ASLFeat: Learning Local Features of Accurate Shape and Localization

Zixin Luo[1] Lei Zhou[1] Xuyang Bai[1] Hongkai Chen[1] Jiahui Zhang[2]
Yao Yao[1] Shiwei Li[3] Tian Fang[3] Long Quan[1]
[1]Hong Kong University of Science and Technology
[2]Tsinghua University [3]Everest Innovation Technology
{zluoag,lzhouai,xbaiad,hchencf,yaoag,quan}@cse.ust.hk
jiahui-z15@mails.tsinghua.edu.cn {sli,fangtian}@altizure.com

Abstract

This work focuses on mitigating two limitations in the joint learning of local feature detectors and descriptors. First, the ability to estimate the local shape (scale, orientation, etc.) of feature points is often neglected during dense feature extraction, while the shape-awareness is crucial to acquire stronger geometric invariance. Second, the localization accuracy of detected keypoints is not sufficient to reliably recover camera geometry, which has become the bottleneck in tasks such as 3D reconstruction. In this paper, we present ASLFeat, with three light-weight yet effective modifications to mitigate above issues. First, we resort to deformable convolutional networks to densely estimate and apply local transformation. Second, we take advantage of the inherent feature hierarchy to restore spatial resolution and low-level details for accurate keypoint localization. Finally, we use a peakiness measurement to relate feature responses and derive more indicative detection scores. The effect of each modification is thoroughly studied, and the evaluation is extensively conducted across a variety of practical scenarios. State-of-the-art results are reported that demonstrate the superiority of our methods. [code release]

1. Introduction

Designing powerful local features is an essential basis for a broad range of computer vision tasks [31, 43, 44, 30, 40, 15, 40]. During the past few years, the joint learning of local feature detectors and descriptors has gained increasing popularity, with promising results achieved in real applications. However, there are two limitations we consider that may have hinged further boost in performance: 1) the lack of shape-awareness of feature points for acquiring stronger geometric invariance, and 2) the lack of keypoint localization accuracy for solving camera geometry robustly.

Traditionally, the local shape is parameterized by hand-crafted scale/rotation estimation [17, 29] or affine shape adaptation [20], while more recently, data-driven approaches [23, 22, 39] have emerged that build a separate network to regress the shape parameters, then transform the *patch inputs* before feature descriptions. Due to the increasing prevalence of the joint learning with keypoint detectors [6, 25, 27, 7, 4], recent research focus has shifted to frameworks that densely extract features from *image inputs*, whereas no pre-defined keypoint is given and thus previous patch-wise shape estimation becomes inapplicable. As an alternative, LF-Net [25] extracts dense features and transforms intermediate feature maps via Spatial Transformer Networks (STN) [12], whereas multiple forward passes are needed and only sparse predictions of shape parameters are practically feasible. In this view, there still lacks a solution that enables efficient local shape estimation in a dense prediction framework.

Besides, the localization accuracy of learned keypoints is still concerned in solving geometry-sensitive problems. For instance, LF-Net [25] and D2-Net [7] empirically yield low precision in two-view matching or introduce large reprojection error in Structure-from-Motion (SfM) tasks, which in essence can be ascribed to the lack of spatial accuracy as the detections are derived from low-resolution feature maps (e.g., $1/4$ times the original size). To restore the spatial resolution, SuperPoint [6] learns to upsample the feature maps with pixel-wise supervision from artificial points, while R2D2 [27] employs dilated convolutions to maintain the spatial resolution but trades off excessive GPU computation and memory usage. Moreover, it is questionable that if the detections from the deepest layer are capable of identifying low-level structures (corners, edges, etc.) where keypoints are often located. Although widely discussed in dense prediction tasks [28, 10, 16], in our context, neither the keypoint localization accuracy, nor the low-level nature of keypoint detection has received adequate attention.

To mitigate above limitations, we present ASLFeat, with three light-weight yet effective modifications. First, we em-

ploy deformable convolutional networks (DCN) [5, 45] in the dense prediction framework, which allows for not only pixel-wise estimation of local transformation, but also progressive shape modelling by stacking multiple DCNs. Second, we leverage the inherent feature hierarchy, and propose a multi-level detection mechanism that restores not only the spatial resolution without extra learning weights, but also low-level details for accurate keypoint localization. Finally, we base our methods on an improved D2-Net [7] that is trained from scratch, and further propose a peakiness measurement for more selective keypoint detection.

Despite the key insights of above modifications being familiar, we address their importance in our specific context, fully optimize the implementation in a non-trivial way, and thoroughly study the effect by comparing with different design choices. To summarize, we aim to provide answers to two critical questions: 1) what *deformation parameterization* is needed for local descriptors (geometrically constrained [23, 22, 39] or free-form modelling [5, 45]), 2) what *feature fusion* is effective for keypoint detectors (multi-scale input [27, 7], in-network multi-scale inference [25], or multi-level fusion [28]). Finally, we extensively evaluate our methods across various practical scenarios, including image matching [1, 2], 3D reconstruction [32] and visual localization [30]. We demonstrate drastic improvements upon the backbone architecture, D2-Net, and report state-of-the-art results on popular benchmarks.

2. Related works

Hand-crafted local features have been widely evaluated in [1, 32], we here focus mainly on the learning approaches.

Local shape estimation. Most existing descriptor learning methods [19, 18, 21, 37, 36, 40] do not explicitly model the local shape, but rely on geometric data augmentation (scaling/rotational perturbation) or hand-crafted shape estimation (scale/rotation estimation [17, 29]) to acquire geometric invariance. Instead, OriNet [23] and LIFT [39] propose to learn a canonical orientation of feature points, AffNet [22] predicts more affine parameters to improve the modelling power, and the log-polar representation [8] is used to handle in particular scale changes. Despite the promising results, those methods are limited to take *image patches* as input, and introduce a considerable amount of computation since two independent networks are constructed for predicting patch shape and patch description separately. As an alternative, LF-Net [25] takes *images* as input and performs STN [12] on intermediate features, while multiple forward passes are needed to transform individual "feature patch", and thus only prediction on sparse locations is practically applicable.

Meanwhile, the modelling of local shape has been shown crucial in image recognition tasks, which inspires works such as scale-adaptive convolution (SAC) for flexible-size dilations [42] and deformable convolution networks (DCN) for tunable grid sampling locations [5, 45]. In this paper, we adopt the similar idea in our context, and propose to equip DCN for dense local transformation prediction, of which the inference requires only a single forward pass and is thus of high efficiency.

