Guiding Local Feature Matching with Surface Curvature

Shuzhe Wang[1]* Juho kannala[1] Marc Pollefeys[2,3] Daniel Barath[2]

[1]Aalto University [2]ETH zurich [3]Microsoft

Abstract

We propose a new method, called curvature similarity extractor (CSE), for improving local feature matching across images. CSE calculates the curvature of the local 3D surface patch for each detected feature point in a viewpoint-invariant manner via fitting quadrics to predicted monocular depth maps. This curvature is then leveraged as an additional signal in feature matching with off-the-shelf matchers like SuperGlue and LoFTR. Additionally, CSE enables end-to-end joint training by connecting the matcher and depth predictor networks. Our experiments demonstrate on large-scale real-world datasets that CSE consistently improves the accuracy of state-of-the-art methods. Fine-tuning the depth prediction network further enhances the accuracy. The proposed approach achieves state-of-the-art results on the ScanNet dataset, showcasing the effectiveness of incorporating 3D geometric information into feature matching.[1]

1. Introduction

Local feature matching is a crucial component for many geometric computer vision tasks, including visual localization [50, 51, 52, 48, 14, 66], structure-from-motion (SfM) [67, 53, 29], and simultaneous localization and mapping (SLAM) [38, 39, 10]. Given a pair of images observing a 3D scene, the task is to find reliable tentative point-to-point correspondences in the two images. Forming such feature matches is often a challenging task as the images may undergo large viewpoint and illumination changes, have occlusions or repetitive patterns.

The standard pipeline for local feature matching typically involves two steps: (1) keypoint detection and description and (2) point-wise feature matching. Traditional approaches mainly focus on improving the robustness of keypoint detection and description through, for example, extending the Harris [21] detector to handle affine transformations and multiple scales [35, 36] or using more discriminative or efficient descriptors [32, 7, 34, 1, 3]. However, despite their unbroken popularity, these algorithms often fail to cope with the challenges that arise in real-world environments, leading to low accuracy.

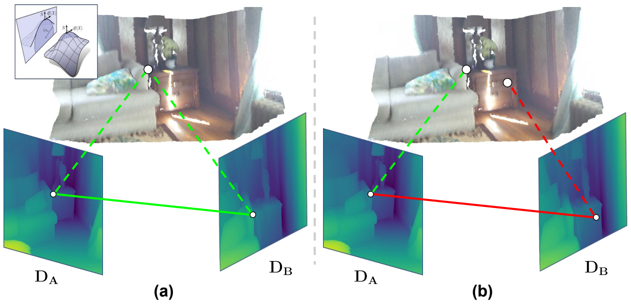


Figure 1. **Curvature-guided match selection:** (a) Correct matches in two views must share the same underlying surface curvature, *e.g*., predicted from monocular depth. (b) Correspondences from different surfaces are guaranteed to be incorrect.

Recent advances in deep learning-based feature matching have made significant progress in addressing the limitations of hand-crafted approaches, such as by jointly training detectors and descriptors [69, 12, 15, 44, 63, 40] with convolutional neural networks (CNNs), or combining hand-crafted and learning-based descriptors in a hybrid manner [41, 6]. SuperGlue [49] introduced the use of transformer networks to learn the matching process and formulate the problem as an optimal transport task [64]. LoFTR [55] and its recent variants [58, 8, 65] leverage both the global and local context from raw images by jointly learning the feature extraction and matching in a single network. While these approaches have led to state-of-the-art performance on several benchmarks, they work entirely in the 2D image domain and ignore the underlying 3D geometry of the scene. As feature matching is, essentially, finding the corresponding projections of an actual 3D point, this could be a critical limitation in real scenarios where the

*Part of the work was done during the author's visit to ETH zurich.

[1] Code and trained models are available at https://github.com/AaltoVision/surface-curvature-estimator.

matched pixels from different views must share the geometry of the underlying surface patch, as shown in Fig. 1.

