Learning to Make Keypoints Sub-Pixel Accurate

Shinjeong Kim[1], Marc Pollefeys[1,2], and Daniel Barath[1]

[1] Department of Computer Science, ETH Zürich
[2] Microsoft Mixed Reality and AI Zurich lab
{shikim, marc.pollefeys, danielbela.barath}@ethz.ch

Abstract. This work addresses the challenge of sub-pixel accuracy in detecting 2D local features, a cornerstone problem in computer vision. Despite the advancements brought by neural network-based methods like SuperPoint and ALIKED, these modern approaches lag behind classical ones such as SIFT in keypoint localization accuracy due to their lack of sub-pixel precision. We propose a novel network that enhances *any* detector with sub-pixel precision by learning an offset vector for detected features, thereby eliminating the need for designing specialized sub-pixel accurate detectors. This optimization directly minimizes test-time evaluation metrics like relative pose error. Through extensive testing with both nearest neighbors matching and the recent LightGlue matcher across various real-world datasets, our method consistently outperforms existing methods in accuracy. Moreover, it adds only around 7 ms to the time of a particular detector. The code is available at https://github.com/KimSinjeong/keypt2subpx.

Keywords: Local Features · Localization · Sub-pixel Accuracy

1 Introduction

The task of identifying and matching sparse 2D feature points across images has long been a cornerstone problem in computer vision [23]. Algorithms for feature detection enable the generation of detailed 3D models from collections of images [26, 58, 68], the construction of maps for robotic navigation [43, 44], the recognition of places [33, 46, 57], and the estimation of precise locations [32, 56, 65], as well as facilitating object recognition [2, 3, 38, 45, 49]. Given its critical importance, the design of algorithms for feature detection and description has attracted considerable attention in computer vision. While the seminal SIFT algorithm [38] has been the benchmark for feature detection pipelines for over three decades, the attention has shifted towards learned methods in recent years, with SuperPoint emerging as the gold standard for various applications.

The introduction of advanced machine learning tools has led researchers to substitute traditional, feature-based vision systems with neural networks [19, 73, 74]. These networks demonstrate increased robustness to variations in features, like changes in viewpoint and illumination, by being explicitly trained on such scenarios with various augmentation techniques. However, independent evaluations indicate that these learned models have yet to achieve the keypoint localization precision of their classical predecessors [38]. This discrepancy is largely

due to the absence of mechanisms for ensuring sub-pixel accuracy—a fundamental aspect of the long-standing status of the SIFT algorithm as the benchmark in feature detection. Additionally, most recent feature detection frameworks design training protocols to mimic the complex conditions faced during the application of feature detectors in vision tasks. Only a select few [10, 60] are tailored to optimize directly for the specific task they are intended to address.

In this work, we present a network capable of augmenting any learned feature set to deliver sub-pixel precision in keypoint detection. The proposed method is structured to append an offset vector to detected features, thereby enabling sub-pixel accuracy without the need for developing entirely new feature detectors. This offset is meticulously learned to directly minimize test-time evaluation metrics, for instance, relative pose error. We demonstrate our improved accuracy through the gold standard SuperPoint [19] and the recent ALIKED [73] features across several datasets. These tests include both nearest neighbors feature matching and the advanced LightGlue matcher [36]. Results show that our approach consistently improves accuracy metrics in a variety of real-world settings, both indoor and outdoor.

2   Related Work

Geometrically Invariant Feature Extraction. Initial efforts in local feature detection and description concentrated on developing carefully engineered algorithms to identify distinctive keypoints and their descriptors, which are robust to variations in viewpoint and illumination. Hand-crafted techniques, such as Harris corners [23], SIFT [38], ORB [54], and others [1, 9, 12, 18, 31, 53], utilize explicit geometric concepts like corners, gradients, and scale-space extrema. Such approaches became so popular, mainly due to their efficiency and robustness to illumination changes, that they are still crucial components of state-of-the-art frameworks, *e.g.*, for SLAM [43] and Structure-from-Motion [58].

Recent advances in deep learning have led to two primary approaches: patch-based descriptor extraction and the integrated learning of keypoints and descriptors. Patch-based techniques [4, 22, 40, 63, 64], along with the majority of combined keypoint and descriptor learning strategies [19, 20, 66], incorporate data augmentation to ensure invariance to scale and orientation changes. Certain combined learning approaches go further by explicitly determining the orientation and scale for keypoints. LIFT [69], for instance, mimics the SIFT methodology [38] by identifying keypoints, estimating their orientations, and extracting descriptors via separate neural networks. It calculates the rotation and scale using a neural network, subsequently applying these transformations to the features to achieve descriptors invariant to orientation and scale changes.

AffNet [41], UCN [13], and LF-Net [47] advance this approach by estimating affine parameters and applying affine transformations on image features through Spatial Transformer Networks (STN) [25], thereby extracting affine invariant descriptors. GIFT [37] innovates by creating groups of images at varying scales and orientations, from which it then derives features to produce descriptors that are
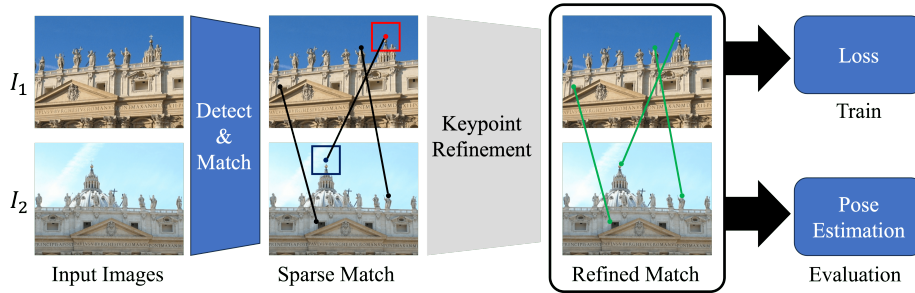
Fig. 1: An overview of the proposed sub-pixel refinement method. Given a pair of images, local features are detected, described, and matched to find correspondences. For each match, image patches centered at the keypoints are extracted. Our proposed Keypoint Refinement module (Fig. 2) takes the patches and descriptors to refine the keypoint locations. When training, the refined keypoint matches are used to calculate loss so that the Keypoint Refinement module can be optimized. On evaluation, the relative pose between the two views is estimated using robust estimators.

invariant to scale and orientation. Additionally, HDD-Net [8] proposes an alternative by rotating convolution kernels instead of the features, aiming to extract descriptors invariant to rotations. REKD [29] and RELF [30] alternatively utilized a rotation-equivariant convolution layer and devised self-supervised learning frameworks to learn a rotation-equivariant detector and rotation-invariant descriptor, respectively.

