International Journal of Computer Vision (2023) 131:2908–2932
https://doi.org/10.1007/s11263-023-01837-3

Deep Corner

Shanshan Zhao[1] · Mingming Gong[2] · Haimei Zhao[3] · Jing Zhang[3] · Dacheng Tao[3]

Received: 1 October 2022 / Accepted: 6 June 2023 / Published online: 5 July 2023

Abstract
Recent studies have shown promising results on joint learning of local feature detectors and descriptors. To address the lack of ground-truth keypoint supervision, previous methods mainly inject appropriate knowledge about keypoint attributes into the network to facilitate model learning. In this paper, inspired by traditional corner detectors, we develop an end-to-end deep network, named Deep Corner, which adds a local similarity-based keypoint measure into a plain convolutional network. Deep Corner enables finding reliable keypoints and thus benefits the learning of the distinctive descriptors. Moreover, to improve keypoint localization, we first study previous multi-level keypoint detection strategies and then develop a multi-level U-Net architecture, where the similarity of features at multiple levels can be exploited effectively. Finally, to improve the invariance of descriptors, we propose a feature self-transformation operation, which transforms the learned features adaptively according to the specific local information. The experimental results on several tasks and comprehensive ablation studies demonstrate the effectiveness of our method and the involved components.

Keywords Detector · Descriptor · Local feature · Keypoint · Image matching

1 Introduction

Local feature detection and description are essential stages for various applications, such as structure-from-motion (Heinly et al., 2015), image retrieval (Sivic & Zisserman, 2003), and visual localization (Svärm et al., 2017; Li et al., 2012). Due to the crucial role in computer vision, these two problems have been studied extensively over several decades. A classical approach to local features is first obtaining the location of keypoints using hand-crafted detectors (Lowe, 2004; Harris et al., 1988; Mikolajczyk & Schmid, 2004; Bay et al., 2006; DeTone et al., 2018), and then extracting the representations for each point using hand-crafted descriptors (Lowe, 2004; Mikolajczyk & Schmid, 2005), *a.k.a, detect-then-describe*. In recent years, the success of Deep Convolutional Neural Networks (DCNNs) in various computer vision tasks has promoted the research on deep learning-based detector (Barroso-Laguna et al., 2019; Savinov et al., 2017; Verdie et al., 2015; Zhang et al., 2017) and descriptor (Simo-Serra et al., 2015; Ebel et al., 2019a; Tyszkiewicz et al., 2020; Tian et al., 2017, 2020b; Yi et al., 2016a; Mishkin et al., 2018; Yi et al., 2016b; Wang et al., 2020; Tian et al., 2019; Luo et al., 2018; Ebel et al., 2019b). However, these methods still follow the *detect-then-describe* pipeline. Recently, joint learning of detector and descriptors, *a.k.a, describe-*

Communicated by Ondra Chum.

Part of this work was done when Shanshan Zhao studied at USYD. Ms. Haimei Zhao, Dr. Jing Zhang, and Prof. Dacheng Tao were partially supported by Australian Research Council Projects IH-180100002 and FL-170100117. Dr. Mingming Gong was supported by ARC DE210101624.

✉ Dacheng Tao
dacheng.tao@gmail.com

Shanshan Zhao
sshan.zhao00@gmail.com

Mingming Gong
mingming.gong@unimelb.edu.au

Haimei Zhao
hzha7798@uni.sydney.edu.au

Jing Zhang
jing.zhang1@sydney.edu

[1] JD Explore Academy, Beijing, China

[2] School of Mathematics and Statistics and Melbourne Centre for Data Science, The University of Melbourne, Melbourne, VIC, Australia

[3] School of Computer Science, Faculty of Engineering, The University of Sydney, Camperdown, NSW, Australia

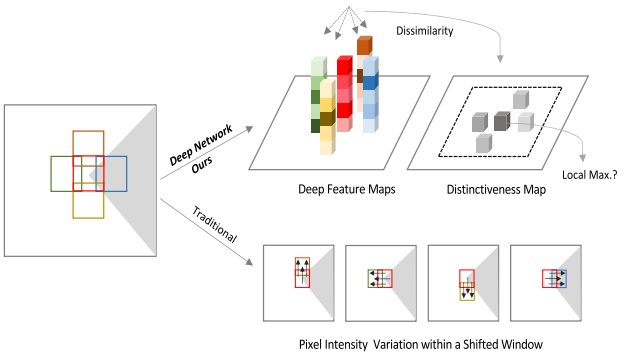

Fig. 1 Illustration of different keypoint detection measures. Bottom: traditional methods (Harris et al., 1988; Moravec, 1977), based on the pixel intensity variation within a shifted window; Up: ours, based on the similarity of deep features

and-detect, has received more and more attention (Luo et al., 2020; Dusmanu et al., 2019; Revaud et al., 2019) due to its simplicity in the pipeline.

Unlike high-level computer vision tasks, such as object detection (Girshick et al., 2014) and semantic segmentation (Long et al., 2015), it is hard to manually label the ground truth location of keypoints, which is semantically ill-defined (DeTone et al., 2018). A feasible way to learn the keypoints is using an available detector to extract potential keypoints as the pseudo-label and then training the keypoint detection model in a supervised manner. For example, TILDE (Verdie et al., 2015) exploits SIFT to detect the keypoints at multiple scales and selects positive and negative samples according to the repeatability. In comparison, SuperPoint (DeTone et al., 2018) firstly trains a detector on a synthetic dataset consisting of simple geometric shapes with no ambiguity in the keypoint locations, such as vertices of triangles. Then, the pre-trained detector is applied to the real image many times by sampling random homographies to generate pseudo-labels which are used to further adapt the detector to real data. To generate reliable pseudo keypoints, the detector is generally required to process each training image many times, such as 100 homographies in SuperPoint.

Another solution is learning the detector and descriptor directly from the training real images without an additional pseudo keypoints extraction process (Dusmanu et al., 2019; Luo et al., 2020; Tyszkiewicz et al., 2020; Revaud et al., 2019). For example, D2Net (Dusmanu et al., 2019) and ASLFeat (Luo et al., 2020) develop a joint optimization approach for detector and descriptor learning, which enables the locally distinctive pixels to get higher detection score, *i.e.,* be potential keypoint. In comparison, DISK (Tyszkiewicz et al., 2020) defines the matching score as a reward and exploits policy gradient method to optimize the keypoint score. For this kind of solution, since there is no ground truth keypoint as the supervision for model learning, it is important to define a training objective which can implicitly guide the network to maximize the detection score of the potential keypoints. To achieve this, D2Net (Dusmanu et al., 2019) directly derives the keypoints from the deep feature maps that are considered as the detection response map in traditional approaches (Lowe, 2004). Following D2Net (Dusmanu et al., 2019), ASLFeat (Luo et al., 2020) proposes the peakiness measure at multiple scales, which benefits the accurate localization of keypoints. R2D2 (Revaud et al., 2019) proposes to jointly learn a reliability map by maximizing the local peakiness and a repeatability map by modeling the descriptor matching precision. These methods mainly study the characteristics of keypoints and devise a keypoint detection score formulation that can guide the network to learn the detector. In fact, traditional hand-crafted detectors have made great efforts to define the keypoint score, which enables us to explore whether we can combine the efficient traditional detectors with deep neural networks in this paper.

Specifically, we investigate insights from the traditional corner detectors, especially those based on either gradient (Harris et al., 1988; Shi et al., 1994; Moravec, 1977) or intensity (Trajković & Hedley, 1998). For instance, the seminal Moravec's corner detector (Moravec, 1977) defines a corner to be a point with low self-similarity, *i.e.,* how similar a patch centered on the pixel is to nearby and overlapping patches, as shown in Fig. 1. The similarity is calculated as the sum of squared differences (SSD) between the corresponding pixels of two patches. Finally, the cornerness measure is defined as the smallest SSD between the patch and its neighbours in horizontal, vertical, and diagonal directions. Inspired by the classical corner detectors, we propose a new deep learning-based approach for joint detector and descriptor learning. In DCNNs, each point located in the learned deep feature maps corresponds to a patch in the original image. Therefore, we can consider the similarity between spatially nearby feature vectors as the similarity between corresponding patches in the original image, as shown in Fig. 1. Based on this connection, we introduce a similarity-based keypoint measure, which evaluates the similarity between each pixel and its neighbours on the CNN feature maps. Our approach can be viewed as a corner detector that utilizes deep neural networks to compute the cornerness measure and thus we term our approach as *Deep Corner*.

Here, we first show the superiority of our Deep Corner through a simple experiment. We train two models with the same structure on GL3D dataset (Shen et al., 2018) using our similarity-based measure (S.M.) and CNN feature-based peakiness measure (F.M.) (Revaud et al., 2019; Luo et al., 2020), respectively. We report the *%Rep* and *%MMA* (higher better) on HPatches (Balntas et al., 2017) at different training stages in Fig. 2. We can find that our method performs better, even in the initialization state. Moreover, we also try to use our learned keypoint measure to guide the detector learning of F.M. using knowledge distillation (Hinton et al.,

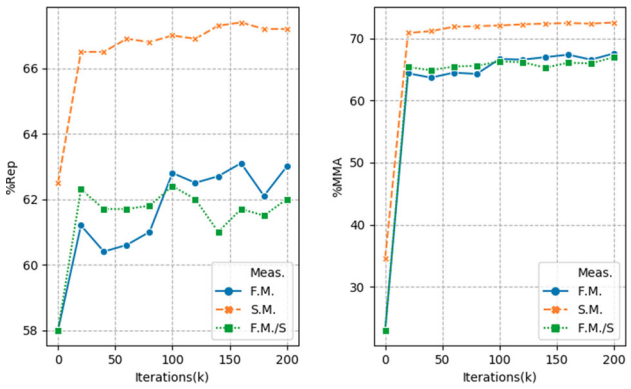

Fig. 2 Repeatability (*%Rep*) and Mean Matching Accuracy (*%MMA*) on HPatches (Balntas et al., 2017) at different training stages

2015; Gou et al., 2020), which is referred to as F.M./S. With the supervision, F.M./S can improve the performance at the beginning stage, but the overall performance does not change. The results indicate that our similarity-based measure, which is derived from the traditional corner detector, is more effective than the keypoint detector measure solely based on CNNs to find the potential keypoints.

Apart from the new similarity-based measure, we further improve the keypoint localization accuracy and the distinctiveness of descriptors by incorporating a multi-level structure and a feature self-transformation layer. Specifically, since the keypoint localization is a pixel-level task, it would be helpful to calculate the keypoint measure in the feature maps with the original resolution. To achieve this, we design a multi-level architecture based on U-Net (Ronneberger et al., 2015), named MU-Net, which deploys the U-Net structure at multiple levels. The developed MU-Net is able to associate the high-level information with the local structure information and also preserve the local details through up-sampling feature maps at different scales to the original resolution. Moreover, in the typical CNN framework, all weights and biases are shared across all spatial locations, which might be not effective in learning invariant features robust to complex changes within image pairs. To make the network more flexible to model robust representations efficiently, we propose a feature self-transformation operation, which transfers the learned features into a new space by learning a scale factor and an offset factor adaptively according to the encoded content in each location. In addition, the feature maps usually contain specific information in each channel, and it is likely that the similarity between some channels is more related to the cornerness. Therefore, we further study to extend the similarity-based measure in Deep Corner to a multi-group version, where the feature maps are split into multiple groups and we compute the keypoint measure for each group. We conduct a series of experiments and ablation studies to analyze our method quantitatively and qualitatively. The experimental results on several benchmarks can demonstrate the effectiveness of our method.

2 Related Work

Detect-then-describe is a classical pipeline for local features, where the keypoints are firstly extracted by the detectors (Lowe, 2004; Zhang et al., 2023; Barroso-Laguna et al., 2019; Savinov et al., 2017; Verdie et al., 2015; Zhang et al., 2017; Harris et al., 1988; Richardson & Olson, 2013; Tian et al., 2020a; Mikolajczyk & Schmid, 2004; Zhao et al., 2022b; Bay et al., 2006) and then represented into a feature vector using the descriptors (Simo-Serra et al., 2015; Tian et al., 2017, 2020b; Lowe, 2004; Mikolajczyk & Schmid, 2005; Yi et al., 2016a; Mishkin et al., 2018; Yi et al., 2016b; Keller et al., 2018; Balntas et al., 2016; Wang et al., 2020; Tian et al., 2019; Luo et al., 2018; Ono et al., 2018; Ebel et al., 2019b; Mishchuk et al., 2017; Wang et al., 2022a). The detection and description are usually two independent processes. In the following, we review related previous works for detector and descriptor separately.

Both detectors and descriptors can be hand-crafted or learning-based. In recent years, with the advent of DCNNs, noticeable progress has been achieved in the learning-based solution, especially for the descriptors. A key object of deep descriptor models is learning shape-invariant features, which are insensitive to scale or view changes. One way to achieve this is exploiting data augmentation techniques, *e.g.,* affine transformations including rotation and scaling, on the image patches (Luo et al., 2019; Tian et al., 2017; Luo et al., 2018; He et al., 2018; Tian et al., 2019). Additionally, there are some methods (Potje et al., 2021; Yi et al., 2016c, a) directly modeling the shape-aware parameters. For example, Yi et al. (2016c), Yi et al. (2016a) attempt to learn a canonical orientation for each feature point by minimizing the feature distance between positive patches (Yi et al., 2016c) or through the Spatial Transformer (Jaderberg et al., 2015) operation (Yi et al., 2016a). To improve existing descriptors, including both hand-crafted and learning-based, Wang et al. (2022d) develop a lightweight neural network with two stages, *i.e.,* self-boosting and cross-boosting, which achieve the descriptor enhancement by exploiting geometric properties of the keypoints and mining the possible correlation between different keypoints, respectively. To address the limitation that more invariance might make descriptors less informative, Pautrat et al. (2020) and Li et al. (2022a) study the invariance selection for adapting the local feature descriptors to adverse changes in images. The former achieves this by developing a meta descriptor approach to automatically select the best invariance from learned several local descriptors with multiple variance properties; while the latter adopts a similar strategy but exploits a parallel self-attention module to get

the meta descriptors. To alleviate the requirement for per-pixel correspondence-level supervision, Revaud et al. (2022) devise an unsupervised learning strategy for local descriptors through explicitly integrating two matching priors (*i.e.,* local consistency and uniqueness of the matching) in the loss objective. Recent years have witnessed that the features extracted in the original image can be exploited to recover the image appearance (Weinzaepfel et al., 2011; Mai et al., 2018), which might cause privacy disclosure. To protect sensitive information, privacy-preserving local descriptors have been also studied recently (Dusmanu et al., 2021; Ng et al., 2022).