In this paper, we focus on investigating and exploiting 3D geometry cues, *e.g*. coming from monocular depth predictions, for feature matching. The only prior work is [61], which adopts depth priors to find planar image regions that are then rectified to eliminate viewpoint changes. Finally, a handcrafted detector and descriptor are applied to the rectified image regions. This, however, relies on the heavy assumption that the images consist mostly of dominant planes, which severely restricts its out-of-the-box applicability. In this work, we go further by exploring the local surface geometry coming from depth priors. This additional geometric clue is coupled with *any* off-the-shelf feature matcher increasing its accuracy. Benefiting from advanced monocular depth predictors [42, 43, 16], obtaining depth priors is easy and costs only a few milliseconds. We use these depth predictions to extract curvature at the observed feature points. Exploiting the fact that the curvature is invariant to viewpoint changes (*i.e*., rotation, translation, and scaling), we enforce that matched features must lie on similar surfaces. Our contributions are as follows:

1. We propose an approach for feature matching, leveraging dense depth via utilizing local surface curvature similarity, which is invariant to viewpoint changes. This approach is a departure from traditional matching methods that rely solely on image information.

2. The proposed curvature similarity extractor can be seamlessly integrated with any recent feature matcher, making it a versatile tool that can be easily adopted. The experiments show that it improves several recent algorithms on various benchmarks by $1-3\%$.

3. The proposed algorithm can be used to train feature matchers and monocular depth prediction networks jointly in an *end-to-end* fashion. To demonstrate this, we fine-tune the state-of-the-art MiDaS [42] depth predictor to increase the feature matching accuracy.

2. Related work

2.1. Local Feature Matching

Detector-based Matchers build sparse correspondences on top of the detected keypoints. Therefore, robust keypoint detection and feature extraction are essential for these approaches. Traditional hand-crafted descriptors, such as SIFT [32] and SURF [7], follow a *detect-then-describe* pipeline and have shown great success since the 2000s. More recently, learned detectors and descriptors [6, 69, 12, 15, 44, 63, 60, 30] with convolutional neural networks show their superiority in matching images with large viewpoint and appearance changes. This traditional pipeline is further developed to *detect-and-describe* [12, 15, 44] and *describe-to-detect* [60, 30] strategies leveraging high-level image information for accurate and reliable matching in challenging conditions. Besides substituting the traditional detectors and descriptors with their learned counterparts for image matching, the recent SuperGlue [49] focuses on the matching stage itself. It replaces the naive mutual nearest neighbour search with Graph Neural Networks (GNN). SuperGlue takes the sparse descriptors and their positional encoding as inputs, then leverage a transformer-based [64] network to create a more robust feature representation for the optimal partial assignment. The follow-up work [54] further improves the performance and efficiency by adaptively clustering sparse features into different subgraphs and using a coarse-to-fine paradigm. Although showing significant improvements with the recent components, the detector-based matchers are naturally limited by the accuracy of detected keypoints coming from independent detectors.

Detector-free Matchers directly estimate the dense correspondences from raw images without an independent keypoint detection stage. These approaches leverage low and high-level image information with neural networks and can distinguish indistinctive regions across images. The pioneer work of [46] constructs 4D cost volumes, significantly increasing the matching accuracy. However, it is computationally expensive as the complexity is $\mathcal{O}(n^2)$, where n is the number of pixels or patches. [45, 26] alleviate the problem by applying either sparse convolutions or a coarse-to-fine paradigm. More recently, transformer-based detector-free matchers [55, 58, 65, 8, 24, 57] have received attention due to their strong performance on local feature matching. The most representative work, LoFTR [55], inherits the advantages of graph matching from SuperGlue and leverages a linear transformer [64] for efficient dense matching. [58, 8] utilise more efficient transformer structures to improve the performance. Detector-free matchers show a promising direction towards local feature matching by encoding rich image information. We claim that extracting 3D geometric information could further enhance accuracy.

Correspondence Pruning approaches [17, 4, 5, 72, 70, 56] apply a consensus mechanism to filter out the outliers from putative correspondences coming from the feature matcher procedure. RANSAC [17] and its follow-up works [4, 5] are the most popular correspondence pruning algorithms. In the era of deep learning, OANet [72] infers the probabilities of correspondences being inliers with an order-aware network. Other works improve the accuracy of correspondence pruning by applying motion coherence constraints [31, 33], leveraging local-to-global consensus learning procedure [74, 11], and adopting the attention mechanism [56]. These approaches provide inlier probability predictions for each putative correspondence that can be used either to filter or order matches. Finally, least squares fitting or traditional robust estimation is applied to the matches.