Joint Keypoint and Descriptor Learning. A number of studies have proposed methods to simultaneously estimate both the score map, from which keypoints are detected, and the descriptor map, from which descriptors are sampled. SuperPoint [19] introduces a lightweight network architecture trained on image pairs augmented automatically through Homographic Adaptation. R2D2 [51] innovates further by calculating repeatability and reliability maps for keypoint detection and employing AP loss for descriptor training. A notable enhancement by Suwichaya involves augmenting R2D2 with a Low-Level Feature (LLF) detector to refine keypoint precision [61]. ALIKE [74] distinguishes itself with a differentiable keypoint detection module, enabling precise keypoint training within a highly efficient network suitable for real-time applications. Its successor, ALIKED [73], incorporates deformable convolutional networks for enhanced performance. While ALIKED obtains sub-pixel accurate features through employing the SoftArgMax operator, it is not explicitly designed to optimize keypoint localization precision. D2-Net [20], diverging from score map estimation, identifies keypoints through channel and spatial maxima within the feature map, albeit with a noted limitation in keypoint localization accuracy due to low-resolution feature maps. ASLFeat [39] leverages multi-level features and deformable convolutions to detect keypoints and model local shapes, thereby improving localization accuracy and descriptor quality. D2D [62], drawing inspiration from D2-Net, utilizes a descriptor map alongside absolute and relative saliency for keypoint detection. Rao et al. [50] introduce a hierarchical view consistency approach to
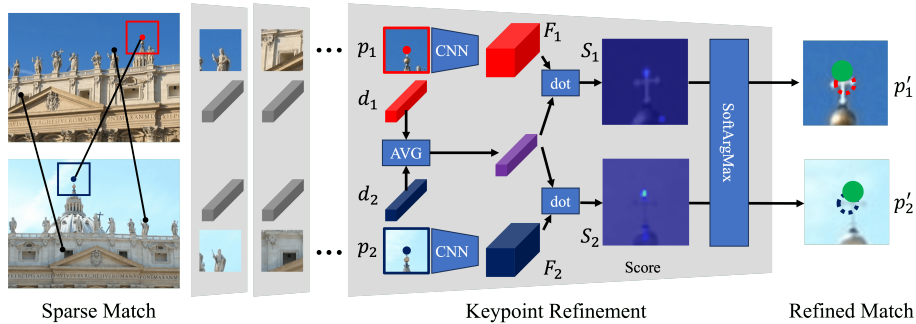
Fig. 2: Visualization of how our Keypoint Refinement module works. For detectors producing dense score maps as an intermediate representation, the patches of the score map are concatenated to the image patches. The feature map of each patch is extracted with a small convolutional neural network (CNN) and dot-produced with an average of matched descriptor pairs. Taking the SoftArgMax operation on the resulting score map gives the sub-pixel accurate displacement of each keypoint. Note that the weights of two CNNs of the Keypoint Refinement module are shared.

generalize feature descriptors for visual measurements. While these detectors achieve impressive results across a range of datasets, the keypoint localization accuracy is often poor, as they do not explicitly aim at sub-pixel accuracy.

Feature Detection Trained for Specific Tasks. The majority of learning-based works in feature detection craft training schemes that replicate the challenging conditions encountered when deploying a feature detector for a specific vision task. Contrarily, Key.Net [60] introduces a differentiable pipeline adept at autonomously identifying category-level keypoints, specifically tailored for the task of relative pose estimation. DISK [66] utilizes reinforcement learning to train the score and descriptor maps, focusing on the end results of the pipeline to inform its learning process. Bhowmik et al. [10] further integrate feature detection and description within a comprehensive vision pipeline, where the model encounters real-world challenges inherently during training. Similarly to DISK, this approach leverages reinforcement learning to address the discrete nature of keypoint selection and descriptor matching. Its performance is showcased throughout improving the SuperPoint detector [19]. Additionally, Roessle et al. [52] concentrate on simultaneously improving feature matching and pose optimization by applying differentiable pose estimation techniques.

3   Sub-Pixel Accurate Local Features

We provide a generaliable method to refine any keypoints to be sub-pixel accurate. To this end, we propose a detector-agnostic Keypoint Refinement module guided by local feature descriptors and optimizing a two-view geometric objective directly. Our pipeline is designed in a way not only to distill the context information of descriptors to our Keypoint Refinement module but also to consider geometric supervision. In this section, we will first introduce our sub-pixel

accurate Keypoint Refinement module, followed by the geometric supervision we are providing.

3.1   Sub-Pixel Accurate Keypoint Refinement

In this section, we discuss the proposed Keypoint Refinement module to learn an offset vector on top of a detected feature. Following de-facto standard process of relative pose estimation based on sparse correspondences, local features are first detected, described, and matched given an image pair. Based on the matched feature pairs, we perform our keypoint refinement. The brief overview of the proposed architecture is given in Fig. 2.

Next, we describe the refinement procedure for each tentative point correspondence. As discussed in several works [35,73,74], local refinement of keypoints can be safely assumed to be independent of the global context. Based on the same assumption, we consider a certain size of square windows centering the detected keypoints as our region of interest. We extract the region of interest of the image as patches. For the keypoint detectors producing score maps as an intermediate representation, we extract the score patches on the same region and concatenate them with image patches to consider them as side information.

To refine a feature based on the extracted patch centered on the original location of the feature point, we consider the keypoint refinement procedure as predicting new or refining an existing score map. To detect keypoints from predicted score maps, either Non-Maximum Suppression (NMS) or its differentiable counterpart, SoftArgMax, are the standard approaches. To ensure that the gradient effectively flows through our module, we choose SoftArgMax. Given score map $S \in \mathbb{R}^{P \times P}$, SoftArgMax is calculated as follows:

$$\text{SoftArgMax}(S) = \sum_{u \in \{-\lfloor \frac{P}{2} \rfloor, \cdots, \lfloor \frac{P}{2} \rfloor\}^2} \text{SoftMax}(S)[u]u = \sum_{u \in \{-\lfloor \frac{P}{2} \rfloor, \cdots, \lfloor \frac{P}{2} \rfloor\}^2} \frac{e^{S[u]}u}{\sum_{u'} e^{S[u']}} \quad (1)$$

This can be viewed as an expectation of relative keypoint location inside the score patch, where the probability distribution is modeled with SoftMax.