DCNNs based detector learning focuses on the repeatability. For example, Verdie et al. (2015) propose to learn an efficient piece-wise linear regressor robust to drastic illumination changes as the keypoint detector, while Barroso-Laguna et al. (2019) study the hand-crafted and learned features together in a shallow multi-scale network and extract keypoints at different scales. Aiming at detecting rotation-invariant keypoints against geometric variations, (Lee et al., 2022) develop a self-supervised equivariant learning strategy based on group-equivariant convolutional neural networks with a proposed dense orientation alignment loss. To achieve the spatial distribution uniformity of keypoints and then improve the high-level matching tasks, Yan et al. (2022) devise an objective function integrating uniformity and repeatability. Due to the non-differentiable property, an alternate optimization algorithm is further developed to optimize the objective efficiently. In addition, a straightforward and interesting way to take advantage of the capability of deep learning for keypoint detection is applying the traditional corner detection strategy (Harris et al., 1988) directly on the deep features extracted from a pre-trained deep model. For example, D2D (Tian et al., 2020a) introduces two terms, named absolute saliency measure and relative saliency measure, to find keypoints from a pre-trained descriptor without any additional training. D2D only focuses on the keypoint detection and does not learn the detector and descriptor jointly, which might be not able to mine the capacity of deep neural networks effectively. In comparison, we investigate the traditional detector strategy in a *describe-and-detect* framework by designing suitable measures and operations for both detector and descriptor learning.

Describe-and-detect (Luo et al., 2020; Barroso-Laguna et al., 2020; Dusmanu et al., 2019; Christiansen et al., 2019; Zhang et al., 2020; DeTone et al., 2018; Revaud et al., 2019; Liu et al., 2021; Bhowmik et al., 2020; Shen et al., 2019; Suwanwimolkul et al., 2021; Zhao et al., 2022a; Tyszkiewicz et al., 2020; Wang et al., 2022c; Santellani et al., 2022; Yang et al., 2022; Siqueira et al., 2022; Sun et al., 2022b) aims to extract the keypoints and corresponding descriptors in a single stage. Dusmanu et al. (2019) propose the first solution, *i.e.,* D2Net. D2Net couples the feature detector with the feature descriptor tightly, where the detection map and the descriptors are from the same deep feature maps. They use the VGG16 (Simonyan & Zisserman, 2015) pre-trained on ImageNet (Krizhevsky et al., 2012) to initialize the backbone network. However, D2Net is prone to low accuracy of keypoint localization. To address this issue, Luo et al. (2020) propose a simple multi-level keypoint detection maps fusion strategy. Additionally, they exploit the deformable convolution (Zhu et al., 2019) to extract geometric-invariant features. In comparison, Revaud et al. (2019) propose to estimate a reliability map as well as a repeatability map for learning repeatable and reliable matches. To address the problem of ambiguity in the ground truth, DeTone et al. (2018) propose to train the network with two branches, one for detector and the other for the descriptor, on synthetic data with the pseudo-ground truth using self-training. ASLFeat achieves state-of-the-art scores on multiple tasks among these methods. To make the model perform better on the downstream task, like matching, Tyszkiewicz et al. (2020) exploit a different learning strategy by optimizing the matching reward in a reinforcement learning framework. Instead of using a series of convolutional operations as previous methods did, Wang et al. (2021) exploit the Transformer structure (Vaswani et al., 2017) to capture the long-range dependencies and then improve the feature representation. Similarly, in Wang et al. (2022b), the transformer is also exploited to capture wider spatial context to construct robust local descriptors. To fully exploit both low-level and high-level features, Sun et al. (2022b) develop an adaptive multi-level feature fusion structure for descriptor learning. Considering that it might be challenging to jointly train the detector and descriptor in the describe-and-detect pipeline, Li et al. (2022b) propose to decouple the detection stage from the description step and first learn the description network which is then frozen when the detection network is training. In this paper, we follow the training scheme in D2Net and ASLFeat, while aiming at studying the learning of detectors and descriptors through taking advantage of the similarity between neighboured points.

3 Deep Corner

In this section, we introduce our approach in detail, including the similarity based measure for keypoint detection, feature self-transformation for descriptor learning. Before presenting our method, we first briefly review two previous works, *i.e.,* D2Net (Dusmanu et al., 2019) and ASLFeat (Luo et al., 2020), which are closely related to our work.

3.1 Revisiting D2Net and ASLFeat

Let $I \in \mathbb{R}^{H\times W}$ and $X = F(I) \in \mathbb{R}^{C\times H'\times W'}$ denote the input image and deep representation acquired from the network F,

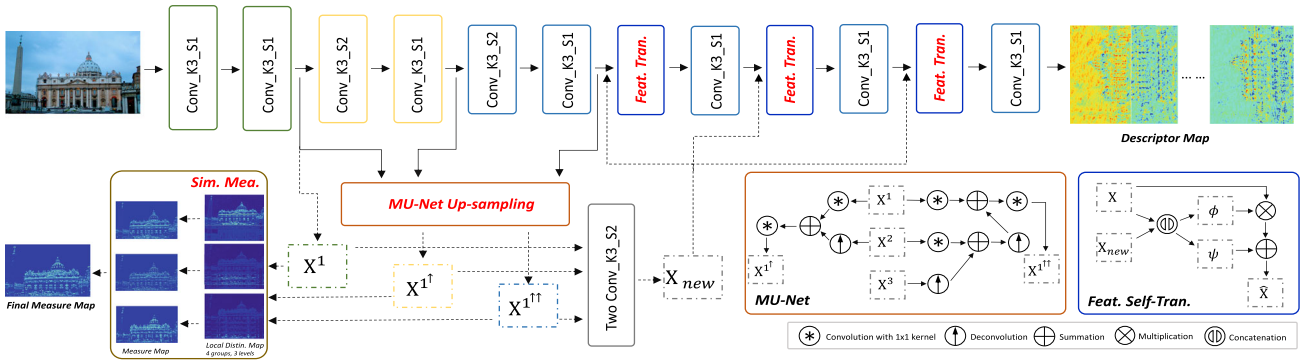

Fig. 3 Architecture of Deep Corner. The notations are identical to the text. Conv_km_sn represents the convolution with $m \times m$ kernel and stride of $n \times n$. Zoom in for best view

where H (H') and W (W') represent the height and width of I (X), respectively, and C represents the number of channels. Based on the representation X, D2Net (Dusmanu et al., 2019) gives the following definition of a keypoint:

$$(i, j) \text{ is a keypoint} \Longleftrightarrow X^c_{ij} \text{ is a local max. in } X^c, \quad \text{with } c = \underset{t=1,2,\ldots,C}{argmax}\ X^t_{ij}. \tag{1}$$

According to the definition, D2Net and ASLFeat design different formulations to calculate the keypoint measure. In addition, in contrast to D2Net, which only considers the last feature maps for the computation of keypoint measure map, ASLFeat exploits the feature maps at multiple scales, and fuses the measures via the summation operation.

To train the network, a set of image pairs and the correspondences between them are required. Considering an image pair (I, I') and the correspondence set $\mathcal{O}$ between them, we denote the keypoint measure by s_o and s'_o, the descriptor (feature vector) by $\boldsymbol{x}_o$ and $\boldsymbol{x}'_o$ for each correspondence $o \in \mathcal{O}$. Then, the loss function in D2Net and ASLFeat is written as:

$$\mathcal{L}(I, I') = \frac{1}{|\mathcal{O}|} \sum_{o \in \mathcal{O}} \frac{s_o s'_o}{\sum_{q \in \mathcal{O}} s_q s'_q} \mathcal{M}(\boldsymbol{x}_o, \boldsymbol{x}'_o), \tag{2}$$

where $\mathcal{M}(\boldsymbol{x}, \boldsymbol{x}')$ denotes the ranking loss for descriptor learning. In ASLFeat, the hardest-contrastive form (Choy et al., 2019) is exploited to implement $\mathcal{M}(\boldsymbol{x}, \boldsymbol{x}')$ as follows:

$$\begin{aligned} \mathcal{M}(\boldsymbol{x}_o, \boldsymbol{x}'_o) &= [d(\boldsymbol{x}_o, \boldsymbol{x}'_o) - m_p]_+ \\ &+ [m_n - min(\min_{l \neq o} d(\boldsymbol{x}_o, \boldsymbol{x}'_l), \min_{l \neq o} d(\boldsymbol{x}_l, \boldsymbol{x}'_o))]_+, \end{aligned} \tag{3}$$

where $d(\boldsymbol{x}, \boldsymbol{x}')$ denotes the Euclidean distance, and m_p and m_n represent the predefined margins. In the following, we introduce our method with the notations defined above.

3.2 Keypoint Detection

Similarity-based keypoint measure. Traditional works on corner detection select the points with distinctive properties (Harris et al., 1988; Shi et al., 1994). Our Deep Corner, inspired by Moravec's corner detector (Moravec, 1977), defines a corner to be a point with low self-similarity. The self-similarity is defined as the similarity between the patch centered on the pixel and the nearby overlapping patches. Our method is based on the same principle but our approach differs from Moravec's corner detector in two aspects. First, we consider the self-similarity property on the learned deep feature maps instead of the raw pixels. As a location in the deep feature maps corresponds to a patch in the original image, the similarity between deep feature vectors corresponds to the measuring similarity of corresponding patches using CNN features, which contain richer structural information than the raw pixels. Second, based on self-similarity, we propose a new distinctiveness measure function that is differentiable and thus enables end-to-end learning of the network parameters.

Specifically, we expect the network to be able to detect the point which is distinctive in the local patches and at the same time the distinctiveness of which is also a local maximum. The former condition guarantees the local uniqueness of the keypoints. However, for a region with complicated textures, it is easy to find a point that differs from its neighbours. Therefore, the network might detect the keypoints with low repeatability. To alleviate this issue, we introduce the second requirement, which further constrains the local structure of the keypoints. As a result, we provide the following definition for the detector:

$$\begin{aligned} (i, j) \text{ is a keypoint} \Longleftrightarrow &(1)\ S(X_{ij}, X_{ij}) \text{ is the only max.} \\ &\text{in } \{S(X_{ij}, X_{pq})\}_{(p,q) \in \mathcal{N}(i,j)}, \\ &\text{and } (2)\ D_{ij} \text{ is a local max. in } D. \end{aligned} \tag{4}$$

Here, $S(\boldsymbol{x}_1, \boldsymbol{x}_2) = \frac{1}{exp(||\boldsymbol{x}_1 - \boldsymbol{x}_2||_2)}$ represents the similarity between two feature vectors $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, D denotes the local distinctiveness map, and $\mathcal{N}(i, j)$ represents the neighbours of point (i, j). As $S(X_{ij}, X_{ij})$ is always equal to 1, we thus define the distinctiveness as the average distance between the pixel (i, j) and its neighbors instead of $S(X_{ij}, X_{ij})$ itself. In detail, we calculate the distinctiveness D_{ij} measuring the difference between the point (i, j) and its neighbours as follows:

$$\begin{aligned} D_{ij} &= S(X_{ij}, X_{ij}) - \frac{1}{|\bar{\mathcal{N}}(i, j)|} \sum_{(p,q)\in\bar{\mathcal{N}}(i,j)} S(X_{ij}, X_{pq}), \\ &= 1 - \frac{1}{|\bar{\mathcal{N}}(i, j)|} \sum_{(p,q)\in\bar{\mathcal{N}}(i,j)} S(X_{ij}, X_{pq}), \end{aligned} \tag{5}$$

where $\bar{\mathcal{N}}(i, j) = \{(p, q)|(p, q) \in \mathcal{N}(i, j), (p, q) \neq (i, j)\}$, *i.e.,* $\bar{\mathcal{N}}(i, j)$ does not contain the point (i, j) itself.

Based on the local distinctiveness map D, we calculate the measures for the two conditions in Eq. 4, which can be optimized in a deep neural network. Specifically, for each location (i, j), to achieve the first condition in Eq. 4, we define a measure: $\alpha_{ij} = D_{ij}$.

For the second condition, we calculate a measure reflecting the local maximum property of the distinctiveness of the keypoint, written as:

$$\beta_{ij} = \sigma\left(D_{ij} - \frac{1}{|\bar{\mathcal{N}}(i, j)|} \sum_{(p,q)\in\bar{\mathcal{N}}(i,j)} D_{pq}\right), \tag{6}$$

where σ is a non-linear activation function to enforce all measures to be positive. In our experiments, we select the $SoftPlus$ function as the activation. The final measure s_{ij} of point (i, j) is obtained by:

$$s_{ij} = \alpha_{ij}\beta_{ij}. \tag{7}$$

Multi-level detection. To improve the keypoint localization accuracy, ASLFeat (Luo et al., 2020) resorts to the feature maps at multiple levels, which have different resolutions/scales. Specifically, it first gets the measure maps at different scales. Then, it up-samples the maps with low resolution to the original resolution, and a summation operation is exploited to fuse these maps. However, the fine information is lost at the low-resolution maps, and as a result, it is difficult to directly compute the keypoint detection measure from those feature maps. More specifically, the location in the map with low resolution corresponds to a large region in the original image, and the adjacent points are distant from each other in the original image. Therefore, the relationship (local maximum or dis-similarity) between the adjacent points cannot measure the distinctiveness well. To address this issue, we choose to firstly up-sample the feature maps to have the same spatial resolution as the feature maps at the first level (original resolution), and then calculate the keypoint measure on all up-sampled feature maps. As a result, the multi-level information can be exploited without the negative impact caused by the low resolution. Despite the subtle difference, the performance gain brought by our multi-level is significant, as shown in the experimental results.

To achieve this, we propose a multi-level U-Net structure (MU-Net), in which low-resolution feature maps at each level are up-sampled progressively within a U-Net architecture (Lin et al., 2017), as shown in Fig. 3. In our experiments, we consider three levels, *i.e.,* $X^1 \in \mathbb{R}^{C_1\times H\times W}$, $X^2 \in \mathbb{R}^{C_2\times \frac{H}{2}\times \frac{W}{2}}$, and $X^3 \in \mathbb{R}^{C_3\times \frac{H}{4}\times \frac{W}{4}}$, where C_* denotes the number of channels. We up-sample X^2 and X^3 into the same spatial resolution as X^1 and change the number of feature channels by using the deconvolution operation, and denote the up-sampled features by $X^{1^\uparrow} \in \mathbb{R}^{C_1\times H\times W}$ and $X^{1^{\uparrow\uparrow}} \in \mathbb{R}^{C_1\times H\times W}$, respectively. Then, we calculate the keypoint measure map according to Eqs. 5–7 from X^1, $X^{1^\uparrow}$ and $X^{1^{\uparrow\uparrow}}$, and combine the measures together through the summation operation.

Multi-group detection. The deep features tend to encode specific information/concept (Yang et al., 2020) in different channels. Therefore, it is likely that the similarity between some channels is more related to the cornerness, while others are not very related, such as those encoding the brightness. In other words, it is sufficient to consider the similarity in those channels. Motivated by this, we split the feature maps X into G groups along the channel dimension, each group containing $\frac{C}{G}$ channels. Then we compute the keypoint measure for each group, which is represented as s^g, and take the maximum as the final measure. Therefore, Eq. 7 can be extended to:

$$\begin{aligned} s_{ij} &= \max_{g=1,2,\ldots,G} s^g_{ij} \\ &= \max_{g=1,2,\ldots,G} \alpha^g_{ij}\beta^g_{ij}, \end{aligned} \tag{8}$$

where $*^g_{ij}$ denotes the measure value of point (i, j) at the g^{th} group. We apply Eq. 8 into X^1, $X^{1^\uparrow}$ and $X^{1^{\uparrow\uparrow}}$, respectively. We visualize the measure maps at different groups for a better understanding in the ablations.