2.2. 3D Geometric Priors to Image Matching

Incorporating 3D geometric priors into various vision tasks has been widely explored in recent years. [23] leverages the RGB-D reconstructions to learn view-invariant, geometry-aware 2D representations for downstream tasks. [71] introduces the predicted monocular depth to guide the optimization of neural scene representation. The surface curvature, a popular 3D geometric cue, is widely applied to 3D vision tasks such as 3D point cloud registration [62], multi-view stereo [68, 19]. However, integrating 3D geometric cues into 2D image matching has received little attention. [37, 18] assume that the underlying surface patch is locally planar for feature detection and description, while [61] uses prior depth information to improve feature extraction by rectifying large planar regions. None of the above approaches considers the 3D geometric information in the local region, and we are the first to include 3D surface geometry in general 2D feature matching.

3. Feature Matching with Surface Curvature

This section proposes an approach to improve any local feature matcher by extracting information about the underlying local 3D surface predicted by a deep network. The overview of our method coupled with LoFTR [55] is shown in Fig. 2. Although the proposed algorithm is compatible with any matcher, we demonstrate its effectiveness with LoFTR. Combining it with other ones is straightforward.

3.1. LoFTR-style Matchers

This section provides a brief overview of LoFTR. Given an image pair $I_A, I_B \in \mathbb{R}^{H*W*3}$, the end-to-end trainable matcher, first, extracts coarse-level features $\mathcal{F}_A^c, \mathcal{F}_B^c$ and, then, fine-level ones $\mathcal{F}_A^f, \mathcal{F}_B^f$ with a local feature CNN [28]. At the coarse level transformer module, the features are flattened and processed by a transformer-based architecture with multiple self- and cross-attention layers [64]. A matching confidence map $\mathcal{P}_c$ is predicted at the end of this module, and the matches $\mathcal{M}_c$ with high confidence are selected from $\mathcal{P}_c$ to fine-tune the prediction of fine-level features matching at the coarse-to-fine transformer module. The final outputs are the fine-level matches $\mathcal{M}_f$.

3.2. Curvature Similarity Extractor

In this section, we will discuss multiple potential ways of extracting local surface curvature at detected features. This curvature is then used for improving feature matching.

3.2.1 Curvature Similarity

For an at least two times continuously differentiable surface $\mathcal{S} \in \mathbb{C}^2$ in the 3D space, given a point $p \in \mathbb{R}^3$ and a direction $d \in \mathbb{R}^3$ on $\mathcal{S}$; the normal curvature $k_n \in \mathbb{R}$ measures how curved the surface is at this point along the direction [13]. Since there are infinitely many directions that travel through a point on the surface, there are also infinitely many normal curvatures at a given point. In differential geometry, the minimum normal curvature k_1 and the maximum curvature k_2 at point p are defined as the principal curvatures, where $k_1 \leq k_2$, and the directions of the principal curvatures are the smoothest and steepest directions on the surface. We also define the mean curvature H and Gaussian curvature K as

$$H = \frac{1}{2}(k_1 + k_2), \quad K = k_1 k_2 \tag{1}$$

A highly beneficial property of such curvatures is their invariance to surface rotation and translation. Both are important when describing local surface patches centered on features detected in 2D images. However, these measures do not preserve scale invariance [13], which is essential in our case due to using predicted relative depth maps instead of a metric one. This means that, in each image, the depth is only defined up-to-scale.

To seek the scale invariance of the curvatures for local patch matching, we follow [47] for the idea of scale-invariant curvature measure and define the curvature similarity function $S : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ as follows:

$$S(k_1, k_2) = \frac{\min(|k_1|, |k_2|)}{\max(|k_1|, |k_2|)}, \quad (0 \leq S(k) \leq 1), \tag{2}$$

where k_1 and k_2 are the minimum and maximum curvatures, respectively. It is easy to see that $S(k_1, k_2)$ is rotation, translation and scale invariant. Case $S(k_1, k_2) = 1$ is interpreted as the curvature of a point on a spherical surface. In the case when $k_1 = k_2 = 0$, we define $S(k_1, k_2)$ to be 0. Detecting points on planes is also straightforward from k_1 and k_2 as they both will become ∞. Proof of the invariances is provided in the supplementary material. Next, we will discuss ways of obtaining curvature similarities.

