100.0**	71.7	84.8	**98.4**
PoSFeat	10K	128	78.6	88.8	**100.0**	70.7	85.3	97.9
PoSFeat KP + Our descriptors	10K	256	78.6	89.8	**100.0**	73.8	86.9	97.9
Ours	15K	256	78.6	89.8	**100.0**	71.7	86.4	**98.4**

Table 4. Evaluation results on Aachen Day-Night dataset [27, 41] for visual localization.

MegaDepth dataset. Considering that DISK and PoSFeat are trained on MegaDepth, our method demonstrates its generalization ability in both outdoor and indoor datasets.

5.3. Visual Localization

Datasets We use Aachen Day-Night dataset [27, 41] to evaluate the utility of our method on visual localization. We use both 1.0 and 1.1 versions for the dataset where each dataset contains 98 and 191 queries for estimating pose from 4,328 and 6,697 DB images, respectively, with day-night changes.

Evaluation metrics We evaluate the queries' pose in *The Visual Localization Benchmark*[2]. The evaluation pipeline takes custom features as input for image matching and uses COLMAP [29] to reconstruct 3D models and generates the percentages of successfully localized query images with three error thresholds: $(0.25m, 2°)/(0.5m, 5°)/(5m, 10°)$. We limit the maximum number of features to 20K for all methods.

Results and comparison Table 4 shows that no method prevails in this task. According to the table, our proposed method offers competitive performance to the other SOTA methods and has a potential of offering further performance improvement when applied together with existing methods.

6. Conclusion

In this paper, we investigated the potential of multi-scale approach of keypoint descriptor from pixel to sub-pixel level. We presented a model architecture for multi-scale keypoint descriptor with sub-pixel accuracy. The proposed multi-scale descriptor is trained with negative samples found in a scale-aware and complementary manner. We also presented a sub-pixel descriptor function adopting feature generation on the continuous coordinate domain while learning non-linearity of descriptor for improving discrimination ability. Our experiments with HPatches dataset show that our proposed method outperforms the SOTA methods by a large margin and the evaluations on downstream tasks also show the competitiveness of our proposed method.

Acknowledgement This work was supported by IITP/NRF grants funded by the Korean government (MSIT, 2021-0-00105 Development of Model Compression Framework for Scalable On-Device AI Computing on Edge Applications, 2021-0-02068 Artificial Intelligence Innovation Hub, NRF-2021M3F3A2A02037893), and Inter-university Semiconductor Research Center (ISRC) at SNU.

References

[1] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors, 2017. 2, 6

[2] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In Aleš Leonardis, Horst Bischof, and Axel Pinz, editors, *Computer Vision – ECCV 2006*, pages 404–417, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. 2

[3] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. 7

[4] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function, 2021. 4, 5

[5] Peter Hviid Christiansen, Mikkel Fly Kragh, Yury Brodskiy, and Henrik Karstoft. Unsuperpoint: End-to-end unsupervised interest point detector and descriptor, 2019. 1, 2, 4, 5

[6] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas A. Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. *CoRR*, abs/1702.04405, 2017. 2, 6, 7

[7] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description, 2018. 1, 2, 4

[2] http://visuallocalization.net/

[8] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable cnn for joint detection and description of local features, 2019. 1, 2, 4, 6

[9] Patrick Ebel, Anastasiia Mishchuk, Kwang Moo Yi, Pascal Fua, and Eduard Trulls. Beyond cartesian representations for local descriptors. *CoRR*, abs/1908.05547, 2019. 2

[10] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, jun 1981. 7

[11] Kun He, Yan Lu, and Stan Sclaroff. Local descriptors optimized for average precision. *CoRR*, abs/1804.05312, 2018. 2

[12] Kunhong Li, Longguang Wang, Li Liu, Qing Ran, Kai Xu, and Yulan Guo. Decoupling makes weakly supervised local feature better. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15838–15848, June 2022. 2

[13] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. *CoRR*, abs/1804.00607, 2018. 2, 6, 7

[14] Yuan Liu, Zehong Shen, Zhixuan Lin, Sida Peng, Hujun Bao, and Xiaowei Zhou. Gift: Learning transformation-invariant dense visual descriptors via group cnns, 2019. 2

[15] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004. 1, 2

[16] Zixin Luo, Tianwei Shen, Lei Zhou, Siyu Zhu, Runze Zhang, Yao Yao, Tian Fang, and Long Quan. Geodesc: Learning local descriptors by integrating geometry constraints. *CoRR*, abs/1807.06294, 2018. 2