3.3 Descriptor Learning

Our work is not only aimed at finding keypoints, but also extracting descriptors for each keypoint. In the standard CNN framework, the learned weights and biases are shared across all spatial locations. Since the images to be matched usually contain contents with different conditions, it is challenging to model representations robust to the complex changes. To

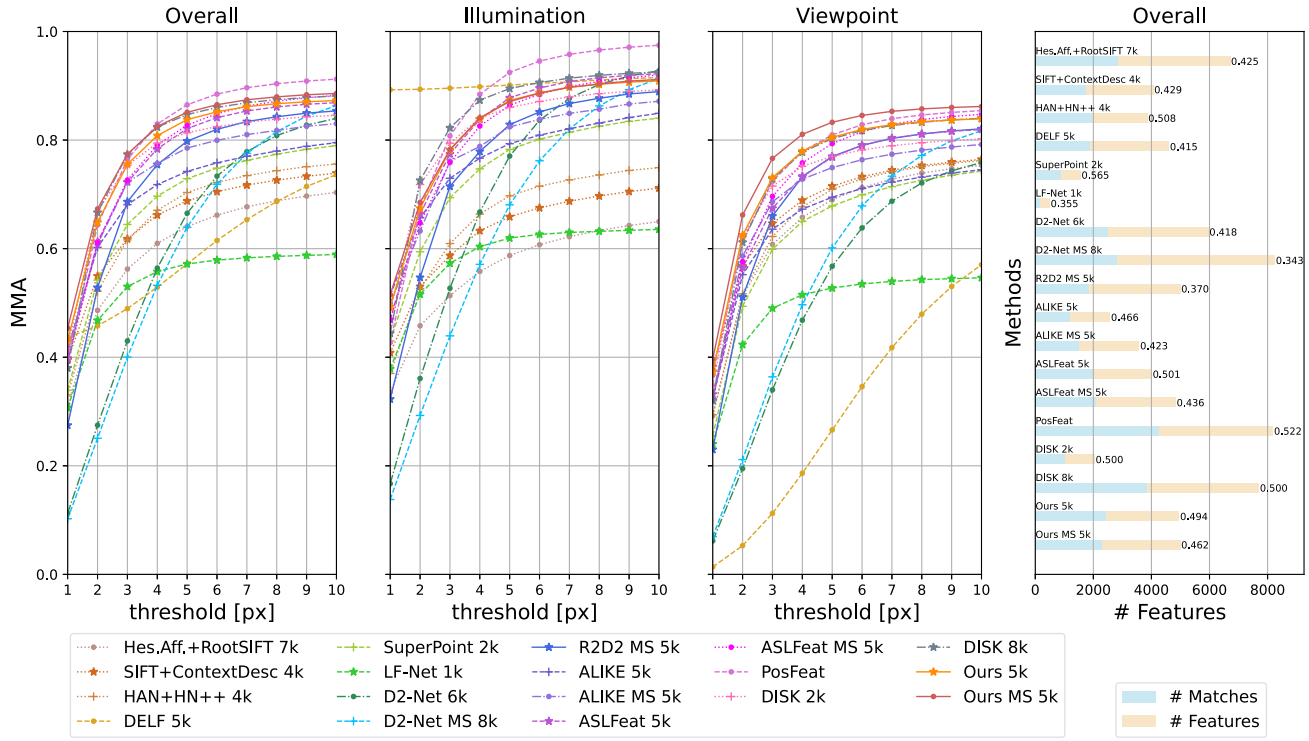

Fig. 4 Experimental results on HPatches (Balntas et al., 2017). We provide the MMA at different error thresholds on the whole dataset, the Illumination set and the Viewpoint set, respectively. We also show the number of features (# Features) and matches (# Matches), and the ratio of # Matches to # Features. The suffix (*k) of the method name indicates the number of detected features. Zoom in for best view (Color figure online)

provide a remedy, we propose the feature self-transformation operation, which transfers the feature representation from the original space into a new space adaptively according to the encoded local information. In detail, a scale factor and an offset factor are first learned adaptively from the feature maps to be transformed. Since the scale factor and offset factor varies from location to location, then the feature extractor has a larger capacity to learn more robust invariant features in a flexible way when we apply the learned scale and offset factors on the corresponding feature maps. For the learned features X, the self-transformation operation is defined as follows:

$$\hat{X} = X \cdot F_\phi(X) + F_\psi(X), \tag{9}$$

where F_ϕ and F_ψ denote two convolutional operations with ReLU activation, respectively, $\hat{X}$ represents the new feature maps, and $\cdot$ denotes the element-wise multiplication. Equation 9 represents that a scale factor $F_\phi(X)$ and an offset factor $F_\psi(X)$ are firstly learned adaptively from the feature itself, which are then used to transform the feature into a new space.

To better transform the features, we exploit the self-transformation in multiple layers. Since here we aim to improve the descriptor, the self-transformation is adopted in the three layers before the learned descriptor, as shown in Fig. 3. Moreover, in our experiments, we find that the detector information is also beneficial to improve the distinctiveness of the descriptor, as the detector can also provide some details for the local information. In detail, as shown in Fig. 3, we exploit the feature self-transformation, which follows the convolutional layer, at the last level. We first encode X^1, $X^{1\uparrow}$ and $X^{1\uparrow\uparrow}$ into a new representation X_{new} by exploiting the concatenation operation and two convolutional operations. Then, for the feature maps X to be transferred, the feature self-transformation in Eq. 9 can be re-written as:

$$\hat{X} = X \cdot F_\phi([X_{new}||X]) + F_\psi([X_{new}||X]), \tag{10}$$

where $[\cdot||\cdot]$ denotes the concatenation operation. In our experiments, we will show the capability of the proposed self-transformation for improving the invariance of the descriptor.

3.4 Implementation

Architecture. The architecture is shown in Fig. 3. The backbone network is similar to that used in L2Net (Tian et al., 2017), ASLFeat (Luo et al., 2020), and R2D2 (Revaud et al., 2019). In detail, the backbone consists of three levels. There

are 2, 2, and 5 convolutional layers at the first, second, and third levels, respectively. The feature maps X^1 and X^2 are the output of the second convolutional layer at the first and second levels, respectively. Since we want to exploit the upsampled feature maps to guide the feature self-transformation in the last convolutional layers, we thus use the output of the second convolution at the third level as X^3 instead of the last one, as shown in Fig. 3. For the outputs of the second to the fourth convolutional layers at the last level, we apply the proposed feature self-transformation operation. $\bar{\mathcal{N}}(i, j)$ in Eq. 5 contains 24 neighbours sampled uniformly from a 9×9 region centered on pixel (i, j) (excluding itself). $\bar{\mathcal{N}}(i, j)$ in Eq. 6 contains 8 neighbours sampled uniformly from a 7×7 region centered on pixel (i, j) (excluding itself), as ASLFeat does. In addition, we set the number of groups to 4. We implement our method using PyTorch.

Training details. Similar to Luo et al. (2020) and Luo et al. (2018), we train our network on around $800k$ image pairs from GL3D (Shen et al., 2018) and (Radenović et al., 2016) containing the ground truth cameras and depths. We train the network from *scratch* with the batch size of 2 and use the SGD optimizer with the momentum of 0.9. We first train the main network without the feature transformation for $400k$ iterations with the initial learning rate of 0.1. Then we train the whole network initialized with the pre-trained weights for another $200k$ iterations with the initial learning rate of 0.01. The training loss is the same as that used in ASLFeat (Luo et al., 2020), *i.e.,* Eqs. 2 and 3, where m_p and m_n are set to 0.2 and 1.0, respectively.

Inference. Following previous works (Dusmanu et al., 2019; Luo et al., 2020), we first exploit a non-maximum suppression sized 3 to filter the keypoints that are adjacent. Then, we use the local refinement (Lowe, 2004) to improve the position of detected key points. Lastly, we extract the descriptors at the refined locations using the bilinear interpolation operation. To address the scale changes, we apply the multi-scale detection (referred to as 'MS' in our experiments) by resizing the image into different resolutions and then detecting keypoints at each scale during testing, as done in previous works (Dusmanu et al., 2019; Luo et al., 2020; Revaud et al., 2019).

4 Experiments

To demonstrate the effectiveness of our method, we provide quantitative comparisons against previous related methods on three tasks, including image matching, 3D reconstruction, and visual localization. We also conduct comprehensive ablation studies to analyze our method.

4.1 Image Matching

Datasets. We consider two datasets, *i.e.,* HPatches (Balntas et al., 2017) and FM-Bench (Bian et al., 2019), for the image matching task.

In the HPatches dataset, there are 116 available image sequences, and we select 108 sequences for evaluation, as done in D2-Net (Dusmanu et al., 2019) and ASLFeat (Luo et al., 2020). Each sequence consists of 6 images, and there exists only **illumination** change in 52 sequences and only **viewpoint** change in the other 56 sequences. Following previous works (Luo et al., 2020; Revaud et al., 2019; Dusmanu et al., 2019), we use three metrics to evaluate our method, including (1) *Repeatability (%Rep)*: the ratio of the number of possible matches found in the two images to the minimum number of detected keypoints in the shared view; (2) *Matching Score (%M.S.)*: the ratio of the number of correct matches found in the image pair and the minimum number of detected keypoints in the shared view; (3) *Mean Matching Accuracy (%MMA)*: the ratio of the number of correct matches to the number of matches found through applying nearest-neighbor search on the descriptors. The 'possible match' in Repeatability indicates that the point distance is below a given threshold after the homography warping. The 'correct match' in Matching Score and Mean Matching Accuracy indicates that the match found through applying a nearest-neighbor search on the descriptors is a 'possible match'.

FM-Bench dataset contains images from four different datasets, including TUM dataset (Sturm et al., 2012), KITTI dataset (Geiger et al., 2012), Tanks and Temples dataset (T&T) (Knapitsch et al., 2017), and Community Photo Collection (CPC) (Wilson & Snavely, 2014). For evaluation, we first estimate the fundamental matrix through keypoints and descriptors extraction, matching by the plain nearest-neighbor search, bad matches rejection (*e.g.,* Lowe's *ratio test* (Lowe, 2004)), and geometric verification (*e.g.,* RANSAC (Fischler & Bolles, 1981)) successively. To measure the estimation accuracy, we compute the Normalized symmetric geometry distance (SGD) (Zhang, 1998) error and classify the estimates with the error below a certain threshold as accurate, as done in FM-Bench. Following (Bian et al., 2019; Luo et al., 2020), we use the *%Recall*, which indicates the ratio of accurate estimates to all estimates, for the overall performance evaluation. In addition, *%Inlier*/*%Inlier-m* is also used to show the matching performance after/before RANSAC, while the correspondence number after/before RANSAC (*%Corr*/*%Corr-m*) is also reported for analysis on the results rather than performance comparison.

Comparisons on HPatches. We report the results of previous approaches which follow the *detect-then-describe* or *describe-and-detect* pipeline. For the former pipeline, we consider (1) Hessian Affine keypoint detector (Mikolajczyk & Schmid, 2004) + RootSIFT descriptor (Arandjelović &

Table 1 Results on FM-Bench (Bian et al., 2019) for pair-wise image matching

Methods	*%Recall* ↑	*%Inlier* ↑	*%Inlier-m* ↑	*#Corrs(-m)*
TUM				
SIFT (Lowe, 2004)	57.40	75.33	59.21	65 (316)
SIFT+HN++ (Mishchuk et al., 2017)	58.90	75.74	62.07	67 (315)
HAN (Mishkin et al., 2018) + HN++	51.70	75.70	62.06	101 (657)
SIFT + ContextDesc (Luo et al., 2019)	59.70	75.53	62.61	69 (325)
LF-Net (MS) (Ono et al., 2018)	53.00	70.97	56.25	143 (851)
D2-Net (MS) (Dusmanu et al., 2019)	34.50	67.61	49.01	74 (1279)
SuperPoint (DeTone et al., 2018)	45.80	72.79	64.06	39 (200)
R2D2 (MS) (Revaud et al., 2019)	57.70	73.70	61.53	260 (1912)
ASLFeat (Luo et al., 2020)	60.20	76.34	69.09	148 (739)
ASLFeat (MS) (Luo et al., 2020)	59.90	76.72	69.50	258 (1332)
FRLNet (Sun et al., 2022b)	61.90	**77.45**	70.43	260 (1485)
Ours	64.10	76.27	69.26	295 (1662)
Ours (MS)	**67.60**	76.73	**70.51**	581 (3137)
KITTI				
SIFT (Lowe, 2004)	91.70	98.20	87.40	154 (525)
SIFT+HN++ (Mishchuk et al., 2017)	92.00	98.21	91.25	159 (535)
HAN (Mishkin et al., 2018) + HN++	90.40	98.09	90.64	233 (1182)
SIFT + ContextDesc (Luo et al., 2019)	92.20	98.23	91.92	160 (541)
LF-Net (MS) (Ono et al., 2018)	80.40	95.38	84.66	202 (1045)
D2-Net (MS) (Dusmanu et al., 2019)	71.40	94.26	73.25	103 (1832)
SuperPoint (DeTone et al., 2018)	86.10	98.11	91.52	73 (392)
R2D2 (MS) (Revaud et al., 2019)	78.80	97.53	86.49	278 (1804)
ASLFeat (Luo et al., 2020)	92.20	98.69	96.25	444 (1457)
ASLFeat (MS) (Luo et al., 2020)	92.20	98.76	96.16	630 (2222)
FRLNet (Sun et al., 2022b)	**92.60**	**99.13**	96.69	642 (2370)
Ours	91.90	98.71	96.99	872 (3012)
Ours (MS)	92.00	98.71	**96.76**	1268 (4756)
T&T				
SIFT (Lowe, 2004)	70.00	75.20	53.25	85 (795)
SIFT + HN++ (Mishchuk et al., 2017)	79.90	81.05	63.61	96 (814)
HAN (Mishkin et al., 2018) + HN++	82.50	84.71	70.29	97 (920)
SIFT + ContextDesc (Luo et al., 2019)	81.60	83.32	69.92	94 (728)
LF-Net (MS) (Ono et al., 2018)	57.40	66.62	60.57	54 (362)
D2-Net (MS) (Dusmanu et al., 2019)	68.40	71.79	55.51	78 (2603)
SuperPoint (DeTone et al., 2018)	81.80	83.87	70.89	52 (535)
R2D2 (MS) (Revaud et al., 2019)	73.00	80.81	65.31	84 (1462)
ASLFeat (Luo et al., 2020)	89.90	85.33	79.08	295 (2066)
ASLFeat (MS) (Luo et al., 2020)	88.70	85.68	79.74	327 (2465)
FRLNet (Sun et al., 2022b)	**91.00**	**86.72**	**82.03**	346 (2501)
Ours	88.90	85.39	79.39	340 (2073)
Ours (MS)	89.00	84.96	79.56	335 (1970)
CPC				
SIFT (Lowe, 2004)	29.20	67.14	48.07	60 (415)
SIFT + HN++ (Mishchuk et al., 2017)	40.30	76.73	62.30	69 (400)
HAN (Mishkin et al., 2018) + HN++	47.40	82.58	72.22	65 (405)
SIFT + ContextDesc (Luo et al., 2019)	41.80	84.01	72.21	61 (306)
LF-Net (MS) (Ono et al., 2018)	19.40	44.27	44.35	50 (114)