3.2.2 Surface Curvatures Extraction

This section introduces our approximate form of principal curvature estimation. Different from prior work [68, 19] that extract the curvatures from image intensity, we estimate the principal curvatures based on the depth map. This 3D geometric cue could be obtained either from the active sensors (LiDAR, Kinect) or monocular depth predictors [42, 43]. Since our target curvature similarity $S(k_1, k_2)$ is scaling invariant, the scaling ambiguity in monocular depth estimation is neglected and the predicted relative depth is sufficient for the curvature extraction.

In this work, we consider only a pair of RGB images as input and leverage any off-the-shelf depth predictor, *e.g.*

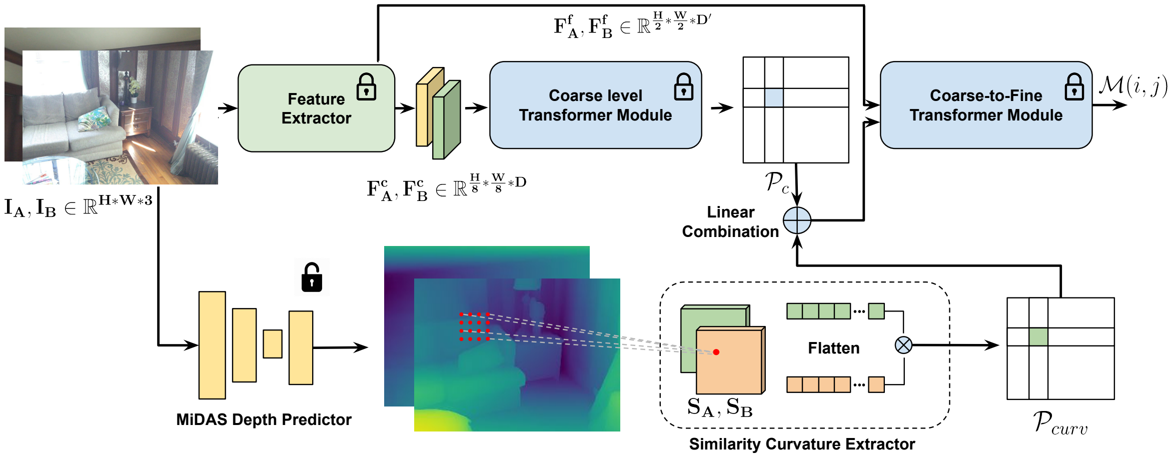

$\mathbf{F}^{\mathrm{f}}_{\mathrm{A}}, \mathbf{F}^{\mathrm{f}}_{\mathrm{B}} \in \mathbb{R}^{\frac{H}{2}*\frac{W}{2}*D'}$
Feature Extractor
Coarse level Transformer Module
Coarse-to-Fine Transformer Module
$\mathcal{M}(i,j)$
$\mathbf{I}_{\mathrm{A}}, \mathbf{I}_{\mathrm{B}} \in \mathbb{R}^{\mathrm{H}*\mathrm{W}*3}$
$\mathbf{F}^{\mathrm{c}}_{\mathrm{A}}, \mathbf{F}^{\mathrm{c}}_{\mathrm{B}} \in \mathbb{R}^{\frac{H}{8}*\frac{W}{8}*D}$
$\mathcal{P}_c$
Linear Combination
MiDAS Depth Predictor
Flatten
$\mathrm{S}_{\mathrm{A}}, \mathrm{S}_{\mathrm{B}}$
Similarity Curvature Extractor
$\mathcal{P}_{curv}$

Figure 2. **Overview of the CSE with LoFTR.** For a pair of images, it follows the LoFTR [55] matching pipeline to extract the coarse and fine level image features and generate a coarse matching confidence map $\mathcal{P}_c$. In the meantime, the images are fed to a monocular depth network to predict the depth map and extract the curvature similarity upon the depth map. We formulate a new curvature score matrix $\mathcal{P}_{curv}$ based on the pixel-wise $L2$ distance of the curvature similarities between two images and mix it with $\mathcal{P}_c$. The mixed score matrix is then processed by the coarse-to-fine transformer module together with the fine-level features to predict the final matches $\mathcal{M}_f$.