Our score map prediction procedure is modeled with a Convolutional Neural Network (CNN). We extract D-dimensional $P \times P$ square-shape normalized features from the image patches, $F_{1i}, F_{2i} \in \mathbb{R}^{P \times P \times D}$ where F_{1i} and F_{2i} correspond to the i-th patch of the first image and the second image, respectively. We do not assume the convolution layers to have the 'same' padding, that is, P does not have to be the dimension of the original image patch used for feature extraction. Once we have the feature patches along with corresponding descriptors $d_{1i}, d_{2i} \in \mathbb{R}^D$, we calculate the displacement $\delta_{1i}, \delta_{2i} \in \mathbb{R}^2$ of each keypoint $p_{1i}, p_{2i} \in \{0, \cdots, W\} \times \{0, \cdots, H\}$ in a way to cover the whole input image patch, employing the following equation:

$$S_{ki}[u] = F_{ki}[u] \cdot \frac{d_{1i} + d_{2i}}{2} \ (k \in \{1, 2\}), \tag{2}$$

$$\delta_{ki} = \sigma \times \text{SoftArgMax}(S_{ki}). \tag{3}$$

The σ is the scale factor to scale the displacement calculated with the feature map back to the scale of the extracted image patch.

The dot product of Eq. (2) guides F_{1i}, F_{2i} to learn the embedding space of the descriptor but in pixel-level grained, which is far denser than the initial descriptors. The average of two descriptors acts as a regularization on the learned embedding space to be consistent over multiple views, which is a special case of the robust mean of features of PixSfM [35]. The effect of this will be later shown in the experiments. The Eq. (2) and Eq. (3) altogether can also be interpreted as attention with the descriptors as queries, feature patches as keys, and pixel indices inside the patch as values.

The refined keypoints we finally obtain are the sum of the original keypoint and predicted displacement vector as follows:

$$p'_{1i} = p_{1i} + \delta_{1i}, \quad p'_{2i} = p_{2i} + \delta_{2i}, \tag{4}$$

where p'_{1i} and p'_{2i} are the updated keypoint locations in the first and second images, respectively. During training, gradients only flow toward keypoint displacements $(\delta_{1i}, \delta_{2i})$, as we do not assume the differentiability of the detectors and descriptors we use. This property makes the proposed method applicable in conjunction with any existing detector.

3.2   Training Objective

Our proposed Keypoint Refinement module is directly trained with a geometric objective. Here, we explain the formulation and effect of our objective. We will conclude this section with why in the previous works, it was difficult to train detectors only with such an objective.

We aim to train the Keypoint Refinement module without any exhaustive annotation on keypoints to enhance the subsequent relative pose estimation. While many keypoint detectors utilize consistency under various data augmentation to self-supervise the keypoint locations, our supervision is based on multi-view consistency measured by the epipolar error introduced in the supplementary material of [11]. Given an image pair and its ground truth essential matrix E, and matched homogeneous keypoint pairs $\hat{p}_1$ and $\hat{p}_2$, the epipolar error is calculated as follows:

$$e(\hat{p}_1, \hat{p}_2, E) = \frac{(\hat{p}_2^T E \hat{p}_1)^2}{[E\hat{p}_1]_0^2 + [E\hat{p}_1]_1^2 + [E^T\hat{p}_2]_0^2 + [E^T\hat{p}_2]_1^2}, \tag{5}$$

where $[\cdot]_j$ indicates the j-th element of the vector inside the bracket.

Directly optimizing the epipolar error can result in an unstable training procedure, as the sum of the errors of all matches is dominated by outliers and thus the keypoints will be optimized toward outliers. To minimize the effect of outliers, we set a threshold t' and only consider errors smaller than that to calculate the loss. Therefore, our epipolar loss is,

$$\ell(\hat{p}_1, \hat{p}_2, E) = (e(\hat{p}_1, \hat{p}_2, E) \cdot \mathbb{1}[d(\hat{p}_1, \hat{p}_2, E) < t']) + (t' \cdot \mathbb{1}[d(\hat{p}_1, \hat{p}_2, E) \geq t']), \tag{6}$$

Table 1: Relative pose accuracy on datasets MegaDepth [34], KITTI [21] and Scan-Net [17] by running GC-RANSAC [6] on correspondences extracted by vanilla Super-Point [19] (SP), SP refined by the proposed method and by [10] and matched by Mutual Nearest Neighbor (MNN) matching. We report the AUC scores, calculated from the relative pose errors, thresholded at 5°, 10°, and 20°, the mean and median pose errors in degrees, and the inlier ratio of the estimated relative pose. The proposed method consistently outperforms SuperPoint and the reinforcement learning-based approach [10] in all metrics, except for the mean error on ScanNet, where all methods are similar.

Dataset	Method	AUC@5°	AUC@10°	AUC@20°	Inlier Ratio (%)	Mean (°)	Median (°)
	SP [19] + MNN	35.34	45.37	54.22	33.78	27.89	3.94
MegaDepth	Reinforced SP [10] + MNN	36.61	47.56	56.71	33.90	27.91	3.92
	SP + MNN + Ours	**37.16**	**48.07**	**57.15**	**34.42**	**27.69**	**3.82**
	SP [19] + MNN	80.92	89.49	94.41	73.27	1.20	0.62
KITTI	Reinforced SP [10] + MNN	80.97	89.53	94.43	73.27	1.20	0.62
	SP + MNN + Ours	**81.36**	**89.69**	**94.49**	**73.47**	**1.16**	**0.61**
	SP [19] + MNN	13.92	28.16	43.11	37.56	**28.58**	9.76
ScanNet	Reinforced SP [10] + MNN	13.83	28.11	43.17	37.55	28.60	9.76
	SP + MNN + Ours	**14.16**	**28.57**	**43.56**	**37.66**	28.68	**9.59**

where

$$d(\hat{p}_1, \hat{p}_2, E) = \sqrt{e(\hat{p}_1, \hat{p}_2, E)}, \tag{7}$$

with operator $\mathbb{1}[\cdot]$ being the Iverson bracket which equals one if the condition inside holds and zero otherwise.

In the calibrated case, we set the threshold t' in a geometrically meaningful way. As our choice of robust estimator, GC-RANSAC with 1 pixel threshold internally uses MSAC score of which margin for inlier selection as 1.5 times of the threshold, we set t' to correspond to 1.5 pixel in the normalized image space. Following the standard practice, *e.g.*, as done in OpenCV and MAGSAC++ [7], we normalize the threshold $t = 1.5$ by the focal lengths as follows:

$$t' = \frac{t}{(f_{1x} + f_{1y} + f_{2x} + f_{2y})/4}, \tag{8}$$

where f_{ix} and f_{iy} are the focal lengths of image i corresponding to the x and y axes, respectively.