Joint local feature learning. The joint learning of feature detectors and descriptors has received increasing attention, where a unified network is constructed to share most computations of the two tasks for fast inference. In terms of descriptor learning, the ranking loss [25, 7, 6, 4, 27] has been primarily used as a *de-facto* standard. However, due to the difficulty of acquiring unbiased ground-truth data, no general consensus has yet been reached regarding an effective loss design for keypoint detector learning. For instance, LF-Net [25] warps the detection map and minimizes the difference at selected pixels in two views, while SuperPoint [6] operates a self-supervised paradigm with a bootstrap training on synthetic data and multi-round adaptations on real data. More recent R2D2 [27] enforces grid-wise peakiness in conjunction with reliability prediction for descriptor, while UnsuperPoint [4] and Key.Net [14] learn grid-wise offsets to localize keypoints.

By contrast, D2-Net [7] eschews learning extra weights for a keypoint detector, but hand-crafts a selection rule to derive keypoints from the same feature maps that are used for extracting feature descriptors. This design essentially couples the capability of the feature detector and descriptor, and results in a clean framework without complex heuristics in loss formulation. However, it is a known issue that D2-Net lacks of accuracy of keypoint localization, as keypoints are derived from low-resolution feature maps. In this paper, we base our methods on D2-Net, and mitigate above limitation by a light-weight modification that cheaply restores both the spatial resolution and low-level details.

3. Methods

3.1. Prerequisites

The backbone architecture in this work is built upon 1) deformable convolutional networks (DCN) [5, 45] that predict and apply dense spatial transformation, and 2) D2-Net [7] that jointly learns keypoint detector and descriptor.

Deformable convolutional networks (DCN) [5, 45] target to learn dynamic receptive filed to accommodate the ability of modelling geometric variations. Formally, given a regular grid $\mathcal{R}$ that samples values over the input feature maps $\mathbf{x}$, the output features $\mathbf{y}$ of a standard convolution for each spatial position $\mathbf{p}$ can be written as:

$$\mathbf{y}(\mathbf{p}) = \sum_{\mathbf{p}_n \in \mathcal{R}} \mathbf{w}(\mathbf{p}_n) \cdot \mathbf{x}(\mathbf{p} + \mathbf{p}_n). \tag{1}$$

DCN augments the regular convolution by additionally

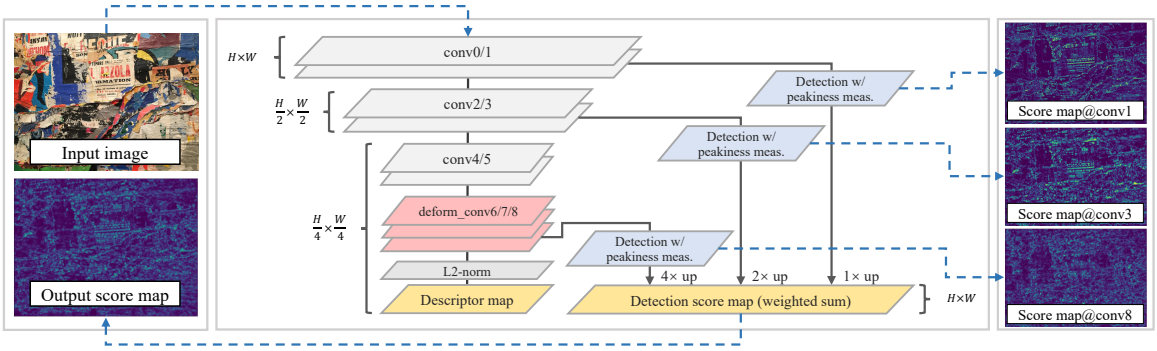

Figure 1. Network architecture, with the proposed equipment of deformable convlutional network (DCN), multi-level detection (MulDet), and peakiness measurement for keypoint scoring.

learning both sampling offsets [5] $\{\triangle \mathbf{p}_n | n = 1, ..., N\}$ and feature amplitudes [45] $\{\triangle \mathbf{m}_n | n = 1, ..., N\}$, where $N = |\mathcal{R}|$, and rewrites Eq. 1 as:

$$\mathbf{y}(\mathbf{p}) = \sum_{\mathbf{p}_n \in \mathcal{R}} \mathbf{w}(\mathbf{p}_n) \cdot \mathbf{x}(\mathbf{p} + \mathbf{p}_n + \triangle \mathbf{p}_n) \cdot \triangle \mathbf{m}_n. \quad (2)$$

As the offset $\triangle \mathbf{p_n}$ is typically fractional, Eq. 2 is implemented via bilinear interpolation, while the feature amplitude $\triangle \mathbf{m}_n$ is limited to $(0, 1)$. During training, the initial values of $\triangle \mathbf{p}_n$ and $\triangle \mathbf{m}_n$ are respectively set to 0 and 0.5, following the settings in [45].

D2-Net [7] proposes a *describe-and-detect* strategy to jointly extract feature descriptions and detections. Over the last feature maps $\mathbf{y} \in \mathbb{R}^{H \times W \times C}$, D2-Net applies channel-wise L2-normalization to obtain dense feature descriptors, while the feature detections are derived from 1) the local score and 2) the channel-wise score. Specifically, for each location (i, j) in $\mathbf{y}^c$ $(c = 1, 2, ..., C)$, the local score is obtained by:

$$\alpha_{ij}^c = \frac{\exp\left(\mathbf{y}_{ij}^c\right)}{\sum_{(i',j') \in \mathcal{N}(i,j)} \exp \mathbf{y}_{i'j'}^c}, \quad (3)$$

where $\mathcal{N}(i, j)$ is neighboring pixels around (i, j), e.g., 9 neighbours defined by a 3×3 kernel. Next, the channel-wise score is obtained by:

$$\beta_{ij}^c = \mathbf{y}_{ij}^c / \max_t \mathbf{y}_{ij}^t. \quad (4)$$

The final detection score is combined as:

$$s_{ij} = \max_t (\alpha_{ij}^c \beta_{ij}^c). \quad (5)$$

The detection score will be later used as a weighting term in loss formulation (Sec. 3.4), and will allow for top-K selection of keypoints during testing.