[17] Zixin Luo, Tianwei Shen, Lei Zhou, Siyu Zhu, Runze Zhang, Yao Yao, Tian Fang, and Long Quan. Geodesc: Learning local descriptors by integrating geometry constraints. In *European Conference on Computer Vision (ECCV)*, 2018. 5

[18] Zixin Luo, Lei Zhou, Xuyang Bai, Hongkai Chen, Jiahui Zhang, Yao Yao, Shiwei Li, Tian Fang, and Long Quan. Aslfeat: Learning local features of accurate shape and localization. *CoRR*, abs/2003.10071, 2020. 1, 2, 4, 5

[19] Anastasiya Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. *CoRR*, abs/1705.10872, 2017. 2, 3, 6

[20] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. Lf-net: Learning local features from images. *CoRR*, abs/1805.09662, 2018. 1, 2, 7

[21] Rémi Pautrat, Viktor Larsson, Martin R. Oswald, and Marc Pollefeys. Online invariance selection for local feature descriptors, 2020. 2

[22] Filip Radenovic, Giorgos Tolias, and Ondrej Chum. CNN image retrieval learns from bow: Unsupervised fine-tuning with hard examples. *CoRR*, abs/1604.02426, 2016. 5

[23] Jerome Revaud, Philippe Weinzaepfel, César De Souza, Noe Pion, Gabriela Csurka, Yohann Cabon, and Martin Humenberger. R2d2: Repeatable and reliable detector and descriptor, 2019. 1, 2

[24] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In Aleš Leonardis, Horst Bischof, and Axel Pinz, editors, *Computer Vision – ECCV 2006*, pages 430–443, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. 2

[25] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International Conference on Computer Vision*, pages 2564–2571, 2011. 1, 2

[26] Torsten Sattler, Will Maddern, Akihiko Torii, Josef Sivic, Tomás Pajdla, Marc Pollefeys, and Masatoshi Okutomi. Benchmarking 6dof urban visual localization in changing conditions. *CoRR*, abs/1707.09092, 2017. 1, 2, 6

[27] Torsten Sattler, Tobias Weyand, Bastian Leibe, and Leif Kobbelt. Image retrieval for image-based localization revisited. 09 2012. 8

[28] Nikolay Savinov, Akihito Seki, Lubor Ladicky, Torsten Sattler, and Marc Pollefeys. Quad-networks: unsupervised learning to rank for interest point detection. *CoRR*, abs/1611.07571, 2016. 1

[29] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 8

[30] Tianwei Shen, Zixin Luo, Lei Zhou, Runze Zhang, Siyu Zhu, Tian Fang, and Long Quan. Matchable image retrieval by learning from surface reconstruction. In *The Asian Conference on Computer Vision (ACCV*, 2018. 5

[31] Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. Circle loss: A unified perspective of pair similarity optimization. *CoRR*, abs/2002.10857, 2020. 4, 6

[32] Jiexiong Tang, Rares Ambrus, Vitor Guizilini, Sudeep Pillai, Hanme Kim, and Adrien Gaidon. Self-supervised 3d keypoint learning for ego-motion estimation. *CoRR*, abs/1912.03426, 2019. 1

[33] Jiexiong Tang, Hanme Kim, Vitor Guizilini, Sudeep Pillai, and Rares Ambrus. Neural outlier rejection for self-supervised keypoint learning, 2019. 1, 2, 3, 4, 5, 6

[34] Yurun Tian, Bin Fan, and Fuchao Wu. L2-net: Deep learning of discriminative patch descriptor in euclidean space. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6128–6136, 2017. 2

[35] Yurun Tian, Xin Yu, Bin Fan, Fuchao Wu, Huub Heijnen, and Vassileios Balntas. Sosnet: Second order similarity regularization for local descriptor learning. *CoRR*, abs/1904.05019, 2019. 2

[36] Michal J. Tyszkiewicz, Pascal Fua, and Eduard Trulls. DISK: learning local features with policy gradient. *CoRR*, abs/2006.13566, 2020. 1, 2

[37] Yannick Verdie, Kwang Moo Yi, Pascal Fua, and Vincent Lepetit. TILDE: A temporally invariant learned detector. *CoRR*, abs/1411.4568, 2014. 1

[38] Qianqian Wang, Xiaowei Zhou, Bharath Hariharan, and Noah Snavely. Learning feature descriptors using camera pose supervision. *CoRR*, abs/2004.13324, 2020. 2, 7

[39] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-

scale dataset for generalized multi-view stereo networks. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 5

[40] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. LIFT: learned invariant feature transform. *CoRR*, abs/1603.09114, 2016. 2

[41] Zichao Zhang, Torsten Sattler, and Davide Scaramuzza. Reference pose generation for visual localization via learned features and view synthesis. *CoRR*, abs/2005.05179, 2020. 8