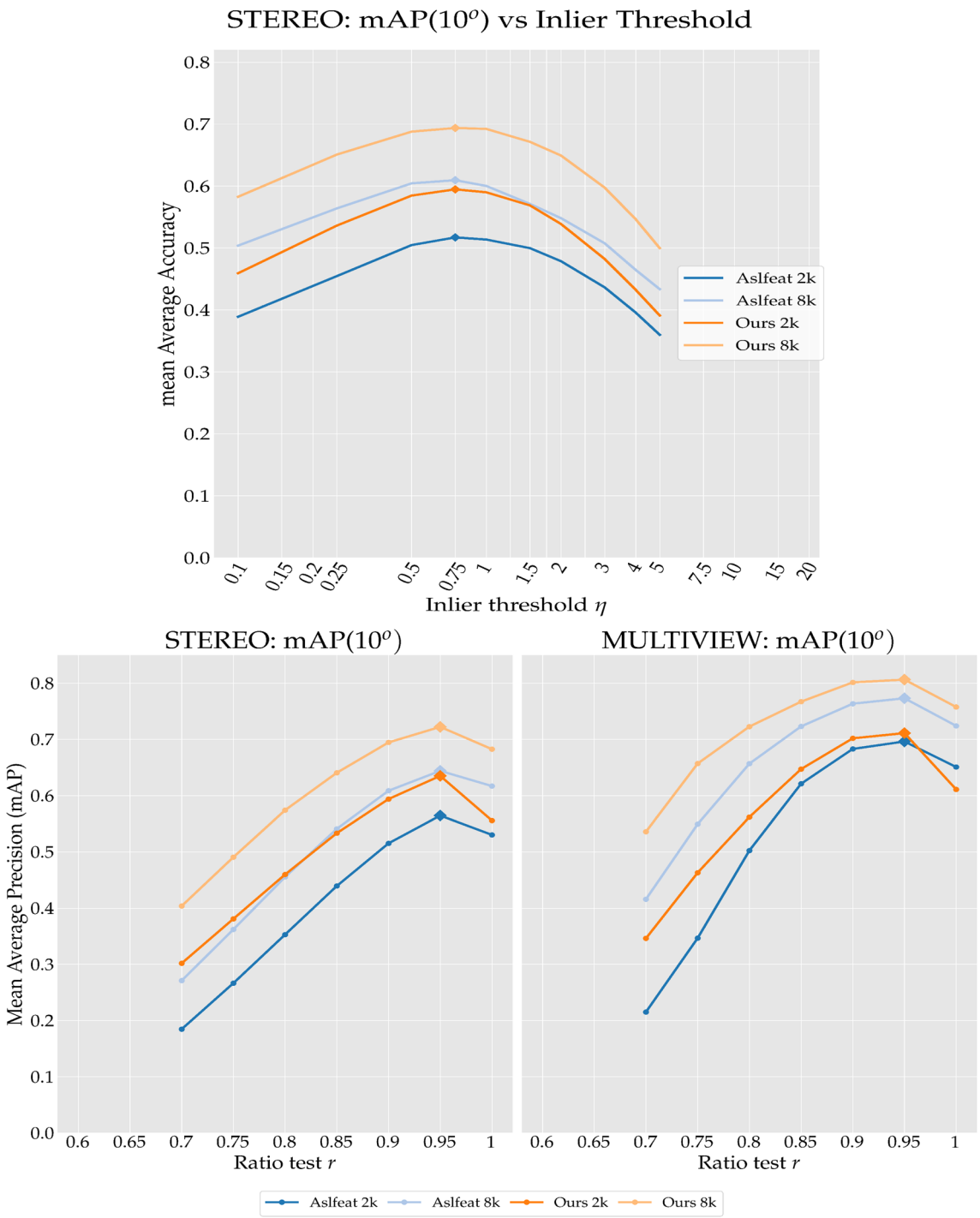

Fig. 5 Comparisons between ours and ASLFeat on IMC benchmark validation set. Our method outperforms ASLFeat *w.r.t.* almost each ratio test threshold and inlier threshold. Up: Results under different inlier thresholds; Bottom: Results under different ratio test thresholds (Color figure online)

Table 1 continued

Methods	*%Recall* ↑	*%Inlier* ↑	*%Inlier-m* ↑	*#Corrs(-m)*
D2-Net (MS) (Dusmanu et al., 2019)	31.30	56.57	49.85	84 (1435)
SuperPoint (DeTone et al., 2018)	40.50	75.28	64.68	31 (225)
R2D2 (MS) (Revaud et al., 2019)	43.00	82.40	67.28	91 (954)
ASLFeat (Luo et al., 2020)	51.50	87.98	82.24	165 (989)
ASLFeat (MS) (Luo et al., 2020)	54.40	89.33	82.76	185 (1159)
FRLNet (Sun et al., 2022b)	54.07	**90.12**	**86.09**	215 (1379)
Ours	57.60	88.93	83.92	228 (1257)
Ours (MS)	**59.00**	89.81	85.28	221 (1237)

The highest score is given in Bold
Our methods outperform most of the existing methods on TUM (Sturm et al., 2012) and CPC (Wilson & Snavely, 2014) while yielding close performance to ASLFeat on T&T (Knapitsch et al., 2017) and KITTI (Geiger et al., 2012). ↑ indicates higher better

Table 2 Results on ETH benchmark (Schonberger et al., 2017) for 3D reconstruction

Methods	*#Reg. image* ↑	*#Sparse poi. (K)*	*%Track len.* ↑	*Reproj. err. (px)* ↓	*#Dense poi. (M)*
Madrid metropolis (1344 images)					
RootSIFT (Arandjelović & Zisserman, 2012; Lowe, 2004)	500	116	6.32	0.60	1.82
GeoDesc (Luo et al., 2018)	495	144	5.97	0.65	1.56
SuperPoint (DeTone et al., 2018)	438	29	9.03	1.02	1.55
D2-Net (MS) (Dusmanu et al., 2019)	495	144	6.39	1.35	1.46
ASLFeat (Luo et al., 2020)	613	96	8.76	0.90	2.00
ASLFeat (MS) (Luo et al., 2020)	649	129	9.56	0.95	1.92
CAPS (Wang et al., 2020)	851	242	6.16	1.03	–
CoAM (Wiles et al., 2021)	702	256	6.09	1.30	–
UP-Net (Yang et al., 2022)	649	153	9.50	0.97	2.02
PoSFeat (Li et al., 2022b)	419	72	9.18	0.86	–

Zisserman, 2012; Lowe, 2004), referred to as Hes.Aff.+ RootSIFT; (2) a learned shape estimator (HesAffNet (Mishkin et al., 2018)) and descriptor (HardNet++ (Mishchuk et al., 2017)), referred to as HAN+HN++; (3) ContextDesc (Luo et al., 2019) with SIFT detector (Lowe, 2004), referred to as SIFT+ContextDesc; (4) LF-Net (Ono et al., 2018), an end-to-end trainable network. For the latter pipeline, we consider D2-Net (Dusmanu et al., 2019), SuperPoint (DeTone et al., 2018), R2D2 (Revaud et al., 2019), DELF (Noh et al., 2017), ASLFeat (Luo et al., 2020), DISK (Tyszkiewicz et al., 2020), ALIKE (Zhao et al., 2022a), and PoSFeat (Li et al., 2022b). Note that, PoSFeat decouples the detector and descriptor by training them in two stages, while others train them together. As shown in Fig. 4, our method with/without multi-scale detection (Ours/Ours MS) yields higher performance than most of the previous methods. Moreover, our methods gain the highest scores on the subset with viewpoint change, especially for matching thresholds below 5 pixels. We can find DELF (Noh et al., 2017) outperforms all the other methods on the subset with illumination change for matching thresholds below 4 pixels. It is because DELF uses a fixed grid of keypoints without further position refinements. This design performs well when there is only illumination change, but it is not robust to viewpoint change, which is universal in real applications. By training the detector and descriptor in two stages, PoSFeat achieves the highest performance.

Comparisons on FM-Bench. As shown in Table 1,[1] our method outperforms most of the existing approaches, especially those also following the *describe-and-detect* pipeline. In comparison with ASLFeat, ours yields close or higher scores.

4.2 3D Reconstruction

We conduct experiments on two datasets, *i.e.,* ETH benchmark (Schonberger et al., 2017) and IMC benchmark (Jin et al., 2021), to evaluate our method for the 3D reconstruction task.

[1] The results are obtained using the code https://github.com/lzx551402/FM-Bench with the default setting.

Table 2 continued

Methods	*#Reg. image* ↑	*#Sparse poi. (K)*	*%Track len.* ↑	*Reproj. err. (px)* ↓	*#Dense poi. (M)*
SCFeat (Sun et al., 2022a)	399	30	10.02	0.84	–
RAP+HardNet (Yan et al., 2022)	434	–	9.06	0.84	1.56
FMT (Jung et al., 2023)	**766**	142	8.13	1.19	–
Ours	649	149	9.37	0.82	1.99
Ours (MS)	653	160	**10.56**	**0.78**	1.94
Gendarmenmarket (1463 images)					
RootSIFT (Arandjelović & Zisserman, 2012; Lowe, 2004)	1035	338	5.52	0.69	4.23
GeoDesc (Luo et al., 2018)	1004	441	5.14	0.73	3.88
SuperPoint (DeTone et al., 2018)	967	93	7.22	1.03	3.81
D2-Net (MS) (Dusmanu et al., 2019)	965	310	5.55	1.28	3.15
ASLFeat (Luo et al., 2020)	1040	221	8.72	1.00	4.01
ASLFeat (MS) (Luo et al., 2020)	1061	320	8.98	1.05	4.00
CAPS (Wang et al., 2020)	1179	627	5.31	1.00	–
CoAM (Wiles et al., 2021)	1072	570	6.60	1.34	–
UP-Net (Yang et al., 2022)	1075	330	8.87	1.11	3.98
PoSFeat (Li et al., 2022b)	956	240	8.40	0.92	–
SCFeat (Sun et al., 2022a)	917	108	9.78	0.94	–
RAP+HardNet (Yan et al., 2022)	999	–	7.80	0.88	4.13
FMT (Jung et al., 2023)	**1316**	516	6.81	1.19	–
Ours	1052	374	9.25	0.87	4.10
Ours (MS)	1073	374	**10.06**	**0.84**	3.84
Tower of London (1576 images)					
RootSIFT (Arandjelović & Zisserman, 2012; Lowe, 2004)	804	239	7.76	0.61	3.05
GeoDesc (Luo et al., 2018)	776	341	6.71	0.63	2.73
SuperPoint (DeTone et al., 2018)	681	52	8.67	0.96	2.77
D2-Net (MS) (Dusmanu et al., 2019)	708	287	5.20	1.34	2.86
ASLFeat (Luo et al., 2020)	821	222	12.52	0.92	3.06
ASLFeat (MS) (Luo et al., 2020)	846	252	13.16	0.95	3.08
CAPS (Wang et al., 2020)	1104	452	5.81	0.98	–
CoAM (Wiles et al., 2021)	804	239	5.82	1.32	–
UP-Net (Yang et al., 2022)	832	245	13.27	0.89	3.15
PoSFeat (Li et al., 2022b)	778	262	11.64	0.90	–
SCFeat (Sun et al., 2022a)	657	108	11.62	**0.79**	–
RAP+HardNet (Yan et al., 2022)	700	–	10.84	0.82	2.75
FMT (Jung et al., 2023)	**1186**	315	8.63	1.21	–
Ours	873	290	12.44	0.83	3.19
Ours (MS)	907	272	**13.60**	**0.79**	3.20

Bold values indicate the highest performance among the recent approaches
↑ indicates higher better, while ↓ indicates lower better

Evaluation on ETH Benchmark. Following (Luo et al., 2020; Dusmanu et al., 2019), we conduct the evaluation on three medium-scale internet-collected datasets from the ETH benchmark (Schonberger et al., 2017), and make comparisons against several existing approaches, including ASLFeat (Luo et al., 2020), D2-Net (Dusmanu et al., 2019), SuperPoint (DeTone et al., 2018), GeoDesc (Luo et al., 2018), RootSIFT (Arandjelović & Zisserman, 2012; Lowe, 2004), CAPS (Wang et al., 2020), CoAM (Wiles et al., 2021), UP-Net (Yang et al., 2022), PoSFeat (Li et al., 2022b), SCFeat (Sun et al., 2022a), RAP+HardNet (Yan et al., 2022), and FMT (Jung et al., 2023).

For evaluation, we first perform exhaustive image matching with both ratio test at 0.8 and mutual check for outlier

Table 3 Results on IMC benchmark (Jin et al., 2021) for 3D reconstruction

Methods	Up to 2048 features per image							Up to 8000 features per image						
	Task 1: Stereo			Task 2: Multiview				Task 1: Stereo			Task 2: Multiview			
	NM	NI	mAA(10°)	NM	NL	TL	mAA(10°)	NM	NI	mAA(10°)	NM	NL	TL	mAA(10°)
SuperPoint	292.8	126.8	0.2964	169.3	1184.3	4.34	0.5464	–	–	–	–	–	–	–
LF-Net	191.1	106.5	0.2344	196.7	1385.0	4.14	0.5141	–	–	–	–	–	–	–
D2-Net (SS)	505.7	188.4	0.1813	513.1	2357.9	3.39	0.3943	1258.2	482.3	0.2228	1278.7	5893.8	3.62	0.4598
D2-Net (MS)	327.8	134.8	0.1355	337.6	2177.3	3.01	0.3007	1028.6	470.6	0.2506	1054.7	6759.3	3.39	0.4751
R2D2	273.6	213.9	0.3346	280.8	1228.4	4.29	0.6149	1408.8	842.2	0.4437	739.8	4432.9	4.59	0.6832
ASLFeat (MS)	–	–	–	–	–	–	–	805.5	390.8	0.4610	510.8	3817.1	4.58	0.6825
ALIKE	289.4	222.6	0.4958	298.3	1693.3	5.02	0.7022	440.1	338.3	0.5031	450.1	2655.5	5.08	0.7071
PoSFeat	430.2	348.3	0.4624	442.6	2311.5	5.11	0.7069	1070.3	904.1	0.4822	1106.3	6474.5	5.42	0.7192
DISK	514.2	404.2	**0.5132**	527.5	2428.0	5.55	**0.7271**	1621.9	1238.5	0.5585	1663.8	7484.0	5.92	0.7502
LoFTR_v4*	–	–	–	–	–	–	–	–	737.9	0.6091	741.4	7816.4	4.53	0.7610
SDS*	–	–	–	–	–	–	–	–	1707.2	**0.6398**	1739.7	8924.4	5.37	**0.7856**
Ours (MS)	257.0	187.1	0.4318	262.2	1474.6	4.49	0.6506	939.0	699.2	0.5154	958.7	5176.4	4.85	0.7156