3.2. DCN with Geometric Constraints

The original free-form DCN predicts local transformation of high degrees of freedom (DOF), e.g., 9×2 offsets for a 3×3 kernel. On the one hand, it enables the potential to model complex deformation such as non-planarity, while on the other hand, it takes a risk of over-paramertizing the local shape, where simpler affine or perspective transformation are often considered to serve as a good approximation [20, 23, 22]. To find out what deformation is needed in our context, we compare three shape modellings via enforcing different geometric constraints in DCN, including 1) similarity, 2) affine and 3) homography. The shape properties of the investigated variants are summarized in Tab. 1.

Variants	Modeling Power	DOF
unconstrained	non-planarity	$2k^2$
s.t. similarity	scale, rotation	2
s.t. affine	scale, rotation, shear	4
s.t. homography	perspective	6

Table 1. The shape properties of DCN variants, where DOF denotes the degrees of freedom and k denotes the kernel size of convolution. Translation is omitted as is fixed for keypoints.

Affine-constrained DCN. Traditionally, the local shape is often modelled by similarity transformation with estimates of rotation and scale [17, 29]. In a learning framework such as [23, 25], this transformation is decomposed as:

$$\mathbf{S} = \lambda R(\theta) = \lambda \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix}. \quad (6)$$

Moreover, a few works such as HesAff [20] further includes an estimate of shearing, which is cast as a learnable problem by AffNet [22]. Here, we follow AffNet and decompose the affine transformation as:

$$\begin{aligned} \mathbf{A} &= \mathbf{S}A' = \lambda R(\theta) A' \\ &= \lambda \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} a'_{11} & 0 \\ a'_{21} & a'_{22} \end{pmatrix}, \end{aligned} \quad (7)$$

where $\det A' = 1$. The network is implemented to predict one scalar for scaling (λ), another two for rotation ($\cos(\theta)$, $\sin(\theta)$), while the other three for shearing (A').

Homography-constrained DCN. Virtually, the local deformation can be better approximated by a homography (perspective) transformation $\mathbf{H}$, and we here adopt the Tensor Direct Linear Transform (Tensor DLT) [24] to solve the 4-point parameterization of $\mathbf{H}$ in a differentiable manner.

Formally, a linear system can be created that solves $\mathbf{Mh} = \mathbf{0}$, where $\mathbf{M} \in \mathbb{R}^{8\times 9}$ and $\mathbf{h}$ is a vector with 9 elements consisting of the entries of $\mathbf{H}$, and each correspondence provides two equations in $\mathbf{M}$. By enforcing the last element of $\mathbf{h}$ equals to 1 [11] and omitting the translation, we set $\mathbf{H}_{33} = 1$ and $\mathbf{H}_{13} = \mathbf{H}_{23} = 0$, then rewrite the above system of equations as $\hat{\mathbf{M}}_{(i)}\hat{\mathbf{h}} = \hat{\mathbf{b}}_{(i)}$, where $\hat{\mathbf{M}}_{(i)} \in \mathbb{R}^{2\times 6}$ and for each correspondence,

$$\hat{\mathbf{M}}_{(i)} = \begin{bmatrix} 0 & 0 & -u_i & -v_i & v'_i u_i & v'_i v_i \\ u_i & v_i & 0 & 0 & -u'_i u_i & -u'_i v_i \end{bmatrix}, \quad (8)$$

$\hat{\mathbf{b}}_{(i)} = [-v'_i, u'_i]^T \in \mathbb{R}^{2\times 1}$ and $\hat{\mathbf{h}}$ consists of 6 elements from the first two columns of $\mathbf{H}$. By stacking the equations of 4 correspondences, we derive the final linear system:

$$\hat{\mathbf{M}}\hat{\mathbf{h}} = \hat{\mathbf{b}}. \quad (9)$$

Suppose that correspondence points are not collinear, $\hat{\mathbf{h}}$ can be then efficiently and uniquely solved by using the differentiable pseudo-inverse of $\hat{\mathbf{A}}$[1]. In practice, we initialize 4 corner points at $\{(-1,-1),(1,-1),(1,1),(-1,1)\}$, and implement the network to predict 8 corresponding offsets lying in $(-1, 1)$ so as to avoid collinearity.

After forming the above transformation $\mathbf{T} \in \{\mathbf{S}, \mathbf{A}, \mathbf{H}\}$, the offset values in Eq. 2 are now obtained by:

$$\triangle\, \mathbf{p}_n = \mathbf{T}\mathbf{p}_n - \mathbf{p}_n, \text{where } \mathbf{p}_n \in \mathcal{R}, \quad (10)$$

so that geometry constraints are enforced in DCN. More implementation details can be found in the Appendix.

3.3. Selective and Accurate Keypoint Detection

Keypoint peakiness measurement. As introduced in Sec. 3.1, D2-Net scores a keypoint regarding both spatial and channel-wise responses. Among many possible metrics, D2-Net implements a *ratio-to-max* (Eq. 4) to evaluate channel-wise extremeness, whereas one possible limitation lies on that it only weakly relates to the actual distribution of all responses along the channel.

To study this effect, we first trivially modify Eq. 4 with a channel-wise `softmax`, whereas this modification deteriorates the performance in our experiments. Instead, inspired by [27, 41], we propose to use *peakiness* as a keypoint measurement in D2-Net, which rewrites Eq. 4 as:

$$\beta^c_{ij} = \text{softplus}(\mathbf{y}^c_{ij} - \frac{1}{C}\sum_t \mathbf{y}^t_{ij}), \quad (11)$$

[1]Implemented via function `tf.matrix_solve` in TensorFlow.

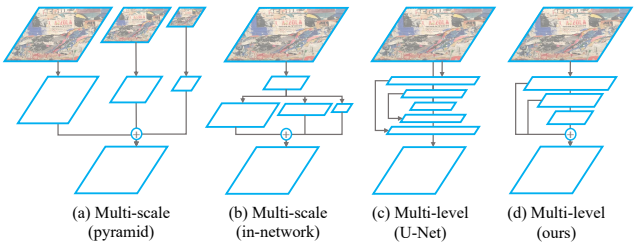

Figure 2. Different design choices to leverage feature hierarchy, shorten as variants of MulDet.

where `softplus` activates the peakiness to a positive value. To balance the scales of both scores, we also rewrites Eq. 3 in the similar form:

$$\alpha^c_{ij} = \text{softplus}(\mathbf{y}^c_{ij} - \frac{1}{|\mathcal{N}(i,j)|}\sum_{(i',j')\in\mathcal{N}(i,j)} \mathbf{y}^c_{i'j'}), \quad (12)$$

and the two scores are again combined as in Eq. 5.