The similar losses based on epipolar geometry and essential matrix are frequently used on optimizing robust estimators [67] or local feature matchers [11, 70–72], however, has yet been applied for keypoint detection.

The challenge of achieving differentiation has been a significant barrier to enhancing feature detection methods with sub-pixel precision. Previous attempts struggled with differentiating components of the vision pipeline, such as detectors like SuperPoint [19] or matchers like Nearest Neighbor Matching. These efforts often encountered unstable training due to the complexities of integrating detectors and matches. Successful differentiation in prior work was either achieved by forgoing end-to-end differentiability through the use of the REINFORCE algorithm [10] or was only demonstrated in simple, driving scenarios [27]. Our

Table 2: Relative pose accuracy on datasets MegaDepth [34], KITTI [21] and Scan-Net [17] by running GC-RANSAC [6] on SuperPoint [19] or ALIKED [73] features matched by LightGlue [36] and refined by the proposed method. We report the AUC scores, calculated from the relative pose errors, thresholded at 5°, 10°, and 20°, the mean and median pose errors in degrees, and the inlier ratio of the estimated relative pose. The proposed method consistently improves both detectors in almost all metrics.

Dataset	Method	AUC@5°	AUC@10°	AUC@20°	Inlier Ratio (%)	Mean (°)	Median (°)
MegaDepth	ALIKED+LG	64.69	76.77	85.17	87.10	6.65	1.01
	ALIKED+LG+Ours	**65.34**	**77.27**	**85.52**	**87.73**	**6.57**	**0.98**
	SP+LG	59.82	72.88	82.31	79.34	7.39	1.25
	SP+LG+Ours	**61.82**	**74.46**	**83.41**	**81.24**	**7.22**	**1.15**
KITTI	ALIKED+LG	84.24	91.45	**95.36**	79.88	1.24	**0.50**
	ALIKED+LG+Ours	**84.27**	**91.46**	**95.36**	**80.02**	**1.20**	**0.50**
	SP+LG	81.50	89.68	94.37	81.28	1.40	0.59
	SP+LG+Ours	**81.99**	**89.95**	**94.51**	**81.59**	**1.34**	**0.57**
ScanNet	ALIKED+LG	20.99	38.68	54.81	70.92	**20.35**	5.86
	ALIKED+LG+Ours	**21.21**	**38.90**	**55.00**	**70.98**	20.39	**5.82**
	SP+LG	19.40	37.96	55.46	85.51	20.07	5.97
	SP+LG+Ours	**19.71**	**38.26**	**55.80**	**85.67**	**19.84**	**5.89**

approach overcomes these issues by integrating our Keypoint Refinement module directly before the relative pose estimation stage of the pipeline, facilitating stable and effective differentiation.

4   Implementation Details

Network Architecture Details. Our Convolutional Neural Network responsible for refining (or producing, when used along with detectors involving no heatmaps) the score heatmap consists of a series of convolutional layers with 3×3 sized filters followed by ReLU activations, without any extra poolings in between. The numbers of the channel sizes are 16, 16, 64, 64, d, where d is the dimension of the descriptors. The 'same' padding is only applied on layers where the input and output channel sizes are the same. Thus, the dimension of the feature map starts from 11×11 and is eventually reduced to 5×5, which covers distances of 2 pixel toward each direction from the original keypoints. To scale this amount back to cover the entire region of the original patch, σ value for displacement scaling is set to 2.5, since $\frac{\lfloor 11/2 \rfloor}{\lfloor 5/2 \rfloor} = 2.5$. Thus, the resulting displacement vector covers distances of $2 \times 2.5 = 5$ pixel toward each direction. This enables our model to confine its refined keypoints inside its input image patch, and thus the refined keypoints do not diverge too far away from their original location, stabilizing the training of our module. The implementation of our model is based on PyTorch, taking only 123K (ALIKED) and 197K (SuperPoint) more parameters than the original detector.

Training Details. We train our networks with Adam optimizer with the default hyperparameters of PyTorch 1.10, except the learning rate which is set to 1×10^{-4}. Our pipeline is trained for 200K steps with a batch size of 8, taking approximately 12 hours with a decent modern GPU (NVIDIA RTX 3080Ti), which is also used for all evaluations. Our training is done on the training set of the MegaDepth [34] dataset. We use this model—trained for each detector—in *all* experiments, indoors and outdoors.

Evaluation Details. Our choice of robust estimator to estimate essential matrices is the state-of-the-art GC-RANSAC [6]. The official implementation of GC-RANSAC runs an inlier resampling-based local optimization [16] and ends with a Levenberg-Marquardt [42]. It employs PROSAC sampling [14] and the SPRT test [15] to achieve efficiency. We set the minimum and maximum iteration number to 1000 to minimize the effect of stochasticity and ran each experiment 3 times to report the average. We set the threshold of GC-RANSAC 1 px.

We evaluate the proposed sub-pixel refinement method in conjunction with the gold standard SuperPoint [19] and the recent ALIKED [73] detectors, with mutual nearest neighbors (MNN) and LightGlue [36] matching. We will compare to the original detectors and also to the reinforcement learning-based method proposed by Bhowmik et al. [10].

Evaluation Metrics. To assess the performance of a particular method, we follow the standard practice [28] and measure the relative pose error as the maximum of the rotation and translation errors. From these numbers, we will compute the Area Under the recall Curves (AUC) thresholded at 5°, 10°, and 20°. Additionally, we will report the mean, median of pose error, and inlier ratios.

5   Experiments

In this section, the evaluation is conducted on two outdoor datasets (MegaDepth and KITTI [21]) and one indoor dataset (ScanNet [17]). Additionally, we provide computational analysis and an analysis of the proposed method.

5.1   Relative Pose Estimation

To perform relative pose estimation, we run GC-RANSAC to estimate essential matrices. These matrices are then decomposed to relative rotation and translation by the standard SVD-based solution [24].

The **MegaDepth** dataset [34] is derived from a diverse collection of internet-sourced images depicting landmarks across the globe. This dataset leverages sophisticated structure-from-motion (SfM) and multi-view stereo (MVS) techniques to generate detailed reconstructions and depth maps for over 1000 distinct scenes. We utilize the benchmark consisting of 1500 image pairs, following the selection of LoFTR [59], as the validation set and the test set consisting of 655K image pairs coupled by [36,48] as our test set.