Bold values indicate the highest mAA score

LoFTR_v4* is the latest performance of LoFTR while SDS* yielding the highest score in the benchmark website is a combination of SuperPoint, DISK, and SuperGlue. We provide these two methods for reference

Table 4 Results on Aachen Day-Night dataset for visual localization

Methods	#Feats	Dim.	Percentage of correction		
			0.25m, 2°	0.5m, 5°	5m, 10°
Aachen Day-Night v1.0					
R2D2* (Revaud et al., 2019)	5k	128	71.4	88.1	98.3
R2D2* (Revaud et al., 2019)	10k	128	**77.2**	86.0	99.3
D2-Net (Dusmanu et al., 2019)	5k	512	68.7	87.4	99.0
D2-Net (Dusmanu et al., 2019)	10k	512	73.8	**89.5**	**100.0**
ASLFeat (Luo et al., 2020)	5k	128	66.7	82.3	94.9
ASLFeat (Luo et al., 2020)	10k	128	76.5	87.1	99.0
Ours	5k	128	69.7	82.7	94.9
Ours	10k	128	75.5	86.8	99.0
Aachen Day-Night v1.1					
R2D2* (Revaud et al., 2019)	5k	128	69.0	**85.0**	96.0
R2D2* (Revaud et al., 2019)	10k	128	67.3	84.3	**97.9**
D2-Net (Dusmanu et al., 2019)	5k	512	62.0	83.2	95.8
D2-Net (Dusmanu et al., 2019)	10k	512	66.5	**85.0**	96.7
ASLFeat (Luo et al., 2020)	5k	128	64.1	79.4	94.4
ASLFeat (Luo et al., 2020)	10k	128	**70.0**	84.7	96.7
Ours	5k	128	63.4	79.6	94.2
Ours	10k	128	67.5	84.8	96.3

Bold values indicate the best localization score among the compared methods
Note that, R2D2 is trained on this dataset

Table 5 Comparisons with SOTA methods on Aachen Day-Night dataset v1.1

Methods	Percentage of correction		
	0.25m, 2°	0.5m, 5°	5m, 10°
R2D2 (Revaud et al., 2019)	68.1	83.8	96.9
ASLFeat (Luo et al., 2020)	72.8	87.4	97.4
DISK (Tyszkiewicz et al., 2020)	73.3	86.9	97.9
PoSFeat (Li et al., 2022b)	73.8	87.4	98.4
SS (Sarlin et al., 2020)	73.3	88.0	98.4
SP (Revaud et al., 2022)	74.4	88.0	98.4
LoFTR (Sun et al., 2021)	**78.5**	**90.6**	**99.0**
Ours	71.3	87.4	97.4

Bold values indicate the best localization score among the compared methods
'SS' indicates the combination of SuperPoint and SuperGlue, one for detecting and representing and one for matching; 'SP' indicates the combination of SuperPoint and PUMP, one for detecting and one for representing; LoFTR is an efficient end-to-end matching method

rejection. Following the protocol defined by (Schonberger et al., 2017), we run SfM (Schonberger & Frahm, 2016) for sparse reconstruction and MVS (Schönberger et al., 2016) for dense reconstruction. For the former task, we report the number of registered images (referred to as *#Reg. Images*), the number of sparse points (*#Sparse Poi.*), the mean track length of the 3D points (*Track Len.*) and the mean reprojection error (*Reproj. Err.*). For the latter task, we report the number of dense points (*#Dense Poi.*). Both ASLFeat (Luo et al., 2020) and ours limit the maximum number of keypoints to 20k. We report the results obtained using single-scale detection and multi-scale detection. As shown in Table 2, our method performs better than most of the other methods. In comparison to ASLFeat (Luo et al., 2020), our model yields lower reprojection error, which demonstrates the effectiveness of our method for 3D reconstruction.

Evaluation on IMC. The Image Matching Challenge 2020 benchmark (IMC) provides a dataset with thousands of phototourism images of 25 landmarks, which are taken from diverse viewpoints, with different cameras, in different illumination and weather conditions. For evaluation, this benchmark provides two tasks, *i.e.,* stereo and multiview

Table 6 Analysis about the main components on HPatches (Balntas et al., 2017)

Ablation	Config	*%Rep*	*%M.S.*	*%MMA*
	F.M	63.4	37.2	68.5
Similarity	S.M	67.2	39.8	72.5
based	S.M. (α)	64.6	37.9	71.5
measure	S.M. (β)	65.1	37.0	70.4
	F.M. (1^{st})	67.8	40.2	73.4
	F.M. (3^{rd})	32.5	24.9	48.8
	S.M. (1^{st})	68.5	40.5	74.4
Multi-	S.M. (3^{rd})	31.5	20.3	41.1
level	F.M. (U-Net)	62.6	28.8	72.6
detection	S.M. (U-Net)	66.2	38.3	72.2
	F.M. (MU-Net)	69.9	37.7	75.2
	S.M. (MU-Net)	70.2	38.6	76.4
Feature	F.T	70.4	40.4	77.3
	F.T. (L2)	70.2	39.4	77.1
transformation	F.T. (L1)	70.2	39.1	76.9
Multi-	G=1	70.4	40.4	77.3
group	G=2	70.4	40.3	77.3
measure	G=4	71.0	40.2	77.5
Comp. with	ASLFeat	68.1	39.8	72.5
ASLFeat	Ours	71.0	40.2	77.5

F.M.: feature-based peakiness measure (Luo et al., 2020); S.M.: similarity-based measure (Ours); F.T.: feature self-transformation

reconstruction, where the reconstructed poses are compared to the ground truth. In the stereo task, we first extract local features across every pair of images and then use RANSAC to reconstruct the relative pose, while in multiview task, we use COLMAP (Schonberger & Frahm, 2016) to reconstruct the pose from small subsets of 5, 10, and 25 images.

Table 8 Comparison with D2D measure

Method	*%Rep*	*%M.S.*	*%MMA*
D2D	60.9	31.5	65.4
Ours	70.2	38.6	76.4
D2D+F.T	62.5	32.4	66.3
Ours+F.T	70.4	40.4	77.3

F.T.: feature self-transformation

According to the benchmark documentation, we consider two categories, a limited budget of 2048 keypoints and a limited budget of 8000 keypoints. We select the hyperparameters on the validation set of three scenes, including "Reichstag", "Sacre Coeur", and "St. Peter's Square". Specifically, we set the ratio test threshold to 0.95, and use DEGENSAC (Chum et al., 2005) with an inlier threshold of 0.75 pixels for the stereo task. With the selected hyperparameters, we submit the extracted features from nine test sets to the website and report the results in Table 3. We compare our method with several previous describe-and-detect methods, like SuperPoint (DeTone et al., 2018), LF-Net (Ono et al., 2018), D2-Net (Dusmanu et al., 2019), R2D2 (Revaud et al., 2019), ASLFeat (Luo et al., 2020), ALIKE (Zhao et al., 2022a), DISK (Tyszkiewicz et al., 2020), and PoSFeat (Li et al., 2022b). We also provide the performance of two state-of-the-art methods, LoFTR (Sun et al., 2021) (an end-to-end image matching method) and SDS (a combination of SuperPoint (DeTone et al., 2018), DISK (Tyszkiewicz et al., 2020), and SuperGlue (Sarlin et al., 2020).) The results are reported in the benchmark website or (Jin et al., 2021). The most important metric is the mean Average Accuracy (mAA) up to a 10-degree error threshold. Our method does not perform better than recent methods, like DISK and PoSFeat,

Table 7 Performance on illumination and viewpoint subsets of HPatches (Balntas et al., 2017)

Methods	Illumination			Viewpoint		
	%Rep	*%M.S.*	*%MMA*	*%Rep*	*%M.S.*	*%MMA*
Without F.T. (3pix)	69.5	41.4	78.2	70.7	36.0	74.6
With F.T. (3pix)	69.6	43.2	78.3	71.1	37.8	76.4
Without F.T. (4pix)	75.9	43.9	83.5	77.3	37.9	79.0
With F.T. (4pix)	76.2	46.2	84.0	77.9	39.9	80.9
Without F.T. (5pix)	80.4	45.1	86.1	81.8	38.8	81.1
With F.T. (5pix)	80.7	47.7	87.1	82.1	40.9	83.2
Without F.T. (6pix)	83.3	45.6	87.3	85.0	39.3	82.3
With F.T. (6pix)	83.6	48.4	88.7	85.3	41.5	84.5
Without F.T. (7pix)	85.7	46.0	88.1	87.5	39.6	82.9
With F.T. (7pix)	86.0	48.9	89.9	87.7	41.9	85.3
Without F.T. (8pix)	87.5	46.1	88.6	89.4	39.8	83.3
With F.T. (8pix)	87.8	49.2	90.5	89.6	42.1	85.7

We set the threshold for 'possible match' to 3, 4, 5, 6, 7, and 8 pixels, respectively. F.T. refers to feature self-transformation

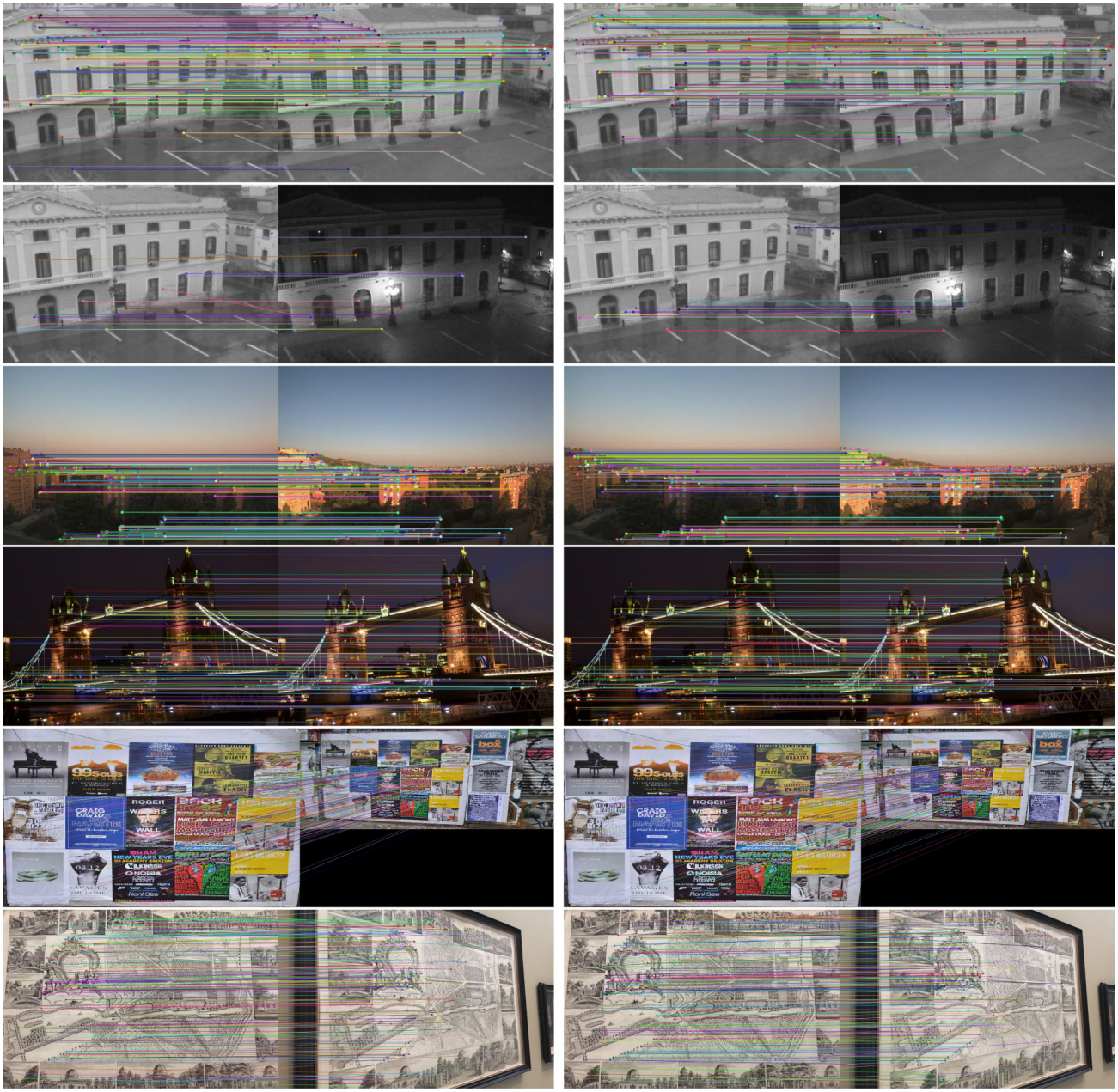

Fig. 6 Some visualization examples on HPatches. We can find that when the illumination change is not severe (the first, third, and fourth examples), two methods perform well, while when there exists severe illumination change (the second example), both perform worse but the method without F.T. generates obvious incorrect matches. For viewpoint change, in the fifth example, incorrect matches (the poster with "BOX") are also found in the result of the baseline without F.T. We only show up to 200 matches. Zoom in for more details (in color) (Color figure online)

while yielding highest mAA scores among R2D2, D2-Net, and ASLFeat.

To better compare our method with ASLFeat, we also provide the results under different ratio test thresholds and inlier thresholds on the IMC validation set. As shown in Fig. 5, we can find that in both categories (2048 and 8000 features), our method performs better than ASLFeat almost for each ratio test threshold and inlier threshold.