Multi-level keypoint detection (MulDet). As aforementioned, one known limitation of D2-Net [7] is the lack of keypoint localization accuracy, since detections are obtained from low-resolution feature maps. There are multiple options to restore the spatial resolution, for instance, by learning an additional feature decoder (SuperPoint [6]) or employing dilated convolutions (R2D2 [27]). However, those methods either increase the number of learning parameters, or consume huge GPU memory or computation. Instead, we propose a simple yet effective solution without introducing extra learning weights, by leveraging the inherent pyramidal feature hierarchy of ConvNets and combining detections from multiple feature levels.

Specifically, given a feature hierarchy consisting of feature maps at different levels $\{\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, ..., \mathbf{y}^{(l)}\}$ strided by $\{1, 2, ..., 2^{(l-1)}\}$, respectively, we apply the aforementioned detection at each level to get a set of score maps $\{\mathbf{s}^{(1)}, \mathbf{s}^{(2)}, ..., \mathbf{s}^{(l)}\}$. Next, each score map is upsampled to have the same spatial resolution as input image, and finally combined by taking the weighted sum:

$$\hat{\mathbf{s}} = \frac{1}{\sum_l w_l}\sum_l w_l \mathbf{s}^{(l)}. \quad (13)$$

To better address the superiority of the proposed method, we implement 1) the multi-scale detection used in D2-Net [7] and R2D2 [27] (Fig. 4a) by constructing an image pyramid with multiple forward passes, 2) the in-network multi-scale prediction used in LF-Net [25] (Fig. 4b) by resizing the intermediate feature maps, and 3) the standard U-Net architecture [28] (Fig. 4c) that builds a feature decoder and skip connections from low-level feature maps.

The proposed multi-level detection (Fig. 4d) is advantageous in three aspects. Firstly, it adopts implicit multi-scale detection that conforms to classical scale-space theory [17]

by having different sizes of receptive field for localizing keypoints. Secondly, compared with U-Net architecture, it cheaply restores the spatial resolution without introducing extra learning weights to achieve pixel-wise accuracy. Thirdly, different from U-Net that directly fuses low-level and high-level features, it keeps the low-level features untouched, but fuses the *detections* of multi-level semantics, which helps to better preserve the low-level structures such as corners or edges. The implementation details of above variants can be found in the Appendix.

3.4. Learning Framework

Network architecture. The network architecture is illustrated in Fig. 1. To reduce computations, we replace the VGG backbone [34] used in D2-Net with more light-weight L2-Net [36]. Similar to R2D2 [27], we further replace the last 8×8 convolution of L2-Net by three 3×3 convolutions, resulting in feature maps of 128 dimension and $1/4$ times resolution of the input. Finally, the last three convolutions, `conv6`, `conv7` and `conv8`, are substituted with DCN (Sec. 3.1). Three levels, `conv1`, `conv3` and `conv8`, are selected to perform the proposed MulDet (Sec. 3.3). The combination weights in Eq. 13 are empirically set to $w_i = 1, 2, 3$, and the dilation rate to find neighboring pixels $\mathcal{N}(i, j)$ in Eq. 3 is set to $3, 2, 1$, respectively, which we find to deliver best trade-offs to balance the attention on low-level and abstracted features.

Loss design. We identify a set of correspondences $\mathcal{C}$ for an image pair (I, I') via densely warping I to I' regarding ground-truth depths and camera parameters. To derive the training loss for both detector and descriptor, we adopt the formulation in D2-Net [7], written as:

$$\mathcal{L}(I, I') = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \frac{\hat{s}_c \hat{s}'_c}{\sum_{q \in \mathcal{C}} \hat{s}_q \hat{s}'_q} \mathcal{M}(\mathbf{f}_c, \mathbf{f'}_c), \tag{14}$$

where $\hat{s}_k$ and $\hat{s}'_k$ are combined detection scores in Eq. 13 for image I and I', $\mathbf{f}_k$ and $\mathbf{f}'_k$ are their corresponding descriptors, and $\mathcal{M}(\cdot, \cdot)$ is the ranking loss for representation learning. Instead of using the hardest-triplet loss in D2-Net [7], we adopt the hardest-contrastive form in FCGF [3], which we find guarantee better convergence when training from scratch and equipping DCN, written as:

$$\begin{aligned} \mathcal{M}(\mathbf{f}_c, \mathbf{f}'_c) = &[D(\mathbf{f}_c, \mathbf{f}'_c) - m_p]_+ + \\ &[m_n - \min(\min_{k \neq c} D(\mathbf{f}_c, \mathbf{f}'_k), \min_{k \neq c} D(\mathbf{f}_k, \mathbf{f}'_c))]_+, \end{aligned} \tag{15}$$

where $D(\cdot, \cdot)$ denotes the Euclidean distance measured between two descriptors, and m_p, m_n are respectively set to $0.2, 1.0$ for positives and negatives. Similar to D2-Net [7], a safe radius sized 3 is set to avoid taking spatially too close feature points as negatives.

3.5. Implementations

Training. In contrast to D2-Net [7] which starts from an ImageNet pretrained model with only the last convolution fine-tuned, we train our model *from scratch* with ground-truth cameras and depths obtained from [33, 26] (the same data used in [19, 18]). The training consumes $800K$ image pairs sized 480×480 and batched 2. Learning gradients are computed for image pairs that have at least 32 matches, while the maximum match number is limited to 512. Each input image is standardized to have zero mean and unit norm, and independently applied with random photometric augmentation including brightness, contrast and blurriness. The SGD optimizer is used with momentum of 0.9, and the base learning rate is set to 0.1.

Although an end-to-end learning with DCN is feasible, we find that a two-stage training yields better results in practice. Specifically, in the first round we train the model with *all regular convolutions* for $400K$ iterations. In the second round, we tune *only the DCNs* with the base learning rate divided by 10 for another $400K$ iterations. Our implementation is made in TensorFlow with single NVIDIA RTX 2080Ti card, and the training finishes within 42 hours.