MiDaS [42], to extract the depth map $Z_i : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ for each input image, $i \in \{1, 2\}$. Let $\mathbf{r}_i(u, v, Z_i(u, v))$ be the 3D surface patch corresponding to the image region at coordinates (u, v) with window size $\delta \in \mathbb{R}^+$ in the ith image, where $Z_i(u, v)$ is the depth of point (u, v).

Let us formulate a local point set $\mathcal{P}_i = \{(u, v, Z_i(u, v)) \mid (u, v) \in \mathcal{M}^i_f\} \subseteq \mathbb{R}^3$, where $\mathcal{M}^i_f$ are the points detected in the ith image.

Principal Curvature via Surface Fitting. The most straightforward approach is first applying the quadratic surface approximation method [20] to estimate the surface $\mathbf{r}_i(u, v, Z_i(u, v))$ at all points in $\mathcal{P}$, then employing the first and second fundamental forms of a surface. The normal curvature is calculated as follows:

$$k_n = \frac{II}{I} = \frac{Ldu^2 + 2Mdudv + Ndv^2}{Edu^2 + 2Fdudv + Gdv^2}, \tag{3}$$

where $L = \mathbf{r}_{uu}\mathbf{n}$, $M = \mathbf{r}_{uv}\mathbf{n}$, $N = \mathbf{r}_{vv}\mathbf{n}$, $E = \mathbf{r}_u\mathbf{r}_u$, $F = \mathbf{r}_u\mathbf{r}_v$, $G = \mathbf{r}_v\mathbf{r}_v$; $\mathbf{r}_u, \mathbf{r}_v, \mathbf{r}_{uu}, \mathbf{r}_{uv}, \mathbf{r}_{vv}$ are the first and second-order derivatives of the surface patch $\mathbf{r}_i(u, v, Z_i(u, v))$, and $\mathbf{n} \in \mathbb{R}^3$ is the normal.

$$k_n = \frac{L + 2M\lambda + N\lambda^2}{E + 2F\lambda + G\lambda^2}. \tag{4}$$

Once we have the normal curvature k_n, we then calculate the partial derivative of k_n w.r.t. λ to obtain the principal curvatures k_1, k_2. The details are presented in the supplementary. While this approach works accurately and efficiently in practice, we observed that its gradients become unstable during back-propagation. Therefore, we explore another way as well for calculating the surface curvature.

Curvature Extraction via Quadrics. Another lightweight procedure for estimating the curvature is via fitting quadric surfaces, *e.g.* ellipsoid, locally to the neighborhood of the observed point. Thus, instead of directly estimating a quadratic surface from the given depth map, we constrain the surface to be a quadric and, more specifically, an ellipsoid. An ellipsoid has the beneficial property that all its curvatures are semi-positive. Its principal axes define three directions which we can use the define the similarity. We use the radii $\alpha, \beta, \gamma \in \mathbb{R}^+$ along the three axes and define the curvature similarity score as follows:

$$S(\alpha, \beta, \gamma) = \frac{\min(\alpha^2, \beta^2, \gamma^2)}{\max(\alpha^2, \beta^2, \gamma^2)}. \tag{5}$$

Similarity S is rotation, translation, scale and thus, viewpoint invariant. The advantage from Eq. (5) is that the computation of first and second-order gradients are no longer required for the curvature similarity calculation.

The general constraint that a quadric imposes on a point $x \in \mathcal{P}_i$ lying on the surface is

$$\mathbf{x}\mathbf{Q}_i\mathbf{x}^{\mathrm{T}} + \mathbf{P}_i\mathbf{x}^{\mathrm{T}} + \mathbf{R}_i = 0,$$

where $\mathbf{Q}_i$, $\mathbf{P}_i$, and $\mathbf{R}_i$ are the parameters in a matrix form. This can be written as

$$\begin{aligned} &ax^2 + by^2 + cz^2 + 2dyz + 2exz + \\ &2fyz + 2gx + 2hy + 2iz + j = 0. \end{aligned} \tag{6}$$

As described in [25], a sufficient condition that guarantees the quadric surface to be an ellipsoid is

$$4M - N^2 = 1,$$

where

$$N = a + b + c,$$
$$M = ab + bc + ac - d^2 - e^2 - f^2.$$

Also, since we aim at measuring the curvature of the surface patch located on the observed point, we have to constrain the ellipsoid so that the observed point $\mathbf{p}$ is on its surface. To do so, we translate the point set $\mathcal{P}$ as $\mathcal{P}' = \{\mathbf{q} - \mathbf{p} \mid \mathbf{q} \in \mathcal{P}\}$ and fix $j = 0$. The central point now is $\mathbf{p}' = [0, 0, 0]^{\mathrm{T}}$.