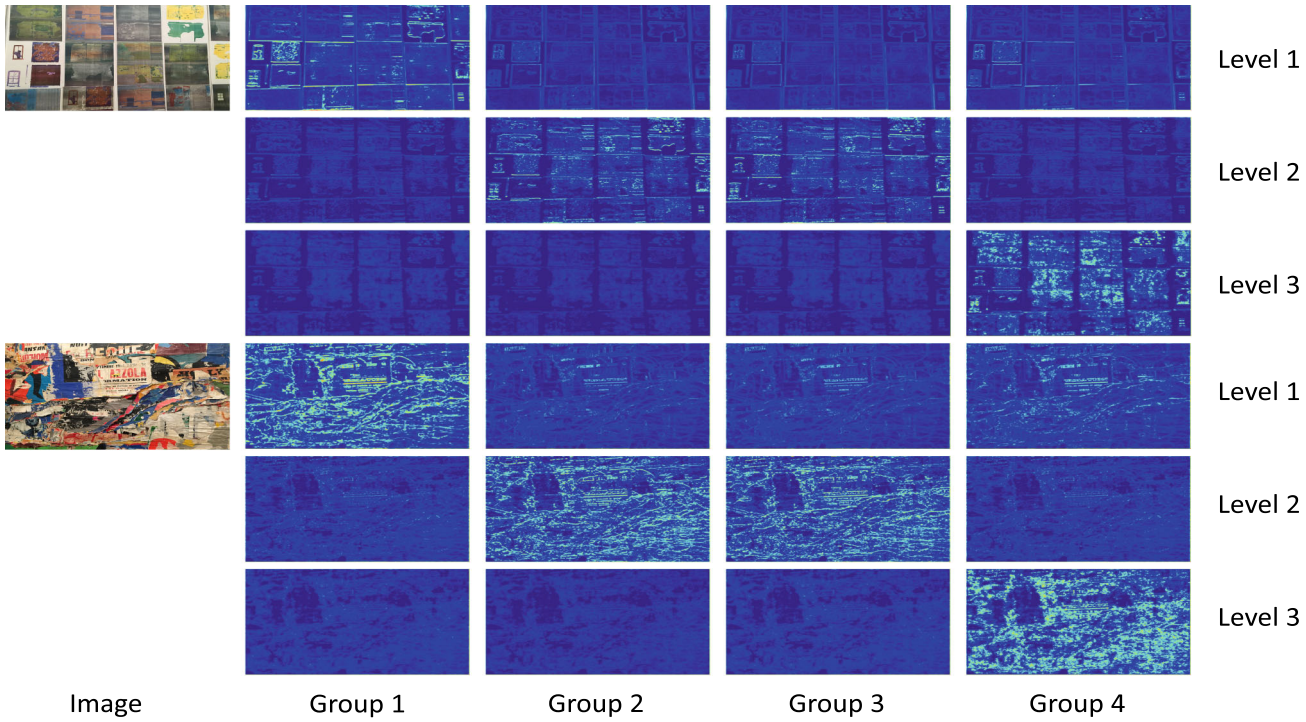

Fig. 7 The learned distinctiveness maps at different levels and groups (Color figure online)

4.3 Visual Localization

Here, we evaluate our method's performance in the visual localization task on the Aachen Day-Night dataset v1.0 and v1.1 (Sattler et al., 2012; Zhang et al., 2021), where the objects are matching images with extreme day-night changes. We first use the compared methods to generate the localization and description of the keypoints respectively, and then use the code from Sattler et al. (2012) for image registration. We limit the maximum feature number of all methods to 5000 and 10000, respectively. Through submitting the results to the benchmark, we can get the percentages of successfully localized night-time images within three given error bounds, *i.e.,* $(0.25m, 2°)$, $(0.5m, 5°)$, and $(5m, 10°)$. We make comparisons against R2D2 (Revaud et al., 2019), D2-Net (Dusmanu et al., 2019), and ASLFeat (Luo et al., 2020), whose codes are publicly available. We repeat the experiments three times and report the average results. Due to the extreme illumination changes, it is challenging to match the night images to the day images, which all methods do not address well. As shown in Table 4, our method yields comparable scores to these three previous methods, especially ASLFeat.

In Table 5, we further make comparisons with other state-of-the-art (SOTA) methods, including DISK (Tyszkiewicz et al., 2020), PoSFeat (Li et al., 2022b), SuperPoint (DeTone et al., 2018)+SuperGlue (Sarlin et al., 2020) (SS), SuperPoint+PUMP (Revaud et al., 2022) (SP), and an end-to-end matching method, LoFTR (Sun et al., 2021). The maximum number of keypoints is limited to 20000. We can find that our method achieves close performance to other describe-and-detect methods, *i.e.,* ASLFeat, DISK, and PoSFeat.

4.4 Ablation Studies

Here, we analyze the main components in our method, including the similarity-based measure, multi-level U-Net structure, feature self-transformation, and multi-group keypoint measure, on HPatches dataset (Balntas et al., 2017). We report the *%Rep*, *%M.S.*, and *%MMA* in Table 6. Here, we select up to 5000 keypoints with the keypoint measure over 0.5 and set the threshold for 'possible match' to three pixels. *Similarity-based Measure*. First, we train the baseline network (no feature self-transformation and no MU-Net) using the feature-based measure (*abbr.* F.M.) in ASLFeat (Luo et al., 2020) and the proposed similarity-based measure (*abbr.* S.M.), respectively. Specifically, as done in (Luo et al., 2020), we compute the detection map on three feature maps coming from three levels. As shown in Table 6, S.M. yields higher scores than F.M. *w.r.t.* all three metrics, which shows the superiority of the proposed similarity-based keypoint measure. We also evaluate the two requirements of the detector, *i.e.,* local distinctiveness (α) and local maximum of distinctiveness (β). The comparisons between S.M. (α), S.M. (β), S.M., and F.M. show that any one of the requirements (α and β) can perform better than feature-based peakiness measure, and the joint modelling can bring further improvements.

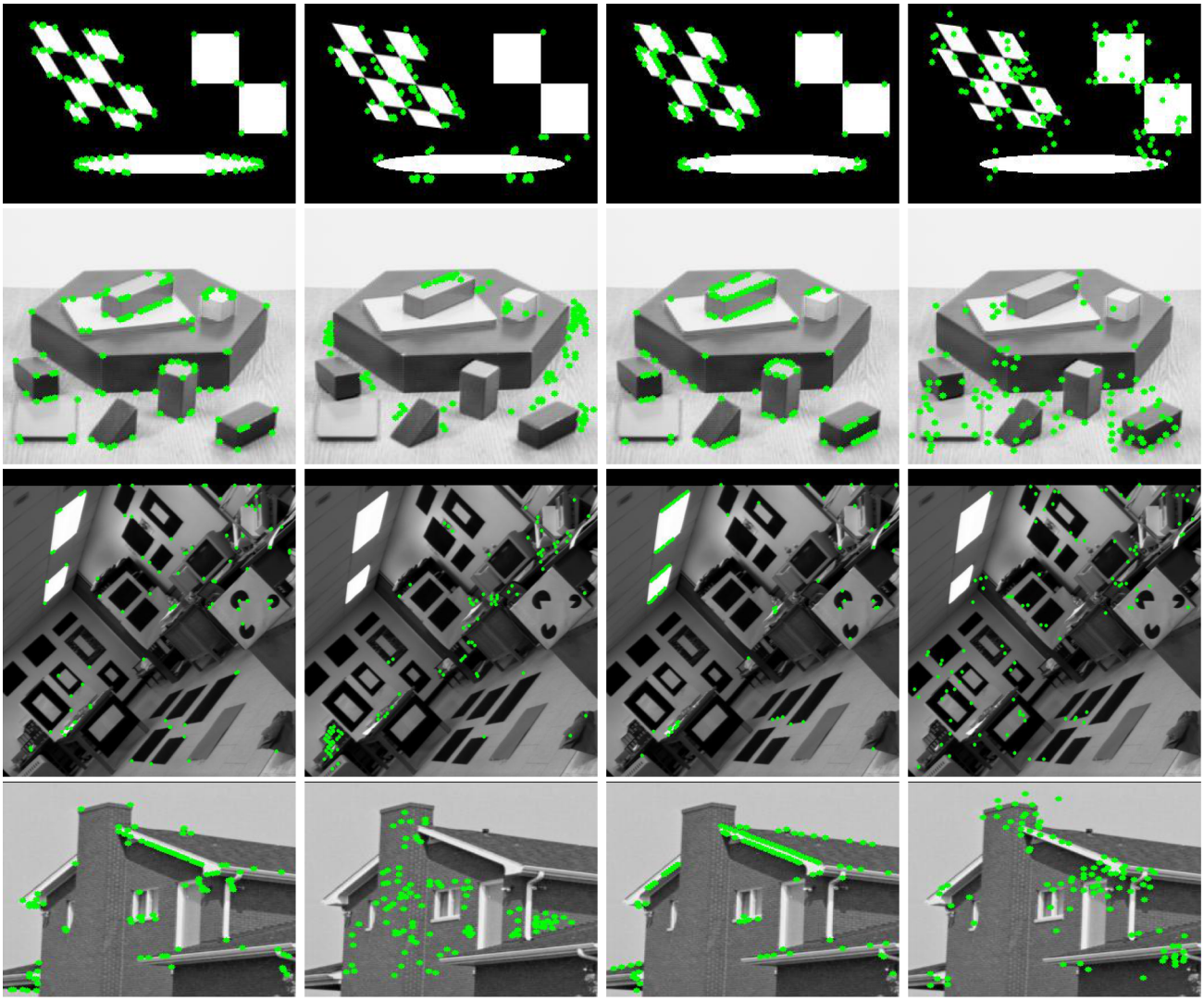

Fig. 8 Detection results on simple scenes. The first is from skimage [6]; The last three come from Shui and Zhang (2013) .[7] We limit the maximum number of keypoitns to 100

Multi-level Detection. We study different multi-level detection strategies, including U-Net (only up-sampling feature maps at the last layer), multi-level detection in ASLFeat, and our MU-Net. Firstly, we provide the performance of single-level detection by calculating the measures from the first (*i.e.,* the original resolution) and third level, respectively. As shown in Table 6, we can find that both F.M. and S.M. performs better at the original scale (marked by 1^{st}) than the third level (3^{rd}) and even better than the multi-level detection proposed in ASLFeat according to the comparisons of F.M/S.M. and F.M. (1^{st})/S.M. (1^{st}). It means that the measure map calculated from the feature maps with low resolution does not reflect the distinctiveness of keypoints. By introducing the U-Net structure to up-sample the feature maps at the third level to the original scale, the performance is improved greatly, *e.g.,* F.M. (3^{rd}) *v.s.* F.M. (U-Net) and S.M. (3^{rd}) *v.s.* S.M. (U-Net), which can also support the analysis. Therefore, to exploit the multi-level information and avoid losing detailed information, we introduce a multi-level U-Net architecture to better achieve multi-level detection. The performance improvements brought by MU-Net on both F.M. and S.M. show the effectiveness of the proposed multi-level detection strategy.

Feature self-transformation. Based on the network with the multi-level U-Net, *i.e.,* S.M. (MU-Net), we add the proposed feature self-transformation operation (referred as F.T.), as shown in Fig. 3. The comparison between F.T. and S.M. (MU-Net) demonstrates that the feature transformation is able to improve the discriminative power of the descriptors. In our experiments, we deploy the feature transformation in the

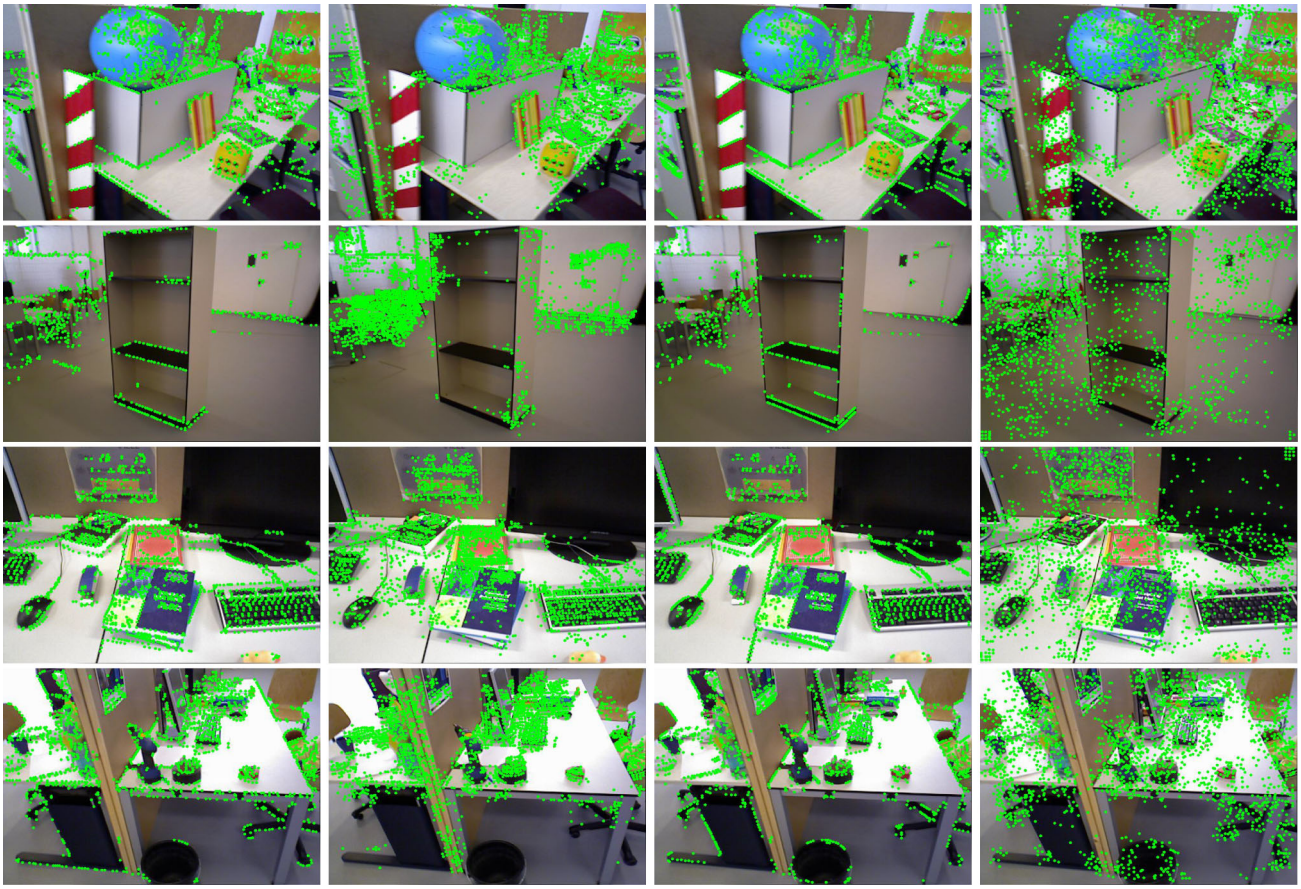

Fig. 9 Detection results on indoor scenes. We limit the maximum number of keypoints to 2000

last three convolution layers. Here, we also try to deploy the transformation in the last two (marked by L2) and one (L1), respectively, but observe the performance drops. Nevertheless, all three models, *i.e.,* F.T., F.T. (L2), and F.T. (L1), perform better than S.M. (MU-Net), which demonstrates the effectiveness of the proposed feature self-transformation. Considering that the improvements brought by the transformation might result from the increase of convolutional parameters, we replace the transformation operation with (1) two successive convolutions and (2) a residual block with the same number of parameters as the transformation. We do not find any meaningful improvements in comparison with S.M. (MU-Net). For example, with the first replacement, we only observe a 0.02% improvement (76.43% to 76.45%) for $\%MMA$, while our F.T. improves S.M. (MU-Net) by 0.9%.

Furthermore, as the HPatches (Balntas et al., 2017) dataset consists of two subsets, one containing illumination changes and one containing viewpoint changes, here we provide the improvements brought by the proposed feature self-transformation for the two scenarios separately. Table 7 shows that the proposed feature transformation can bring improvements *w.r.t* $\%MMA$ for both viewpoint change and illumination change, in comparison with the baseline without feature self-transformation. Since most of the illumination changes in HPatches dataset are not severe, previous methods yield higher performance on the illumination subset than the viewpoint subset, as shown in Fig. 4. Compared with these methods, our model brings fewer improvements on the illumination subset than the viewpoint subset but still performs better than previous methods on both subsets.

We also provide some visualization examples in Fig. 6. We can find that both methods do not perform well when there exists remarkable illumination change. In addition, in Table 7, we can observe that the lower $\%Rep$ are acquired on the illumination subset, since the illumination might make the images low-quality, which is a challenging scenario to be addressed in the future.