Testing. A non-maximum suppression (NMS) sized 3 is applied to remove detections that are spatially too close. Similar to D2-Net, we postprocess the keypoints with the SIFT-like edge elimination (with threshold set to 10) and sub-pixel refinement, the descriptors are then bilinearly interpolated at the refined locations. We select top-K keypoints regarding detection scores obtained in Eq. 13, and empirically discard those whose scores are lower than 0.50.

4. Experiments

In the following sections, we evaluate our methods across several practical scenarios, including image matching, 3D reconstruction and visual localization. Further experiments on dense reconstruction and image retrieval can be found in the Appendix.

4.1. Image Matching

Datasets. First, we use the popular HPatches dataset [1], which includes 116 image sequences with ground-truth homography. Following D2-Net [7], we exclude 8 high-resolution sequences, leaving 52 and 56 sequences with illumination or viewpoint variations, respectively.

Though widely used, HPatches dataset exhibits only homography transformation, which may not comprehensively reflect the performance in real applications. Thus, we resort to the newly proposed FM-Bench [2], which comprises four datasets captured in practical scenarios: the TUM dataset [35] in indoor SLAM settings, the KITTI dataset [9] in driving scenes, the Tanks and Temples dataset

(T&T) [13] for wide-baseline reconstruction, and the Community Photo Collection (CPC) [38] for wild reconstruction from web images. For each datasets, 1000 overlapping image pairs are randomly chosen for evaluation, with ground-truth fundamental matrix pre-computed.

Evaluation protocols. On HPatches dataset [1], three standard metrics are used: 1) Keypoint repeatability (*%Rep.*), a.k.a. the ratio of possible matches and the minimum number of keypoints in the shared view. 2) Matching score (*%M.S.*), a.k.a. the ratio of correct matches and the minimum number of keypoints in the shared view. 3) Mean matching accuracy (*%MMA*), a.k.a. the ratio of correct matches and possible matches. Here, a match is defined to correspond if the point distance is below some error threshold after homography wrapping, and a correct match is further required to be a mutual nearest neighbor during brute-force searching. For above metrics, we report their average scores over all image pairs in the dataset.

In terms of FM-Bench [2], a full matching pipeline including outlier rejection (e.g., ratio test [17]) and geometric verification (e.g., RANSAC) is performed, and the final pose recovery accuracy is evaluated. To determine the correctness of a pose estimate, FM-Bench uses ground-truth pose to generate a set of virtual correspondences, then measures the average of normalized symmetric epipolar distance regarding a pose estimate, and finally computes *%Recall* as the ratio of estimates where the distance error is below a certain threshold (0.05 by default). At correspondence level, FM-Bench also reports intermediate results such as the inlier ratio (*%Inlier/%Inlier-m*) and correspondence number (*%Corr/%Corr-m*) after/before RANSAC.

HPatches dataset (error threshold @ 3px)				
	Config.	*%Rep.*	*%M.S.*	*%MMA*
D2-Net	*orig.*	**47.86**	23.58	43.00
	our impl.	43.34	29.55	45.36
	peakiness meas.	46.24	**32.27**	**48.54**
+ MulDet	*multi-scale (pyramid)*	46.12	32.55	48.72
	multi-scale (in-network)	45.17	31.74	47.94
	multi-level (U-Net)	75.35	40.12	66.42
	multi-level (ours)	**77.37**	**42.99**	**68.66**
+ MulDet & DCN	*s.t. similarity*	78.33	44.79	71.67
	s.t. affine	78.49	45.35	71.80
	s.t. homography	78.39	45.08	71.89
	free-form, 1 layer	78.27	45.12	71.08
	free-form	78.31	**46.28**	72.26
	free-form, multi-scale	**86.03**	39.37	**72.64**

Table 2. Ablation experiments of the proposed modifications, where **peakiness meas.** improves the detection scoring upon D2-Net, **+ MulDet** studies the effect of different feature fusion strategies, and **+ MulDet & DCN** further compares the effect of different parameterzation of deformation.

Comparative methods. We compare our methods with 1) patch descriptors, including HardNet++ [21] with SIFT [17] detector (*SIFT + HN++*), or plus a shape estimator HesAffNet [22] (*HAN + HN++*). Also, ContextDesc [18] with SIFT detector (*SIFT + ContextDesc*). 2) Joint local feature learning approaches including SuperPoint [6], LF-Net [25], D2-Net (fine-tuned) [7] and more recent R2D2 [27]. Unless otherwise specified, we report either results reported in original papers, or derived from authors' public implementations with default parameters. We limit the maximum numbers of features of our methods to 5K and 20K on HPatches dataset and FM-Bench, respectively.

On FM-Bench, both the mutual check and ratio test [17] are applied to reject outliers before RANSAC. A ratio at 0.8 is used for all methods except for D2-Net and R2D2[2].

Baseline. To avoid overstatement, we first present our re-implementation of D2-Net (*our impl.*) as the baseline. As mentioned in Sec. 3.4 and Sec. 3.5, the new baseline differs from the original D2-Net (*orig.*) in three aspects: 1) Different backbone architecture (L2-Net [36] with 128-d output vs. VGG [34] with 512-d output). 2) Different loss formulation (hardest-contrastive [3] vs. hardest-triplet [7]). 3) Different training settings (trained from scratch vs. fine-tuned only the last convolution from a pre-trained model). As shown in Tab. 2 and Tab. 3, the new baseline outperforms original D2-Net in general, while being more parameter- and computation-efficient regarding model size.

Ablations on peakiness measurement. We first adopt the peakiness measurement for more indicative keypoint scoring (Sec. 3.3). As shown in Tab. 2, this modification (*peakiness meas.*) notably improves the results regarding all evaluation metrics on HPatches dataset. This effect is validated on FM-Bench, which is shown to apply for all different scenarios as shown in Tab. 3 (*ASLFeat w/o peakiness meas.*). Our later modifications will be thus based on this model.

Ablations on MulDet. As shown in Tab. 2, applying multi-scale detection solely does not take obvious effect, as spatial accuracy is still lacking. Instead, adopting multi-level detection, with spatial resolution restored, remarkably boosts the performance, which conforms the necessity of pixel-level accuracy especially when small pixel error is tolerated. It is also note-worthy that, despite less learning weights and computation, the proposed multi-level detection outperforms the U-Net variant, addressing the low-level nature of this task where a better preservation of low-level features is beneficial. Although the proposed multi-level detection also includes feature fusion of difference scales, we find that combining a more explicit multi-scale (pyramid) detection (*free-form, multi-scale*) is in particular advantagous in order to handle the scale changes. This combination will be denoted as ASLFeat (MS) in the following context.