The results when using SuperPoint features and mutual nearest neighbors matching are reported in the top three rows of Table 1. The proposed refinement method consistently improves in all accuracy metrics. It improves upon

Table 3: Fundamental matrix accuracy on datasets MegaDepth [34], KITTI [21] and ScanNet [17] by running GC-RANSAC [6] on correspondences extracted by vanilla SuperPoint [19] (SP), SP refined by the proposed method and by [10] and matched by Mutual Nearest Neighbor (MNN) matching. We report the F1 scores, the inlier ratio, and the mean and median epipolar errors given the ground truth fundamental matrix. The proposed method consistently outperforms SuperPoint and the reinforcement learning-based approach [10] in all metrics, except for the mean error on ScanNet.

Dataset	Method	F1-score	Epi. Inlier Ratio (%)	Epi. Mean	Epi. Med
MegaDepth	SP [19] + MNN	47.64	18.64	**84.15**	1.69
	Reinforced SP [10] + MNN	47.62	18.65	84.32	1.70
	SP + MNN + Ours	**48.42**	**19.64**	84.41	**1.67**
KITTI	SP [19] + MNN	74.15	48.48	0.91	0.81
	Reinforced SP [10] + MNN	74.12	48.47	0.91	0.81
	SP + MNN + Ours	**75.48**	**51.82**	**0.87**	**0.77**
ScanNet	SP [19] + MNN	18.83	15.83	**45.86**	4.96
	Reinforced SP [10] + MNN	18.76	15.82	46.42	4.95
	SP + MNN + Ours	**18.91**	**16.23**	47.11	**4.83**

the vanilla SuperPoint, by approximately 2 AUC points on average. It also increases the inlier ratio and decreases both the mean and median errors. We also outperform the previous refinement method [10] in all evaluated metrics.

The results with SuperPoint and ALIKED feature when employing the recent LightGlue matcher are shown in the top rows of Table 2. Similarly to Table 1, the proposed method improves upon both ALIKED and SuperPoint. While the accuracy increment is sometimes small, it is consistent—we improve in all tested accuracy metrics.

The **KITTI** dataset [21] is a real-world benchmark for tasks stereo, optical flow, visual odometry, 3D object detection, and tracking. It is captured by driving in the city of Karlsruhe with accurate ground truth from the laser scanner and GPS localization system. We tested our method on the 11 visual odometry sequences that are provided with ground truth. The KITTI experiments are based on the odometry test set with 2790 image pairs defined in [27]. We form image pairs from the subsequent images.

The results of SuperPoint and MNN matching are shown in the middle three rows of Table 1. Similar to the results on the MegaDepth dataset, the proposed method consistently improves in all accuracy metrics. It improves upon the vanilla SuperPoint in AUC points, increases the inlier ratio, and decreases both the mean and median errors. We outperform a previous work that improved SuperPoint with a relative pose supervision [10] in all evaluated metrics.

The results with SuperPoint and ALIKED features when employing the recent LightGlue matcher are shown in the middle of Table 2. Similarly to Table 1, the proposed method improves upon both ALIKED and SuperPoint. A similar trend is as observable as on the other datasets. The proposed technique always improves accuracy, not by a huge margin, but consistently across all metrics.
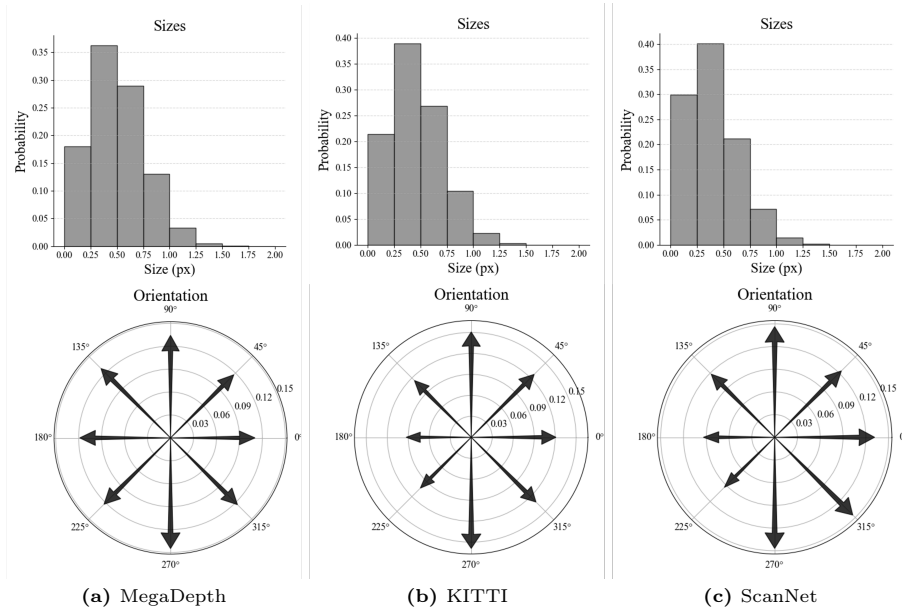
(a) MegaDepth   **(b)** KITTI   **(c)** ScanNet

Fig. 3: Histograms of scale and orientation of offset vectors for SuperPoint across various datasets. The top row illustrates the distribution of lengths of offset vectors, with significant displacements observed on MegaDepth, suggesting a high potential for accuracy improvements through sub-pixel refinement due to initial keypoint localization inaccuracies. The progression from left to right shows decreased lengths, with minimal adjustments required on ScanNet. The bottom row displays the directional histogram of predicted offset vectors, revealing a uniform distribution with a tendency for vertical alignment and a rightward bias, likely reflecting unique characteristics of datasets. We do not show the bins beyond offset size > 2px as they are negligible

The **ScanNet** dataset [17] contains 1613 monocular sequences with ground truth camera poses and depth maps. We evaluate the compared minimal solvers on the 1500 challenging pairs used in SuperGlue [55]. The results in the bottom three rows of Table 1 exhibit similar trends as on the other datasets. We consistently improve in almost all metrics. Here, the mean error increases marginally. However, all methods achieve similar mean errors.