Multi-group Measure. We set the number of groups (G) in the S.M. (MU-Net) with feature self-transformation to 1 (*i.e.,* F.T.), 2, and 4, respectively. As shown in Table 6, we can find that a slight improvement can be brought by increasing the number of groups. Here, We also provide a visualization example to illustrate the learned score maps at different levels and groups. As shown in Fig. 7 where 'Level 1', 'Level

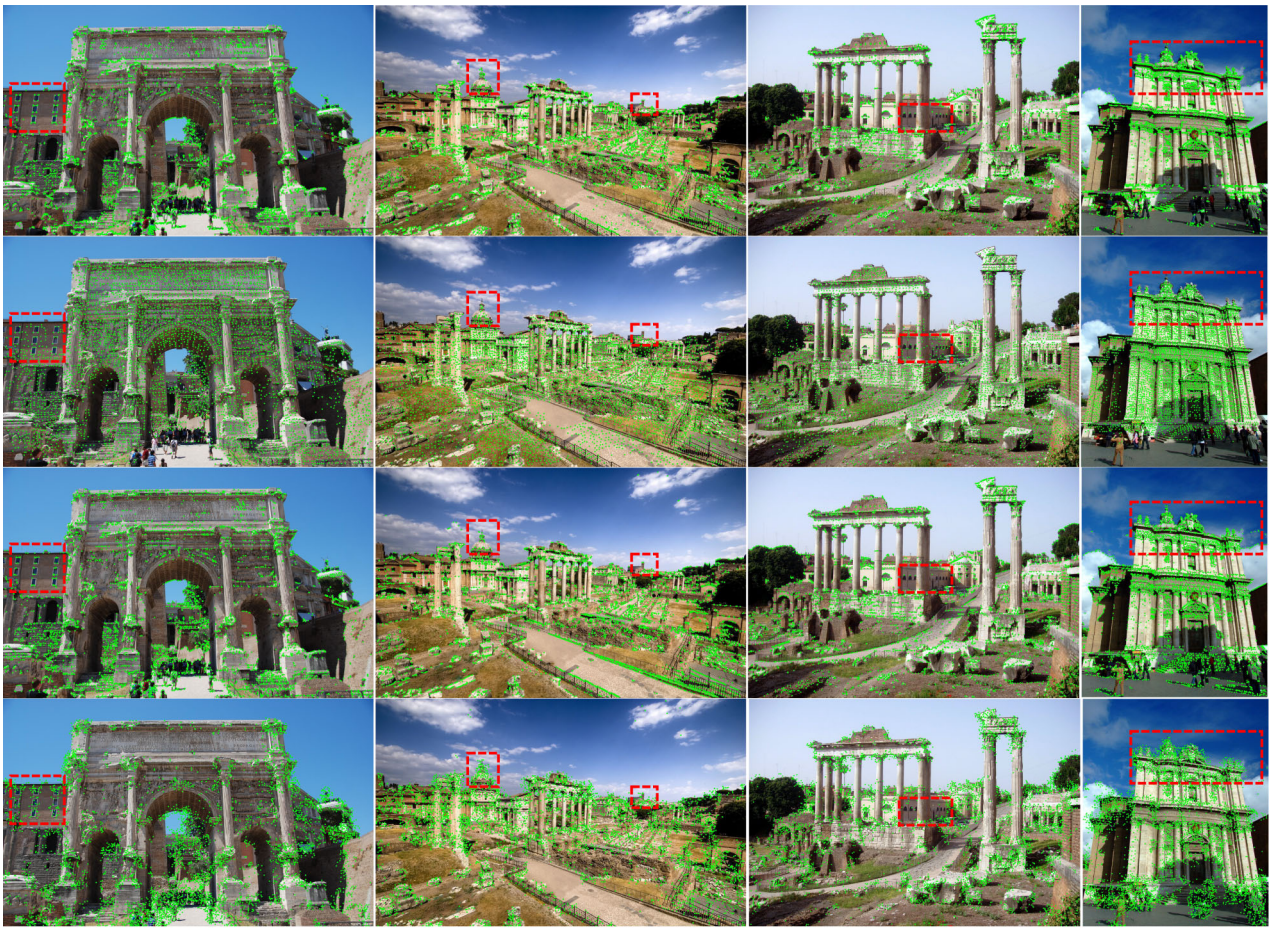

Fig. 10 Detection results on outdoor scenes. We limit the maximum number of keypoints to 5000. We highlight the main differences with a red rectangular. Zoom in for best view (Color figure online)

2', and 'Level 3' correspond the X^1, $X^{1\uparrow}$, and $X^{1\uparrow\uparrow}$ in Fig. 3, respectively, we can find that the distinctiveness map varies from the level and group. For example, at the first level, the similarity in the first group is more important, while the similarity in the last group provides more information for the detector at the third level.

According to the analysis above, we can conclude that (1) the proposed keypoint measure and multi-level detection boost $\%Rep$ and $\%MMA$ remarkably; (2) the feature self-transformation mainly improves the discriminative capability of the descriptors ($\%MMA$); (3) the multi-group measure further refines $\%Rep$ and $\%MMA$ without increasing the number of learned parameters but slightly. Lastly, we also make a comparison against ASLFeat (Luo et al., 2020). As shown in Table 6, our model outperforms ASLFeat *w.r.t* all there metrics, especially *%Rep* and *%MMA*.

In addition, relying on traditional keypoint detection strategies, a previous method, D2D (Tian et al., 2020a), also defines a keypoint measure containing two terms, one for absolute saliency and the other for relative saliency. However, it straightforwardly applies the defined measure on the deep feature maps extracted from a pre-trained descriptor model without extra training. The reported results in the paper show that the performance greatly depends on the pre-trained descriptor models. In contrast, we learn the detector and descriptor jointly in an end-to-end manner. Here, we also use the measure in D2D to replace our keypoint measure, and report the results in Table 8. We can find that the D2D measure which relies on a pre-trained descriptor yields lower scores than ours.

4.5 Visualization Examples

In this part, we first visualize the keypoints detected by our method, DISK (Tyszkiewicz et al., 2020), ASLFeat (Luo et al., 2020), and D2-Net (Dusmanu et al., 2019), respectively. In Fig. 8, we show four samples containing simple contents. We can find that compared with the other three methods, ours can find the corner points better. In Figs. 9 and 10, we provide several examples (Bian et al., 2019) with more complex contents. We can observe that D2-Net generates worse detection results than others. The regions marked

Fig. 11 Visualization examples with view changes. Best viewed in color (Color figure online)

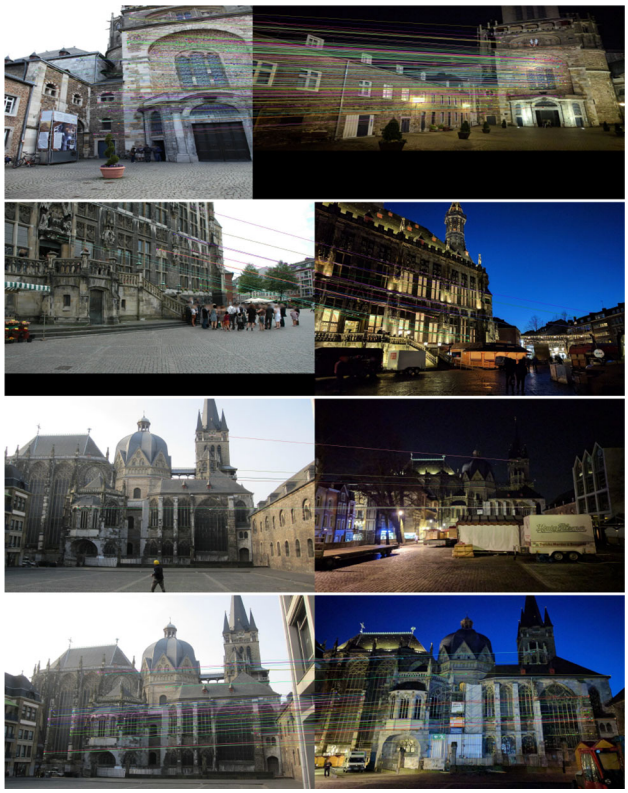

Fig. 12 Visualization examples with both view and illumination changes. Best viewed in color (Color figure online)

by the red rectangular in Fig. 10 illustrate that our method can yield more accurate corners than ASLFeat, especially for the Windows in the first and third examples. We can surprisingly find that DISK is able to detect the points which are very useful for image matching while ignoring the useless. For example, in the first and fourth samples, DISK does not detect the keypoints in the regions of Pedestrian which can be considered as needless information for matching the building, while the other three methods do. We attribute the advantages to the efficient learning strategy.

We then provide several matching examples with the view change in Fig. 11 and several examples with both view and illumination changes in Fig. 12. In each example, we only visualize up to 200 matches. The images come from Aachen Day-Night dataset (Sattler et al., 2012). We can find that our method is capable of addressing the view changes effectively. However, we can also observe that when there exist serious illumination changes and the illumination makes the images low-quality, it is quite challenging for our method to find enough correct matches.

5 Conclusion and Discussion

This paper aims at learning the local feature detector and descriptor jointly, following the *describe-and-detect* pipeline. To achieve that, we propose a new method called Deep Corner, which is inspired the the tradition corner detection methods. Specifically, we first propose the similarity-based measure for keypoint detection, which is able to select repeatable keypoints effectively and thus beneficial for the learning of descriptor. Additionally, to improve the keypoint localization accuracy, we further design a MU-Net structure for multi-level detection and extend the proposed measure into the multi-group version. Finally, we propose a feature self-transformation operation to improve the invariance of the descriptors. Experimental comparisons with previous related

methods and ablation studies demonstrate the effectiveness of our method.

Limitation: According to the results in Fig. 4 and Table 4, the performance advantage over previous works yielded by our method for illumination change is not as high as for viewpoint change. In fact, the severe illumination change is intractable to address, which is also analyzed in previous sections. For example, the night images have low quality compared with the day images, and therefore, it is hard to detect keypoints and extract distinguishable representations. In the future, maybe we could address this issue by exploiting the image translation techniques and/or improving the normalization operation.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions.

Data Availability The datasets generated during and/or analysed during the current study are available publicly: 1. GL3D (Shen et al., 2018): https://github.com/lzx551402/GL3D 2. HPatches (Balntas et al., 2017): http://icvl.ee.ic.ac.uk/vbalnt/hpatches/ 3. FM-Bench (Bian et al., 2019): https://github.com/JiawangBian/FM-Bench 4. ETH Benchmark (Schonberger et al., 2017): http://landmark.cs.cornell.edu/projects/1dsfm/images.Tower_of_London.tar; http://landmark.cs.cornell.edu/projects/1dsfm/images.Gendarmenmarkt.tar; http://landmark.cs.cornell.edu/projects/1dsfm/images.Madrid_Metropolis.tar; 5. IMC Benchmark (Jin et al., 2021): https://www.cs.ubc.ca/research/image-matching-challenge/2020/data/ 6. Aachen Day-Night (Sattler et al., 2012; Zhang et al., 2021): https://www.visuallocalization.net/datasets/

References

Arandjelović, R., & Zisserman, A. (2012). Three things everyone should know to improve object retrieval. In *2012 IEEE conference on computer vision and pattern recognition* IEEE (pp. 2911–2918).

Balntas, V., Lenc, K., Vedaldi, A., & Mikolajczyk, K. (2017). Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5173–5182).

Balntas, V., Riba, E., Ponsa, D., & Mikolajczyk, K. (2016). Learning local feature descriptors with triplets and shallow convolutional neural networks. In *Bmvc* vol. 1 (p. 3).

Barroso-Laguna, A., Riba, E., Ponsa, D., & Mikolajczyk, K. (2019). Key.net: Keypoint detection by handcrafted and learned cnn filters. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*

Barroso-Laguna, A., Verdie, Y., Busam, B., & Mikolajczyk, K. (2020). Hdd-net: Hybrid detector descriptor with mutual interactive learning. In *Proceedings of the Asian conference on computer vision*.

Bay, H., Tuytelaars, T., & Van Gool, L. (2006). Surf: Speeded up robust features. In *European conference on computer vision*. Springer, (pp. 404–417).

Bhowmik, A., Gumhold, S., Rother, C., & Brachmann, E. (2020). Reinforced feature points: Optimizing feature detection and description for a high-level task. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4948–4957).

Bian, J. W., Wu, Y. H., Zhao, J., Liu, Y., Zhang, L., Cheng, M. M., & Reid, I. (2019). An evaluation of feature matchers for fundamental matrix estimation. In *British machine vision conference (BMVC)*.

Choy, C., Park, J., & Koltun, V. (2019). Fully convolutional geometric features. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 8958–8966).

Christiansen, P. H., Kragh, M. F., Brodskiy, Y.,& Karstoft, H. (2019). Unsuperpoint: End-to-end unsupervised interest point detector and descriptor. arXiv preprint arXiv:1907.04011.

Chum, O., Werner, T., & Matas, J. (2005). Two-view geometry estimation unaffected by a dominant plane. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol 1 (pp. 772–779) vol. 1, 10.1109/CVPR.2005.354.

DeTone, D., Malisiewicz, T.,& Rabinovich, A. (2018). Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 224–236).

Dusmanu, M., Schonberger, J. L., Sinha, S. N., & Pollefeys, M. (2021). Privacy-preserving image features via adversarial affine subspace embeddings. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 14267–14277).

Dusmanu, M., Rocco, I., Pajdla, T., Pollefeys, M., Sivic, J., Torii, A., & Sattler, T. (2019). D2-net: A trainable CNN for joint description and detection of local features. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8092–8101).

Ebel, P., Mishchuk, A., Yi, K. M., Fua, P., & Trulls, E. (2019a). Beyond Cartesian representations for local descriptors. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*.

Ebel, P., Mishchuk, A., Yi, K. M., Fua, P., & Trulls, E. (2019b). Beyond cartesian representations for local descriptors. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 253–262).

Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM, 24*(6), 381–395.

Geiger, A., Lenz, P., & Urtasun, R. (2012). Are we ready for autonomous driving? The kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, IEEE (pp. 3354–3361).

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 580–587).

Gou, J., Yu, B., Maybank, S. J.,& Tao, D. (2020). Knowledge distillation: A survey. arXiv preprint arXiv:2006.05525.

Harris, C. G., Stephens, M., et al. (1988). A combined corner and edge detector. *Alvey Vision Conference Citeseer, 15*, 10–5244.

He, K., Lu, Y., & Sclaroff, S. (2018). Local descriptors optimized for average precision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 596–605).