Substituting all these constraints into Eq. (6), we obtain

$$\begin{aligned} ax^2 + by^2 + cz^2 + 2dyz + 2exz + \\ 2fyz + 2gx + 2hy + 2iz = 0. \end{aligned} \tag{7}$$

From Eq. (7), only nine variables need to be estimated in our case. For each point $[u, v, z]^{\mathrm{T}} \in \mathcal{P}'$, we define $\mathbf{X}_i$ as

$$\mathbf{X}_i = [u_i^2, v_i^2, z_i^2, 2v_iz_i, 2u_iz_i, 2u_iv_i, 2u_i, 2v_i, 2z_i]^{\mathrm{T}}, \tag{8}$$

and coefficient matrix $\mathbf{C} = [\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, ..., \mathbf{X}_n]$. Then, we follow the method in [25] for the ellipsoid fitting and estimate its radii $R = (\alpha, \beta, \gamma)$. The algorithms for fitting and calculating the radii are shown in the supplementary.

3.3. Curvature-Guided Match Selection

The proposed curvature similarity extractor (CSE) is a depth-based plug-in component that can be combined with both detector-based [32, 49] and detector-free [55, 58, 8] matchers. For an image pair (I_A, I_B), the monocular depth predictor first estimates the depth map for each image independently. CSE extracts the curvature map (S_A, S_B) on the top of the depth maps. Note that LoFTR selects the candidate matches M_c at the coarse level of 1/8-resolution grids where the ground truth matches are defined based on the reprojection distance of the centers of the grid cells. Thus, it is not necessary to extract the curvature similarity at each pixel. It is calculated only at the center point of each cell, *i.e.*, $S_A, S_B \in \mathbb{R}^{(h*w)*1}$, where $h = H/8, w = W/8$.

Since the proposed curvature similarity is rotation, translation and scale invariant, S_A at point $\mathbf{p}_A$ in the first image should be equal to the similarity S_B of its corresponding pair $\mathbf{p}_B$ in the second image. The correct matches M_c, ideally, have the same curvature values. To measure this, we define the normalised curvature score matrix as

$$\mathcal{P}_{\text{curv}} = 1 - L_2(S(k)_A, S(k)_B), \tag{9}$$

where $L_2(\cdot, \cdot)$ is the L2-norm of $S(k)_A$ and $S(k)_B$. Since both $S(k)_A$ and $S(k)_B$ are in-between zero and one, the curvature score is also normalized into range $[0, 1]$.

The new matching score matrix is the linear combination of $\mathcal{P}_{\text{curv}}$ and LoFTR matching probability $\mathcal{P}_c$ as follows:

$$\mathcal{P} = \lambda \mathcal{P}_c + (1 - \lambda)\mathcal{P}_{\text{curv}}, \tag{10}$$

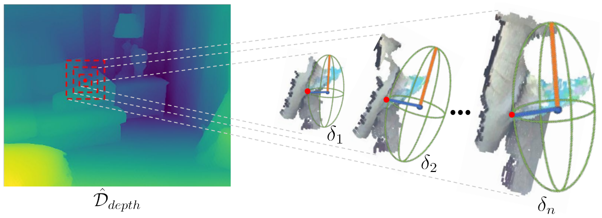


Figure 3. **Multi-scale curvature extraction.** We extract the curvature similarity from ellipsoids centered on the observed point, fitted with different window (*i.e.*, 3D neighborhood) sizes.

where λ is the mixing parameter for balancing the two matching scores. Since our $\mathcal{P}_{\text{curv}}$ is an additional supporting score for coarse-level matching selection, we set the parameter λ to 0.9 in all our experiments. We adopt the scheme of LoFTR for the coarse-level match selection as

$$M_c = \{(i, j) \mid \forall (i, j) \in \text{MNN}(\mathcal{P}), \mathcal{P}(i, j) \geq \theta\}, \tag{11}$$

where MNN is the mutual nearest neighbor operation, θ is a threshold so that we only select the matches with confidence higher than θ. Note that $\mathcal{P}_{\text{curv}}$ can be similarly combined with the matching score matrix of SuperGlue [49].