Ablations on DCN. As shown in Tab. 2, all investigated

[2]We use 0.95 for D2-Net as suggested in its original paper, and conduct a parameter searching for LF-Net, SuperPoint and R2D2, obtaining the ratio at 0.8, 0.8, and 0.9, respectively, to achieve overall best performance.

	TUM [35] (indoor SLAM settings)				**KITTI** [9] (driving settings)			
Methods	*%Recall*	*%Inlier*	*#Inlier-m*	*#Corrs (-m)*	*%Recall*	*%Inlier*	*#Inlier-m*	*#Corrs (-m)*
SIFT [17]	57.40	75.33	59.21	65 (316)	91.70	98.20	87.40	154 (525)
SIFT + HN++ [21]	58.90	75.74	62.07	67 (315)	92.00	98.21	91.25	159 (535)
HAN + HN++ [22]	51.70	75.70	62.06	101 (657)	90.40	98.09	90.64	233 (1182)
SIFT + ContextDesc [18]	59.70	75.53	62.61	69 (325)	**92.20**	98.23	91.92	160 (541)
LF-Net (MS) [25]	53.00	70.97	56.25	143 (851)	80.40	95.38	84.66	202 (1045)
D2-Net (MS) [7]	34.50	67.61	49.01	74 (1279)	71.40	94.26	73.25	103 (1832)
SuperPoint [6]	45.80	72.79	64.06	39 (200)	86.10	98.11	91.52	73 (392)
R2D2 (MS) [27]	57.70	73.70	61.53	260 (1912)	78.80	97.53	86.49	278 (1804)
D2-Net (our impl.)	39.10	70.09	61.58	64 (337)	70.80	97.04	91.97	81 (683)
ASLFeat (w/o peakiness meas.)	53.30	74.96	68.29	116 (703)	89.60	98.47	95.36	223 (1376)
ASLFeat	**60.20**	76.34	69.09	148 (739)	**92.20**	98.69	**96.25**	444 (1457)
ASLFeat (MS)	59.90	**76.72**	**69.50**	258 (1332)	**92.20**	**98.76**	96.16	630 (2222)
	T&T [13] (wide-baseline reconstruction)				**CPC** [38] (wild reconstruction from web images)			
SIFT	70.00	75.20	53.25	85 (795)	29.20	67.14	48.07	60 (415)
SIFT + HN++	79.90	81.05	63.61	96 (814)	40.30	76.73	62.30	69 (400)
HAN + HN++	82.50	84.71	70.29	97 (920)	47.40	82.58	72.22	65 (405)
SIFT + ContextDesc	81.60	83.32	69.92	94 (728)	41.80	84.01	72.21	61 (306)
LF-Net (MS)	57.40	66.62	60.57	54 (362)	19.40	44.27	44.35	50 (114)
D2-Net (MS)	68.40	71.79	55.51	78 (2603)	31.30	56.57	49.85	84 (1435)
SuperPoint	81.80	83.87	70.89	52 (535)	40.50	75.28	64.68	31 (225)
R2D2 (MS)	73.00	80.81	65.31	84 (1462)	43.00	82.40	67.28	91 (954)
D2-Net (our impl.)	83.20	84.19	75.32	74 (1009)	46.60	83.72	77.31	51 (464)
ASLFeat (w/o peakiness meas.)	86.30	84.71	77.84	171 (1775)	49.50	85.80	80.39	97 (780)
ASLFeat	**89.90**	85.33	79.08	295 (2066)	51.50	87.98	82.24	165 (989)
ASLFeat (MS)	88.70	**85.68**	**79.74**	327 (2465)	**54.40**	**89.33**	**82.76**	185 (1159)

Table 3. Evaluation results on FM-Bench [2] for pair-wise image matching, where *#Recall* denotes the percentage of accurate pose estimates, *#Inlier* and *#Inlier-m*, *#Corrs* and *#Corrs-m* denote the inlier ratio and correspondence number after/before RANSAC.

variants of DCN are valid and notably boost the performance. Among those designs, the free-form variant slightly outperforms the constrained version, despite the fact that HPatches datasets exhibit only homography transformation. This confirms that modelling non-planarity is feasible and useful for local features, and we thus opt for the free-form DCN to better handle geometric variations. Besides, we also implement a single-layer DCN (*free-form, 1 layer*) that replaces only the last regular convolution (i.e., `conv8` in Fig. 1), showing that stacking more DCNs is beneficial and the shape estimation can be learned progressively.

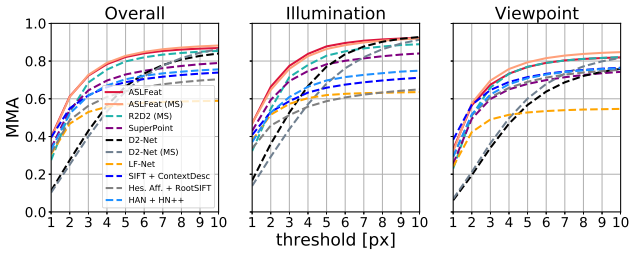

Figure 3. Comparisons on HPatches dataset [1] with mean matching accuracy (MMA) evaluated at different error thresholds, where "MS" denotes that the multi-scale inference is enabled.

Comparisons with other methods. As illustrated in Fig. 3, both ASLFeat and its multi-scale (MS) variant achieve overall best results on HPatches dataset regarding both illumination and viewpoint variations at different error thresholds. Specifically, ASLFeat delivers remarkable improvements upon its backbone architecture, D2-Net, especially at low error thresholds, which in particular demonstrates that the keypoint localization error has been largely reduced. Besides, ASLFeat notably outperforms the more recent R2D2 (72.64 vs. 68.64 for MMA@3 overall), while being more computationally efficient by eschewing the use of dilated convolutions for restoring spatial resolution.

In addition, as shown in Tab. 3 on FM-Bench, the ASLFeat remarkably outperforms other joint learning approaches. In particular, ASLFeat largely improves the state-of-the-art results on two MVS datasets: T&T and CPC, of which the scenarios are consistent with the training data. It is also noteworthy that our methods generalize well to unseen scenarios: TUM (indoor scenes) and KITTI (driving scenes). As a common practice, adding more task-specific training data is expected to further boost the performance.