The efficacy of our proposed method is further evidenced in the bottom section of Table 2, where results using SuperPoint and ALIKED features in conjunction with the recent LightGlue matcher are presented. Mirroring the improvements seen in Table 1, our method enhances the performance of both ALIKED and SuperPoint. This consistency aligns with trends observed on other datasets, where our technique uniformly bolsters accuracy. While the increments in performance are not huge, they are reliably observed across all evaluated metrics, underscoring the method's consistent contribution to accuracy enhancement.
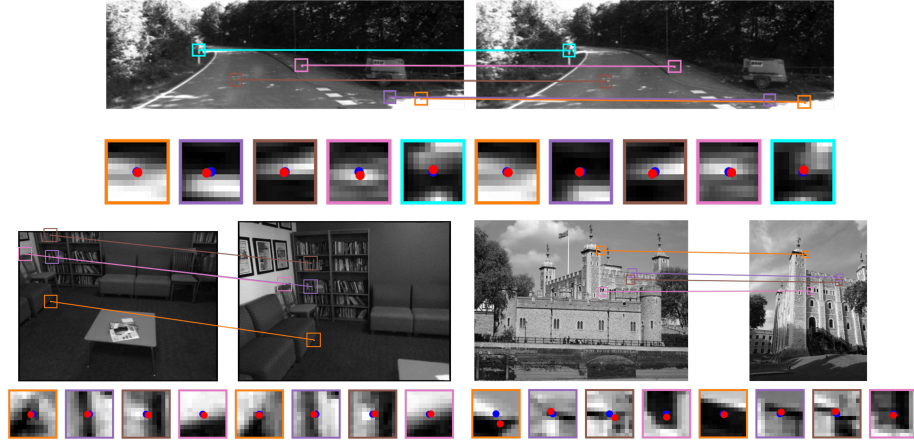
Fig. 4: Image pairs from KITTI (top), ScanNet (bottom left), and MegaDepth (bottom right) datasets showing inliers of refined matches. For each pair, the upper row shows matches, and the bottom one the image patches our Keypoint Refinement module takes, with initial points as blue and refined ones as red. The refinements are larger on the MegaDepth pair, as it is visible also in the analysis provided by Fig. 3. However, both on KITTI and ScanNet, the refined features seem to be better localized visually.

5.2   Fundamental Matrix Estimation

In this section, we extend the evaluation of our proposed method to assess its impact on fundamental matrix estimation, a critical task when camera intrinsics are unknown. Utilizing the identical datasets and experimental setup as outlined in the previous section, we employ GC-RANSAC [6] to derive fundamental matrices from the detected features. The assessment focuses on F1 scores, average inlier ratios, and both mean and median symmetric epipolar errors in relation to the ground truth fundamental matrix.

Findings, as detailed in Table 3, echo the enhancements observed in the relative pose accuracy experiments. The integration of the sub-pixel refinement mechanism with vanilla SuperPoint consistently improves performance across all evaluated accuracy metrics. An exception to this trend is noted in the mean error metric within the ScanNet dataset. It is pertinent to acknowledge that the mean error, while informative, is often regarded as less critical compared to other metrics due to its susceptibility to distortion by outliers; a single significant error can disproportionately affect the mean. Notably, in these experiments, the inclusion of reinforced SuperPoint [10] results in a decrement in accuracy. This observation underscores the nuanced effectiveness of sub-pixel refinement in enhancing feature-based estimations of fundamental matrices, highlighting the proposed method's broad applicability and potential for improving feature detection and matching processes.

Table 4: The computation time of each component of the relative pose estimation pipeline (Fig. 1). The first number of each cell represents the inference time taken on the component for 4096 keypoints (2048 matches) in milliseconds, and the parenthesized second number shows the portion of the inference time compared to the entire pipeline.

Method	Detection&Matching	Ours	Pose Estimation	Total
SP + MNN + Ours	72ms (37%)	7ms (4%)	116ms (60%)	195ms
ALIKED + MNN + Ours	163ms (51%)	6ms (2%)	150ms (47%)	319ms

Table 5: The number of parameters possessed by each component of the relative pose estimation pipeline (Fig. 1). The first number of each cell represents the number of parameters possessed by the component in millions, and the parenthesized second number shows the portion of the parameters compared to the entire pipeline.

Method	Detection	Matching	Ours	Total
SP+LG+Ours	1.30M (10%)	11.85M (89%)	0.20M (2%)	13.35M
ALIKED+LG+Ours	0.68M (5%)	11.85M (94%)	0.12M (1%)	12.65M

5.3   Keypoint Localization

To show that our refined keypoints are more precise in general, rather than merely rearranged for the setup of epipolar geometry, we evaluated the keypoint localization performance of our module on the HPatches benchmark [5]. We follow the setup described in LoFTR [59] and LightGlue [36]. Our method improved the combination of SuperPoint and mutual nearest neighbor matching (SP+MNN) from 27% to 33% Mean Matching Accuracy (MMA) at 1px.

5.4   Computational Analysis

To show that the proposed Keypoint Refinement module is not a computationally significant part of the whole pipeline, we measured the inference time (Table 4.) and number of parameters (Table 5.) of each component of the relative pose estimation pipeline. The inference time is measured for an image pair (2×2048 keypoints), averaged on the MegaDepth-1500 benchmark.

From Table 4, note that the computational time is less than 5% for both detectors used with the Mutual Nearest Neighbor (MNN) matching, and it is straightforward that the portion will be smaller when used with the LightGlue matcher. From Table 5, we chose LightGlue matcher, and it takes the most significant portion of the learnable parameters, while ours only takes around 2%. Even if LightGlue is replaced with MNN that has no parameters, ours only takes 17% of the detector.

5.5   Offset Analysis

To decipher the learning mechanisms and the characteristics of offset vectors within the proposed network, we delve into the visualized scale and orientation

histograms of the offsets predicted from SuperPoint keypoints, as illustrated across all tested datasets in Fig. 3. The upper part of the figures delineates the length distributions of the predicted displacement vectors, revealing pronounced displacements within the MegaDepth dataset. This observation aligns with the notable enhancements achieved on MegaDepth, implying a substantial potential for accuracy refinement through sub-pixel adjustments due to the initial imprecision in keypoint localization.

A progressive transition from left to right within these histograms demonstrates a trend towards diminished displacement lengths, indicating either an overall reduction in the necessity for keypoint adjustments or insufficient generalizability of our model similar to what neural-network based detectors have struggled.

Conversely, the lower section of Fig. 3 presents the directional tendencies of these predicted displacements, showcasing a relatively uniform distribution with a predominant alignment along the vertical image axis. A notable deviation is observed along the horizontal axis, with a bias towards rightward adjustments. This pattern likely reflects dataset-specific attributes, suggesting that the observed directional imbalances may be influenced by the inherent visual structure and orientation within the datasets.

Qualitative examples are shown in Fig. 4. Here, image pairs from KITTI (top), ScanNet (bottom left), and MegaDepth (bottom right) datasets are visualized together with their original and refined inlier matches. For each pair, the upper row shows matches overlayed in the image pair. The bottom one visualizes independent image patches our Keypoint Refinement module takes, with initial points as blue and refined ones as red. The refinements are visibly larger on the image pair taken from the MegaDepth dataset. However, both on KITTI and ScanNet, the refined features seem to be better localized.