Multi-scale Curvature Similarity. Although the proposed curvature similarity is scaling invariant in theory, it still depends on the neighboring pixels of the points. Thus, the multi-scale extraction of this geometric cue could also be adopted for robustness to scale change in the images. For detector-based approaches, such as SuperGlue [49], we use a set of window size $\{\delta_1, \delta_2, ..., \delta_n\}$ with the window centered on detected keypoints. Hence, the curvature similarity maps are $S(k)_A \in \mathbb{R}^{\mathcal{A}*n}$ and $S(k)_B \in \mathbb{R}^{\mathcal{B}*n}$, $\mathcal{A}$ and $\mathcal{B}$ are the detected keypoints at I_A and I_B, respectively. For detector-free matchers, the multi-scale extraction is conducted within the patch grid. Fig. 3 describes the curvature similarity extraction at multiple scales.

3.4. Depth Prediction Fine-tuning

Since the proposed curvature similarity extraction is fully differentiable, it is possible to use it to train feature matching and depth prediction networks jointly end-to-end. To demonstrate this, we fine-tune the depth predictor with the curvature similarity extractor so that it takes the feature matching into account when being trained. The new confidence matrix $\mathcal{P}$ is minimized with the negative log-likelihood, similarly as in [49, 55], where the loss is

$$L = -\frac{1}{|M_c^{gt}|} \sum_{(i,j) \in M_c^{gt}} \log \mathcal{P}(i, j), \tag{12}$$

and M_c^{gt} is the ground truth matches defined at the coarse-level in [55]. Since the loss term focuses only on the matched patches, long training time would crash the depth prediction model. Thus, we fine-tune the depth predictor with only a few epochs and small learning rates.

Methods	YFCC100M [59]					MegaDepth [27]					ScanNet [9]				
	AUC@5°	@10°	@20°	P(%)	MS(%)	AUC@5°	@10°	@20°	P(%)	MS(%)	AUC@5°	@10°	@20°	P(%)	MS(%)
SuperGlue [49]	38.9	59.4	75.7	**98.7**	23.6	43.7	62.2	77.0	**99.6**	32.5	16.1	33.8	51.8	**84.4**	31.5
SuperGlue + CSE	**40.0**	**60.1**	**76.3**	95.6	**26.5**	**46.2**	**64.8**	**78.7**	98.3	**35.7**	**16.2**	**34.0**	**52.2**	80.7	**31.7**
LoFTR [55]	43.4	63.1	78.0	**95.2**	8.2	53.7	70.1	**82.4**	**96.8**	9.0	22.1	40.8	57.6	**87.4**	7.9
LoFTR + CSE	43.5	63.3	**78.1**	93.4	**8.5**	**54.4**	**70.4**	82.1	95.5	**9.2**	22.9	42.4	60.0	86.4	9.3
LoFTR + CSE (w/ FT)	**43.9**	**63.5**	**78.1**	93.3	**8.5**	54.3	**70.4**	82.2	95.5	**9.3**	**23.9**	**43.5**	**60.5**	86.1	**9.5**
QuadTree [58]	44.1	63.5	78.3	**96.5**	8.9	53.5	70.2	82.2	**98.5**	9.6	24.9	44.7	61.6	**89.4**	9.8
QuadTree + CSE	**45.0**	64.3	78.8	95.3	**9.1**	54.0	70.3	82.1	97.9	**9.8**	25.2	45.5	62.5	89.4	11.2
QuadTree + CSE (w/ FT)	**45.0**	**64.4**	**78.9**	95.5	**9.1**	**54.5**	**71.1**	**82.9**	98.0	9.7	**25.8**	**46.1**	**63.4**	89.2	**11.5**

Table 1. **Results of relative pose estimation on YFCC100M, MegaDepth, and ScanNet datasets.** We report AUC score of the translation and rotation errors with different thresholds (5°, 10°, 20°), precision (%), and matching score (%). Our methods and the best performance are marked in bold. **w/FT** indicates the results with fine-tuning. Note that comparison among baselines is not straightforward as some sensitive default parameters, *e.g*. input image size, are different.