Visualizations. We here present some sample detection results on FM-Bench in Fig.4, and more visualizations are provided in the Appendix.

4.2. 3D Reconstruction

Datasets. We resort to ETH benchmark [32] to demonstrate the effect on 3D reconstruction tasks. Following [7], we

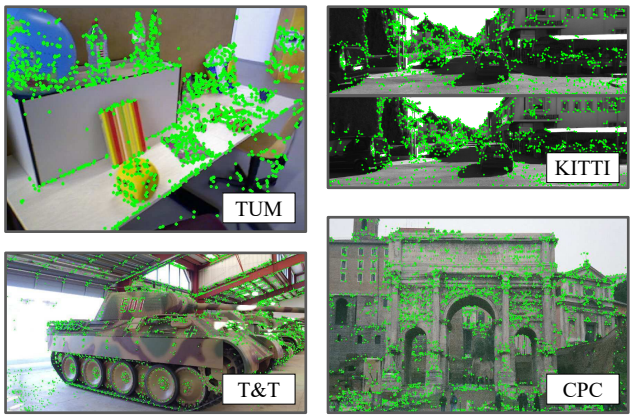

Figure 4. Sample detection results on FM-Bench [2] with top-5000 keypoints displayed.

Datasets	Methods	#Reg. Images	# Sparse Points	Track Length	Reproj. Error	#Dense Points
Madrid Metropolis 1344 images	*RootSIFT*	500	116K	6.32	**0.60px**	1.82M
	GeoDesc	495	**144K**	5.97	0.65px	1.56M
	SuperPoint	438	29K	9.03	1.02px	1.55M
	D2-Net (MS)	495	144K	6.39	1.35px	1.46M
	ASLFeat	613	96K	8.76	0.90px	**2.00M**
	ASLFeat (MS)	**649**	129K	**9.56**	0.95px	1.92M
Gendarmen-markt 1463 images	*RootSIFT*	1035	338K	5.52	**0.69px**	**4.23M**
	GeoDesc	1004	**441K**	5.14	0.73px	3.88M
	SuperPoint	967	93K	7.22	1.03px	3.81M
	D2-Net (MS)	965	310K	5.55	1.28px	3.15M
	ASLFeat	1040	221K	8.72	1.00px	4.01M
	ASLFeat (MS)	**1061**	320K	**8.98**	1.05px	4.00M
Tower of London 1576 images	*RootSIFT*	804	239K	7.76	**0.61px**	3.05M
	GeoDesc	776	**341K**	6.71	0.63px	2.73M
	SuperPoint	681	52K	8.67	0.96px	2.77M
	D2-Net (MS)	708	287K	5.20	1.34px	2.86M
	ASLFeat	821	222K	12.52	0.92px	3.06M
	ASLFeat (MS)	**846**	252K	**13.16**	0.95px	**3.08M**

Table 4. Evaluation results on ETH benchmark [32] for 3D reconstruction.

evaluate on three medium-scale datasets from [38].

Evaluation protocols. We exhaustively match all image pairs for each dataset with both ratio test at 0.8 and mutual check for outlier rejection, then run SfM and MVS algorithms by COLMAP [31]. For sparse reconstruction, we report the number of registered images (*#Reg. Images*), the number of sparse points (*#Sparse Points*), average track length (*Track Length*) and mean reprojection error (*Reproj. Error*). For dense reconstruction, we report the number of dense points (*#Dense Points*). We limit the maximum number of features of ASLFeat to 20K.

Results. As shown in Tab. 4, ASLFeat produces the most complete reconstructions regarding *#Reg. Images* and *#Dense Points*. Besides, ASLFeat results in *Reproj. Error* that is on par with SuperPoint and smaller than D2-Net, which again validates the effect of the proposed MulDet for restoring spatial information. However, the reprojection error produced by hand-crafted keypoints (e.g., RootSIFT) is still notably smaller than all learning methods, which implies that future effort can be spent to further improve the keypoint localization in a learning framework.

4.3. Visual Localization

Datasets. We resort to Aachen Day-Night dataset [30] to demonstrate the effect on visual localization tasks, where the key challenge lies on matching images with extreme day-night changes for 98 queries.

Evaluation protocols. We use the evaluation pipeline provided in *The Visual Localization Benchmark*[3], which takes custom features as input, then relies on COLMAP [31] for image registration, and finally generates the percentages of successfully localized images within three error tolerances (0.5m, 2°) / (1m, 5°) / (5m, 10°). The maximum feature number of our methods are limited to 20K.

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

Results. As shown in Tab. 5, although only mediocre results are obtained in previous evaluations, D2-Net performs surprisingly well under challenging illumination variations. This can be probably ascribed to the superior robustness of low-level features pre-trained on ImageNet. On the other hand, our method outperforms the plain implementation of R2D2, while a specialized R2D2 model (*R2D2 (fine-tuned)*) achieves the state-of-the-art results with doubled model size, training on day image from Aachen dataset and using photo-realistic style transfer to generate night images.

Methods	#Features	Dim	0.5m, 2°	1m, 5°	5m, 10°
RootSIFT	11K	128	33.7	52.0	65.3
HAN + HN++	11K	128	37.8	54.1	75.5
SIFT + ContextDesc	11K	128	40.8	55.1	80.6
SuperPoint	7K	256	42.8	57.1	75.5
D2-Net (MS)	19K	512	44.9	64.3	**88.8**
R2D2 (MS)	10K	128	43.9	61.2	77.6
R2D2 (MS, fine-tuned)	10K	128	**45.9**	66.3	**88.8**
D2-Net (our impl.)	10K	128	40.8	59.2	77.6
ASLFeat	10K	128	**45.9**	64.3	86.7
ASLFeat (MS)	10K	128	44.9	64.3	85.7

Table 5. Evaluation results on Aachen Day-Night dataset [30] for visual localization.

5. Conclusions

In this paper, we have used D2-Net as the backbone architecture to jointly learn the local feature detector and descriptor. Three light-weight yet effective modifications have been proposed that drastically boost the performance in two aspects: the ability to model the local shape for stronger geometric invariance, and the ability to localize keypoints accurately for solving robust camera geometry. We have conducted extensive experiments to study the effect of each modification, and demonstrated the superiority and practicability of our methods across various applications.

Acknowledgments. This work is supported by Hong Kong RGC GRF 16206819, 16203518 and T22-603/15N.