6   Conclusion

This work introduces a novel post-processing procedure designed to endow any learned feature detection model with sub-pixel accuracy, addressing a critical limitation of recent detectors. By appending offset vectors to detected features, our method eliminates the need for designing specialized detectors while directly minimizing test-time evaluation metrics like relative pose error. Through rigorous testing on large-scale and real-world datasets with SuperPoint and ALIKED features and LightGlue matcher, we demonstrate consistent improvements in keypoint localization accuracy. This is shown through relative pose and fundamental matrix estimation. Although the performance improvements are not markedly large, they are *consistent* across various experiments, underscoring the reliability of our approach. Given the minimal computational overhead of our method, which adds only approx. 7 milliseconds to the processing time, it offers a compelling improvement with no discernible drawbacks.

Acknowledgement

References

1. Alahi, A., Ortiz, R., Vandergheynst, P.: Freak: Fast retina keypoint. In: 2012 IEEE conference on computer vision and pattern recognition. pp. 510–517. IEEE (2012)
2. Arandjelović, R., Zisserman, A.: Three things everyone should know to improve object retrieval. In: 2012 IEEE conference on computer vision and pattern recognition. pp. 2911–2918. IEEE (2012)
3. Arandjelovic, R., Zisserman, A.: All about vlad. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 1578–1585 (2013)
4. Balntas, V., Riba, E., Ponsa, D., Mikolajczyk, K.: Learning local feature descriptors with triplets and shallow convolutional neural networks. In: BMVC. vol. 1 (2016)
5. Balntas, V., Lenc, K., Vedaldi, A., Mikolajczyk, K.: 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 (2017)
6. Barath, D., Matas, J.: Graph-cut ransac. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6733–6741 (2018)
7. Barath, D., Noskova, J., Ivashechkin, M., Matas, J.: Magsac++, a fast, reliable and accurate robust estimator. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1304–1312 (2020)
8. Barroso-Laguna, A., Verdie, Y., Busam, B., Mikolajczyk, K.: Hddnet: Hybrid detector descriptor with mutual interactive learning. In: Proceedings of the Asian Conference on Computer Vision (November 2020)
9. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (surf). Computer vision and image understanding **110**(3), 346–359 (2008)
10. Bhowmik, A., Gumhold, S., Rother, C., Brachmann, E.: 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 (2020)
11. Brachmann, E., Rother, C.: Neural-guided ransac: Learning where to sample model hypotheses. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4322–4331 (2019)
12. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: Brief: Binary robust independent elementary features. In: European conference on computer vision. pp. 778–792. Springer (2010)
13. Choy, C.B., Gwak, J., Savarese, S., Chandraker, M.: Universal correspondence network. Advances in neural information processing systems **29** (2016)
14. Chum, O., Matas, J.: Matching with prosac-progressive sample consensus. In: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05). vol. 1, pp. 220–226. IEEE (2005)

15. Chum, O., Matas, J.: Optimal randomized ransac. IEEE Transactions on Pattern Analysis and Machine Intelligence **30**(8), 1472–1482 (2008)
16. Chum, O., Matas, J., Kittler, J.: Locally optimized ransac. In: Pattern Recognition: 25th DAGM Symposium, Magdeburg, Germany, September 10-12, 2003. Proceedings 25. pp. 236–243. Springer (2003)
17. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Computer Vision and Pattern Recognition. pp. 5828–5839 (2017)
18. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05). vol. 1, pp. 886–893. IEEE (2005)
19. DeTone, D., Malisiewicz, T., Rabinovich, A.: Superpoint: Self-supervised interest point detection and description. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 224–236 (2018)
20. Dusmanu, M., Rocco, I., Pajdla, T., Pollefeys, M., Sivic, J., Torii, A., Sattler, T.: D2-Net: A Trainable CNN for Joint Detection and Description of Local Features. In: CVPR (2019)
21. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2012)
22. Han, X., Leung, T., Jia, Y., Sukthankar, R., Berg, A.C.: Matchnet: Unifying feature and metric learning for patch-based matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3279–3286 (2015)
23. Harris, C., Stephens, M., et al.: A combined corner and edge detector. In: Alvey vision conference. vol. 15, pp. 10–5244. Manchester, UK (1988)
24. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge university press (2003)
25. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. Advances in neural information processing systems **28** (2015)
26. Jared, H., Schonberger, J.L., Dunn, E., Frahm, J.M.: Reconstructing the world in six days. In: CVPR (2015)
27. Jau, Y.Y., Zhu, R., Su, H., Chandraker, M.: Deep keypoint-based camera pose estimation with geometric constraints. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 4950–4957 (2020). `https://doi.org/10.1109/IROS45743.2020.9341229`
28. Jin, Y., Mishkin, D., Mishchuk, A., Matas, J., Fua, P., Yi, K.M., Trulls, E.: Image matching across wide baselines: From paper to practice. International Journal of Computer Vision (2020)
29. Jongmin Lee, Byungjin Kim, M.C.: Self-supervised equivariant learning for oriented keypoint detection. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
30. Lee, J., Kim, B., Kim, S., Cho, M.: Learning rotation-equivariant features for visual correspondence. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2023)
31. Leutenegger, S., Chli, M., Siegwart, R.Y.: Brisk: Binary robust invariant scalable keypoints. In: 2011 International conference on computer vision. pp. 2548–2555. IEEE (2011)
32. Li, Y., Snavely, N., Huttenlocher, D., Fua, P.: Worldwide pose estimation using 3d point clouds. In: European conference on computer vision. pp. 15–29. Springer (2012)

33. Li, Y., Snavely, N., Huttenlocher, D.P.: Location recognition using prioritized feature matching. In: Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part II 11. pp. 791–804. Springer (2010)

34. Li, Z., Snavely, N.: Megadepth: Learning single-view depth prediction from internet photos. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2041–2050 (2018)

35. Lindenberger, P., Sarlin, P.E., Larsson, V., Pollefeys, M.: Pixel-Perfect Structure-from-Motion with Featuremetric Refinement. In: ICCV (2021)

36. Lindenberger, P., Sarlin, P.E., Pollefeys, M.: Lightglue: Local feature matching at light speed. ICCV (2023)

37. Liu, Y., Shen, Z., Lin, Z., Peng, S., Bao, H., Zhou, X.: Gift: Learning transformation-invariant dense visual descriptors via group cnns. Advances in Neural Information Processing Systems **32** (2019)

38. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60**(2), 91–110 (2004)

39. Luo, Z., Zhou, L., Bai, X., Chen, H., Zhang, J., Yao, Y., Li, S., Fang, T., Quan, L.: ASLFeat: Learning local features of accurate shape and localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (Apr 2020)