Heinly, J., Schonberger, J. L., Dunn, E., & Frahm, J. M. (2015). Reconstructing the world* in six days*(as captured by the yahoo 100

million image dataset). In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3287–3295).
Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531.
Jaderberg, M., Simonyan, K., Zisserman, A.,& Kavukcuoglu, K. (2015). Spatial transformer networks. arXiv preprint arXiv:1506.02025.
Jin, Y., Mishkin, D., Mishchuk, A., Matas, J., Fua, P., Yi, K. M., & Trulls, E. (2021). Image matching across wide baselines: From paper to practice. *International Journal of Computer Vision, 129*(2), 517–547. https://doi.org/10.1007/s11263-020-01385-0
Jung, Y., Nizam, N. S. S. B. A., & Lee, S. C. (2023). Local feature extraction from salient regions by feature map transformation. arXiv preprint arXiv:2301.10413.
Keller, M., Chen, Z., Maffra, F., Schmuck, P., & Chli, M. (2018). Learning deep descriptors with scale-aware triplet networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2762–2770).
Knapitsch, A., Park, J., Zhou, Q. Y., & Koltun, V. (2017). Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG), 36*(4), 1–13.
Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems, 25*, 1097–1105.
Lee, J., Kim, B., & Cho, M. (2022). Self-supervised equivariant learning for oriented keypoint detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4847–4857).
Li, J., Li, G., & Li, T. H. (2022). Attention guided invariance selection for local feature descriptors. *ICASSP 2022–2022 IEEE international conference on acoustics* (pp. 2215–2219). IEEE: Speech and Signal Processing (ICASSP).
Li, K., Wang, L., Liu, L., Ran, Q., Xu, K., & Guo, Y. (2022b). Decoupling makes weakly supervised local feature better. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 15838–15848).
Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2117–2125).
Li, Y., Snavely, N., Huttenlocher, D., & Fua, P. (2012). Worldwide Pose Estimation Using 3D Point Clouds. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, & C. Schmid (Eds.), *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part I* (pp. 15–29). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-33718-5_2
Liu, X., Meng, C., Tian, F. P., & Feng, W. (2021). Dgd-net: Local descriptor guided keypoint detection network. In: 2021 *IEEE international conference on multimedia and expo (ICME)* (pp. 1–6). https://doi.org/10.1109/ICME51207.2021.9428406.
Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431–3440).
Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision, 60*(2), 91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94
Luo, Z., Shen, T., Zhou, L., Zhang, J., Yao, Y., Li, S., Fang, T., & Quan, L. (2019). Contextdesc: Local descriptor augmentation with cross-modality context. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 2527–2536).
Luo, Z., Zhou, L., Bai, X., Chen, H., Zhang, J., Yao, Y., Li, S., Fang, T., & Quan, L. (2020). Aslfeat: Learning local features of accurate shape and localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 6589–6598).
Luo, Z., Shen, T., Zhou, L., Zhu, S., Zhang, R., Yao, Y., Fang, T., & Quan, L. (2018). GeoDesc: Learning Local Descriptors by Integrating Geometry Constraints. In V. Ferrari, M. Hebert, C. Sminchisescu, & Y. Weiss (Eds.), *Computer Vision–ECCV 2018: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part IX* (pp. 170–185). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-01240-3_11
Mai, G., Cao, K., Yuen, P. C., & Jain, A. K. (2018). On the reconstruction of face images from deep face templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. *41*(5), 1188–1202.
Mikolajczyk, K., & Mikolajczyk, K. (2004). Scale & affine invariant interest point detectors. *International Journal of Computer Vision, 60*(1), 63–86. https://doi.org/10.1023/B:VISI.0000027790.02288.f2
Mikolajczyk, K., & Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 27*(10), 1615–1630.
Mishchuk, A., Mishkin, D., Radenovic, F., & Matas, J. (2017). Working hard to know your neighbor's margins: Local descriptor learning loss. arXiv preprint arXiv:1705.10872.
Mishkin, D., Radenović, F., & Matas, J. (2018). Repeatability Is Not Enough: Learning Affine Regions via Discriminability. In V. Ferrari, M. Hebert, C. Sminchisescu, & Y. Weiss (Eds.), *Computer Vision–ECCV 2018: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part IX* (pp. 287–304). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-01240-3_18
Moravec, H. P. (1977). Techniques towards automatic visual obstacle avoidance. In *Proceedings of the 5th international joint conference on artificial intelligence*. Cambridge, MA, USA, August (pp. 22–25).
Ng, T., Kim, H. J., Lee, V. T., DeTone, D., Yang, T. Y., Shen, T., Ilg, E., Balntas, V., Mikolajczyk, K., & Sweeney, C. (2022). Ninjadesc: Content-concealing visual descriptors via adversarial learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 12797–12807).
Noh, H., Araujo, A., Sim, J., Weyand, T., & Han, B. (2017). Large-scale image retrieval with attentive deep local features. In *Proceedings of the IEEE international conference on computer vision* (pp. 3456–3465).
Ono, Y., Trulls, E., Fua, P., & Yi, K. M. (2018). Lf-net: Learning local features from images. In *Proceedings of the 32nd international conference on neural information processing systems* (pp. 6237–6247).
Pautrat, R., Larsson, V., Oswald, M. R., & Pollefeys, M. (2020). Online Invariance Selection for Local Feature Descriptors. In A. Vedaldi, H. Bischof, T. Brox, & J.-M. Frahm (Eds.), *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II* (pp. 707–724). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-58536-5_42
Potje, G., Martins, R., Chamone, F., & Nascimento, E. (2021). Extracting deformation-aware local features by learning to deform. *Advances in Neural Information Processing Systems* p. 34.
Radenović. F., Tolias, G., & Chum, O. (2016). CNN image retrieval learns from bow: Unsupervised fine-tuning with hard examples. In *European conference on computer vision*. Springer (pp. 3–20).
Revaud, J., Leroy, V., Weinzaepfel, P., & Chidlovskii, B. (2022). Pump: Pyramidal and uniqueness matching priors for unsupervised learning of local descriptors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 3926–3936).
Revaud, J., Weinzaepfel, P., de Souza, C. R., & Humenberger, M. (2019). R2D2: Repeatable and reliable detector and descriptor. In *NeurIPS*.
Richardson, A., & Olson, E. (2013). Learning convolutional filters for interest point detection. In *2013 IEEE international conference on*

robotics and automation (pp. 631–637). https://doi.org/10.1109/ICRA.2013.6630639.
Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International conference on medical image computing and computer-assisted intervention*. Springer (pp. 234–241).
Santellani, E., Sormann, C., Rossi, M., Kuhn, A., & Fraundorfer, F. (2022). Md-net: Multi-detector for local feature extraction. In *2022 26th International conference on pattern recognition (ICPR)*. IEEE (pp. 3944–3951).
Sarlin, P. E., DeTone, D., Malisiewicz, T., & Rabinovich, A. (2020). Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4938–4947).
Sattler, T., Weyand, T., Leibe, B., & Kobbelt, L. (2012). Image retrieval for image-based localization revisited. In *BMVC*, vol. 1 (p. 4).
Savinov, N., Seki, A., Ladicky, L., Sattler, T., & Pollefeys, M. (2017). Quad-networks: Unsupervised learning to rank for interest point detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1822–1830).
Schönberger, J. L., Zheng, E., Frahm, J. M., & Pollefeys, M. (2016). Pixelwise view selection for unstructured multi-view stereo. In *European conference on computer Vision*. Springer (pp. 501–518).
Schonberger, J. L., & Frahm, J. M. (2016). Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4104–4113).
Schonberger, J. L., Hardmeier, H., Sattler, T., & Pollefeys, M. (2017). Comparative evaluation of hand-crafted and learned local features. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1482–1491).
Shen, T., Luo, Z., Zhou, L., Zhang, R., Zhu, S., Fang, T., & Quan, L. (2018). Matchable image retrieval by learning from surface reconstruction. In *The Asian conference on computer vision (ACCV)*.
Shen, X., Wang, C., Li, X., Yu, Z., Li, J., Wen, C., Cheng, M., & He, Z. (2019). Rf-net: An end-to-end image matching network based on receptive field. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8132–8140).
Shi, J., et al. (1994). Good features to track. In *1994 Proceedings of IEEE conference on computer vision and pattern recognition*. IEEE (pp. 593–600).
Shui, P. L., & Zhang, W. C. (2013). Corner detection and classification using anisotropic directional derivative representations. *IEEE Transactions on Image Processing, 22*(8), 3204–3218.
Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *International conference on learning representations*.
Simo-Serra, E., Trulls, E., Ferraz, L., Kokkinos, I., Fua, P., & Moreno-Noguer, F. (2015). Discriminative learning of deep convolutional feature point descriptors. In *Proceedings of the IEEE international conference on computer vision* (pp. 118–126).
Siqueira, H., Ruhkamp, P., Halfaoui, I., Karmann, M., & Urfalioglu, O. (2022). Looking beyond corners: Contrastive learning of visual representations for keypoint detection and description extraction. In *2022 international joint conference on neural networks (IJCNN)*. IEEE (pp. 1–8).
Sivic, Z. (2003). Video google: A text retrieval approach to object matching in videos. In *Proceedings ninth IEEE international conference on computer vision* (pp. 1470–1477) vol. 2 |DOIurl10.1109/ICCV.2003.1238663.
Sturm, J., Engelhard, N., Endres, F., Burgard, W., & Cremers, D. (2012). A benchmark for the evaluation of RGB-D slam systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE (pp. 573–580).
Sun, J., Shen, Z., Wang, Y., Bao, H., & Zhou, X. (2021). Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8922–8931).
Sun, S., Park, U., Sun, S., & Liu, R. (2022b). Fusion representation learning for keypoint detection and description. *The Visual Computer* pp 1–10.
Sun, J., Zhu, J., & Ji, L. (2022a). Shared coupling-bridge for weakly supervised local feature learning. arXiv preprint arXiv:2212.07047.
Suwanwimolkul, S., Komorita, S., & Tasaka, K. (2021). Learning of low-level feature keypoints for accurate and robust detection. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision* (pp. 2262–2271).
Svärm, L., Enqvist, O., Kahl, F., & Oskarsson, M. (2017). City-scale localization for cameras with known vertical direction. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 39*(7), 1455–1461. https://doi.org/10.1109/TPAMI.2016.2598331
Tian, Y., Balntas, V., Ng, T., Barroso-Laguna, A., Demiris, Y., & Mikolajczyk, K. (2020a). D2d: Keypoint extraction with describe to detect approach. In *Proceedings of the Asian conference on computer vision*.
Tian, Y., Barroso Laguna, A., Ng, T., Balntas, V., & Mikolajczyk, K. (2020b). Hynet: Learning local descriptor with hybrid similarity measure and triplet loss. *Advances in Neural Information Processing Systems* 33.
Tian, Y., Fan, B., & Wu, F. (2017). L2-net: Deep learning of discriminative patch descriptor in euclidean space. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 661–669).
Tian, Y., Yu, X., Fan, B., Wu, F., Heijnen, H., & Balntas, V. (2019). Sosnet: Second order similarity regularization for local descriptor learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 11016–11025).
Trajković, M., & Hedley, M. (1998). Fast corner detection. *Image and Vision Computing, 16*(2), 75–87.
Tyszkiewicz, M., Fua, P., & Trulls, E. (2020). Disk: Learning local features with policy gradient. *Advances in Neural Information Processing Systems* 33.
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems* 30.
Verdie, Y., Yi, K., Fua, P., & Lepetit, V. (2015). Tilde: A temporally invariant learned detector. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5279–5288).
Wang, Z., Li, X., & Li, Z. (2021). Local representation is not enough: Soft point-wise transformer for descriptor and detector of local features. In *Proceedings of the thirtieth international joint conference on artificial intelligence, IJCAI 2021, virtual event/Montreal, Canada, 19–27 August 2021* ijcai.org (pp. 1150–1156).
Wang, X., Liu, Z., Hu, Y., Xi, W., Yu, W., & Zou, D. (2022d). Featurebooster: Boosting feature descriptors with a lightweight neural network. arXiv preprint arXiv:2211.15069.
Wang, C., Zhang, G., Cheng, Z., & Zhou, W. (2022c). Rethinking low-level features for interest point detection and description. In *Proceedings of the Asian conference on computer vision* (pp. 2059–2074).
Wang, C., Xu, R., Xu, S., Meng, W., & Zhang, X. (2022a). Cndesc: Cross normalization for local descriptors learning. *IEEE Transactions on Multimedia*.
Wang, C., Xu, R., Zhang, Y., Xu, S., Meng, W., Fan, B., & Zhang, X. (2022). Mtldesc: Looking wider to describe better. *Proceedings of the AAAI Conference on Artificial Intelligence, 36*, 2388–2396.
Wang, Q., Zhou, X., Hariharan, B., & Snavely, N. (2020). Learning feature descriptors using camera pose supervision. In *European conference on computer vision*. Springer (pp. 757–774).
Weinzaepfel, P., & Jégou H, Pérez, P. (2011). Reconstructing an image from its local descriptors. In *CVPR 2011*. IEEE (pp. 337–344).

Wiles, O., Ehrhardt, S., & Zisserman, A. (2021). Co-attention for conditioned image matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 15920–15929).

Wilson, K., & Snavely, N. (2014). Robust global translations with 1dsfm. In *European conference on computer vision*. Springer (pp. 61–75).

Yang, T. Y., Nguyen, D. K., Heijnen, H., & Balntas, V. (2020). Ur2kid: Unifying retrieval, keypoint detection, and keypoint description without local correspondence supervision. arXiv preprint arXiv:2001.07252.

Yang, N., Han, Y., Fang, J., Zhong, W., & Xu, A. (2022). Up-net: Unique keypoint description and detection net. *Machine Vision and Applications, 33*(1), 1–13.

Yan, P., Tan, Y., & Tai, Y. (2022). Repeatable adaptive keypoint detection via self-supervised learning. *Science China Information Sciences, 65*(11), 1–25.

Yi, K. M., Trulls, E., Lepetit, V., & Fua, P. (2016a). Lift: Learned invariant feature transform. In *European conference on computer vision*. Springer (pp. 467–483).

Yi, K. M., Verdie, Y., Fua, P.,& Lepetit, V. (2016b). Learning to assign orientations to feature points. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 107–116).

Yi, K. M., Verdie, Y., Fua, P., & Lepetit, V. (2016c). Learning to assign orientations to feature points. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*.

Zhang, Y., Wang, J., Xu, S., Liu, X., & Zhang, X. (2020). Mlifeat: Multi-level information fusion based deep local features. In *Proceedings of the Asian conference on computer vision*.

Zhang, X., Yu, F. X., Karaman, S., & Chang, S. F. (2017). Learning discriminative and transformation covariant local feature detectors. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*.

Zhang, Z. (1998). Determining the epipolar geometry and its uncertainty: A review. *International Journal of Computer Vision, 27*(2), 161–195.

Zhang, Z., Sattler, T., & Scaramuzza, D. (2021). Reference pose generation for long-term visual localization via learned features and view synthesis. *International Journal of Computer Vision, 129*(4), 821–844.

Zhang, W., Sun, C., & Gao, Y. (2023). Image intensity variation information for interest point detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Zhao, X., Wu, X., Miao, J., Chen, W., Chen, P. C. Y., & Li, Z. (2022). Alike: Accurate and lightweight keypoint detection and descriptor extraction. *IEEE Transactions on Multimedia*. https://doi.org/10.1109/TMM.2022.3155927

Zhao, Z., Zhai, Y., Chen, B. M., & Liu, P. (2022b). Balf: Simple and efficient blur aware local feature detector. arXiv preprint arXiv:2211.14731.

Zhu, X., Hu, H., Lin, S., & Dai, J. (2019). Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9308–9316).