References

[1] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk. Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *CVPR*, 2017. 2, 5, 6, 7

[2] J.-W. Bian, Y.-H. Wu, J. Zhao, Y. Liu, L. Zhang, M.-M. Cheng, and I. Reid. An evaluation of feature matchers forfundamental matrix estimation. *BMVC*, 2019. 2, 5, 6, 7, 8

[3] C. Choy, J. Park, and V. Koltun. Fully convolutional geometric features. In *ICCV*, 2019. 5, 6

[4] P. H. Christiansen, M. F. Kragh, Y. Brodskiy, and H. Karstoft. Unsuperpoint: End-to-end unsupervised interest point detector and descriptor. *arXiv*, 2019. 1, 2

[5] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In *ICCV*, 2017. 2, 3

[6] D. DeTone, T. Malisiewicz, and A. Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPRW*, 2018. 1, 2, 4, 6, 7

[7] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler. D2-net: A trainable cnn for joint detection and description of local features. In *CVPR*, 2019. 1, 2, 3, 4, 5, 6, 7

[8] P. Ebel, A. Mishchuk, K. M. Yi, P. Fua, and E. Trulls. Beyond cartesian representations for local descriptors. *ICCV*, 2019. 2

[9] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 5, 7

[10] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017. 1

[11] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 4

[12] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *NeurIPs*, 2015. 1, 2

[13] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ToG*, 2017. 6, 7

[14] A. B. Laguna, E. Riba, D. Ponsa, and K. Mikolajczyk. Key.net: Keypoint detection by handcrafted and learned cnn filters. *ICCV*, 2019. 2

[15] S. Li, L. Yuan, J. Sun, and L. Quan. Dual-feature warping-based motion model estimation. In *ICCV*, 2015. 1

[16] G. Lin, A. Milan, C. Shen, and I. Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *CVPR*, 2017. 1

[17] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 1, 2, 3, 4, 6, 7

[18] Z. Luo, T. Shen, L. Zhou, J. Zhang, Y. Yao, S. Li, T. Fang, and L. Quan. Contextdesc: Local descriptor augmentation with cross-modality context. In *CVPR*, 2019. 2, 5, 6, 7

[19] Z. Luo, T. Shen, L. Zhou, S. Zhu, R. Zhang, Y. Yao, T. Fang, and L. Quan. Geodesc: Learning local descriptors by integrating geometry constraints. In *ECCV*, 2018. 2, 5

[20] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *ECCV*, 2002. 1, 3

[21] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *NeurIPs*, 2017. 2, 6, 7

[22] D. Mishkin, F. Radenovic, and J. Matas. Repeatability is not enough: Learning affine regions via discriminability. In *ECCV*, 2018. 1, 2, 3, 6, 7

[23] K. Moo Yi, Y. Verdie, P. Fua, and V. Lepetit. Learning to assign orientations to feature points. In *CVPR*, 2016. 1, 2, 3

[24] T. Nguyen, S. W. Chen, S. S. Shivakumar, C. J. Taylor, and V. Kumar. Unsupervised deep homography: A fast and robust homography estimation model. In *IEEE Robotics and Automation Letters*, 2018. 4

[25] Y. Ono, E. Trulls, P. Fua, and K. M. Yi. Lf-net: learning local features from images. In *NeurIPs*, 2018. 1, 2, 3, 4, 6, 7

[26] F. Radenović, G. Tolias, and O. Chum. Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples. In *ECCV*, 2016. 5

[27] J. Revaud, P. Weinzaepfel, C. De Souza, N. Pion, G. Csurka, Y. Cabon, and M. Humenberger. R2d2: Repeatable and reliable detector and descriptor. In *NeurIPs*, 2019. 1, 2, 4, 5, 6, 7

[28] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 2015. 1, 2, 4

[29] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski. Orb: An efficient alternative to sift or surf. In *ICCV*, 2011. 1, 2, 3

[30] T. Sattler, T. Weyand, B. Leibe, and L. Kobbelt. Image retrieval for image-based localization revisited. In *BMVC*, 2012. 1, 2, 8

[31] J. L. Schonberger and J.-M. Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 1, 8

[32] J. L. Schonberger, H. Hardmeier, T. Sattler, and M. Pollefeys. Comparative evaluation of hand-crafted and learned local features. In *CVPR*, 2017. 2, 7, 8

[33] T. Shen, Z. Luo, L. Zhou, R. Zhang, S. Zhu, T. Fang, and L. Quan. Matchable image retrieval by learning from surface reconstruction. In *ACCV*, 2018. 5

[34] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2014. 5, 6

[35] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *IROS*, 2012. 5, 7

[36] Y. Tian, B. Fan, and F. Wu. L2-net: Deep learning of discriminative patch descriptor in euclidean space. In *CVPR*, 2017. 2, 5, 6

[37] Y. Tian, X. Yu, B. Fan, F. Wu, H. Heijnen, and V. Balntas. Sosnet: Second order similarity regularization for local descriptor learning. In *CVPR*, 2019. 2

[38] K. Wilson and N. Snavely. Robust global translations with 1dsfm. In *ECCV*, 2014. 6, 7, 8

[39] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua. Lift: Learned invariant feature transform. In *ECCV*, 2016. 1, 2

[40] J. Zhang, D. Sun, Z. Luo, A. Yao, L. Zhou, T. Shen, Y. Chen, L. Quan, and H. Liao. Learning two-view correspondences and geometry using order-aware network. In *ICCV*, 2019. 1, 2

[41] L. Zhang and S. Rusinkiewicz. Learning to detect features in texture images. In *CVPR*, 2018. 4

[42] R. Zhang, S. Tang, Y. Zhang, J. Li, and S. Yan. Scale-adaptive convolutions for scene parsing. In *ICCV*, 2017. 2

[43] R. Zhang, S. Zhu, T. Fang, and L. Quan. Distributed very large scale bundle adjustment by global camera consensus. In *ICCV*, 2017. 1

[44] L. Zhou, S. Zhu, Z. Luo, T. Shen, R. Zhang, M. Zhen, T. Fang, and L. Quan. Learning and matching multi-view descriptors for registration of point clouds. In *ECCV*, 2018. 1

[45] X. Zhu, H. Hu, S. Lin, and J. Dai. Deformable convnets v2: More deformable, better results. In *CVPR*, 2019. 2, 3