40. Mishchuk, A., Mishkin, D., Radenovic, F., Matas, J.: Working hard to know your neighbor's margins: Local descriptor learning loss. In: Advances in Neural Information Processing Systems (Jan 2018)

41. Mishkin, D., Radenovic, F., Matas, J.: Repeatability is not enough: Learning affine regions via discriminability. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 284–300 (2018)

42. Moré, J.J.: The levenberg-marquardt algorithm: implementation and theory. In: Numerical analysis: proceedings of the biennial Conference held at Dundee, June 28–July 1, 1977. pp. 105–116. Springer (2006)

43. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: Orb-slam: A versatile and accurate monocular slam system. IEEE Transactions on Robotics **31**, 1147–1163 (2015)

44. Mur-Artal, R., Tardós, J.D.: Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. IEEE transactions on robotics **33**(5), 1255–1262 (2017)

45. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06). vol. 2, pp. 2161–2168. Ieee (2006)

46. Noh, H., Araujo, A., Sim, J., Weyand, T., Han, B.: Large-scale image retrieval with attentive deep local features. In: Proceedings of the IEEE international conference on computer vision. pp. 3456–3465 (2017)

47. Ono, Y., Trulls, E., Fua, P., Yi, K.M.: Lf-net: Learning local features from images. In: Advances in Neural Information Processing Systems 31. pp. 6234–6244. Curran Associates, Inc. (2018)

48. Pautrat*, R., Suárez*, I., Yu, Y., Pollefeys, M., Larsson, V.: GlueStick: Robust Image Matching by Sticking Points and Lines Together. In: International Conference on Computer Vision (ICCV) (2023)

49. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: 2007 IEEE conference on computer vision and pattern recognition. pp. 1–8. IEEE (2007)

50. Rao, Y., Yang, J., Ju, Y., Li, C., Rigall, E., Fan, H., Dong, J.: Learning general feature descriptor for visual measurement with hierarchical view consistency. IEEE Transactions on Instrumentation and Measurement **71**, 1–12 (2022)
51. Revaud, J., Weinzaepfel, P., De Souza, C., Pion, N., Csurka, G., Cabon, Y., Humenberger, M.: R2d2: Repeatable and reliable detector and descriptor. In: NeurIPS (2019)
52. Roessle, B., Nießner, M.: End2end multi-view feature matching with differentiable pose optimization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 477–487 (October 2023)
53. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In: European conference on computer vision. pp. 430–443. Springer (2006)
54. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: An efficient alternative to sift or surf. In: 2011 International conference on computer vision. pp. 2564–2571. IEEE (2011)
55. Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: Superglue: Learning feature matching with graph neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4938–4947 (2020)
56. Sattler, T., Leibe, B., Kobbelt, L.: Efficient & effective prioritized matching for large-scale image-based localization. IEEE transactions on pattern analysis and machine intelligence **39**(9), 1744–1756 (2016)
57. Schindler, G., Brown, M., Szeliski, R.: City-scale location recognition. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–7. IEEE (2007)
58. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4104–4113 (2016)
59. Sun, J., Shen, Z., Wang, Y., Bao, H., Zhou, X.: Loftr: Detector-free local feature matching with transformers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8922–8931 (2021)
60. Suwajanakorn, S., Snavely, N., Tompson, J.J., Norouzi, M.: Discovery of latent 3d keypoints via end-to-end geometric reasoning. In: NeurIPS (2018)
61. Suwanwimolkul, S., Komorita, S., Tasaka, K.: Learning of low-level feature keypoints for accurate and robust detection. In: 2021 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 2261–2270. IEEE, Waikoloa, HI, USA (Jan 2021)
62. Tian, Y., Balntas, V., Ng, T., Barroso-Laguna, A., Demiris, Y., Mikolajczyk, K.: D2D: Keypoint extraction with describe to detect approach. In: Proceedings of the Asian Conference on Computer Vision (2020)
63. Tian, Y., Fan, B., Wu, F.: L2-net: Deep learning of discriminative patch descriptor in euclidean space. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition. pp. 6128–6136. Honolulu, HI (Jul 2017)
64. Tian, Y., Yu, X., Fan, B., Wu, F., Heijnen, H., Balntas, V.: Sosnet: Second order similarity regularization for local descriptor learning. In: Conference on Computer Vision and Pattern Recognition (Dec 2019)
65. Toft, C., Stenborg, E., Hammarstrand, L., Brynte, L., Pollefeys, M., Sattler, T., Kahl, F.: Semantic match consistency for long-term visual localization. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 383–399 (2018)
66. Tyszkiewicz, M.J., Fua, P., Trulls, E.: Disk: Learning local features with policy gradient. In: NeurIPS (2020)

67. Wei, T., Patel, Y., Shekhovtsov, A., Matas, J., Barath, D.: Generalized differentiable ransac. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 17649–17660 (2023)
68. Wu, C.: Towards linear-time incremental structure from motion. In: 2013 International Conference on 3D Vision-3DV 2013. pp. 127–134. IEEE (2013)
69. Yi, K.M., Trulls, E., Lepetit, V., Fua, P.: Lift: Learned invariant feature transform. In: European Conference on Computer Vision. vol. 9910, pp. 467–483. Springer (2016)
70. Yi, K.M., Trulls, E., Ono, Y., Lepetit, V., Salzmann, M., Fua, P.: Learning to find good correspondences. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2666–2674 (2018)
71. Zhang, J., Sun, D., Luo, Z., Yao, A., Zhou, L., Shen, T., Chen, Y., Quan, L., Liao, H.: Learning two-view correspondences and geometry using order-aware network. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 5845–5854 (2019)
72. Zhao, C., Ge, Y., Zhu, F., Zhao, R., Li, H., Salzmann, M.: Progressive correspondence pruning by consensus learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6464–6473 (2021)
73. Zhao, X., Wu, X., Chen, W., Chen, P.C.Y., Xu, Q., Li, Z.: ALIKED: A lighter keypoint and descriptor extraction network via deformable transformation. IEEE Transactions on Instrumentation & Measurement **72**, 1–16 (2023). `https://doi.org/10.1109/TIM.2023.3271000`, `https://arxiv.org/pdf/2304.03608.pdf`
74. Zhao, X., Wu, X., Miao, J., Chen, W., Chen, P.C.Y., Li, Z.: Alike: Accurate and lightweight keypoint detection and descriptor extraction. IEEE Transactions on Multimedia (2022). `https://doi.org/10.1109/TMM.2022.3155927`, `http://arxiv.org/abs/2112.02906`