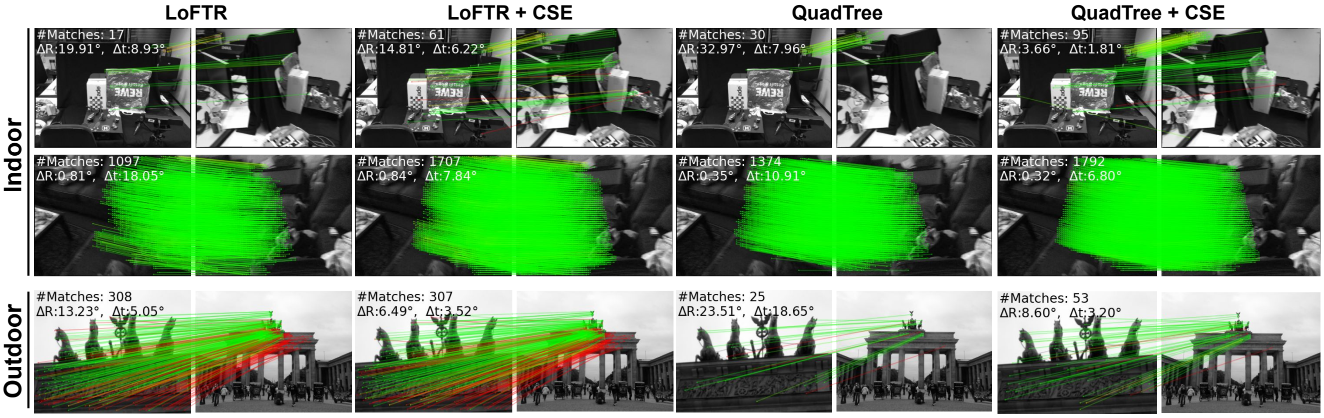


Figure 4. **Qualitative matching results**. We add the curvature similarity extractor module to LoFTR [55] and QuadTree [58] and compare the two approaches in both indoor and outdoor scenes. By adding the CSE module, we are able to obtain more correct matches (green lines) and better pose accuracy while keeping similar matching precision. The matches are coloured by the epipolar error threshold in [55].

4. Experiments

We first demonstrate the effectiveness of the proposed similarity curvature extractor on relative pose estimation and the downstream visual localization tasks. Next, we analyze the choice of architecture and hyper-parameters in the ablation study. Finally, we provide qualitative results to help understand the similarity curvature extractor.

Implementation Details. Our proposed CSE adopts MiDAS + ViT [42] as the default depth predictor. For detector-based approaches [55, 58], we extract the curvature similarity at keypoint locations and use multi-scales $\{7*7, 9*9, 11*11\}$. For detector-free matchers, we use the default grid patch size $8*8$ for curvature estimation.

For the depth predictor training, following the default setting in LoFTR [55], we fine-tune the predictor with the outdoor matching model on MegaDepth [27]. The model is optimized using Adam with a fixed learning rate of $r = 1e-5$ and batch size of 2. The indoor model is fine-tuned on ScanNet [9] with $r = 5e-6$. In the training process, the weights from the matcher are frozen, and only the weights from refineNets in MiDAS + ViT are updated. Instead of sampling 200 pairs from each scene, we reduce it to 10 pairs. The full training and evaluation process of the outdoor model is conducted on 2 Telsa-V100 GPUs with 32GB of memory. We train and evaluate the indoor model on 2 RTX 2080 Ti GPUs with a memory of 12GB.

4.1. Relative Pose Estimation

Datasets. We evaluate our CSE on three popular datasets, ScanNet [9], MegaDepth [27], and YFCC100M [59], for the two-view pose estimation task. ScanNet is an indoor RGB-D dataset consisting of 1,613 sequences with 2.5 million views, and each view is associated with a ground truth camera pose and depth map. Similar to the setting in LoFTR [49], the images and depth maps are resized to 640*480 for training and evaluation. MegaDepth and YFCC100M are two outdoor datasets which consist of multiple scenarios. We fine-tune the outdoor model on MegaDepth with the longest dimension of the images resized to 840 and evaluate our method on MegaDepth-1500 from [55] and YFCC100M-4000 from [49].

Baselines. The proposed CSE is a plug-in module that can be applied with both detector-based and detector-free matchers. We add this module to detector-based SuperGlue with SuperPoint [12] features, detector-free LoFTR, and its recent variant QuadTree [58]. The experiments follow the default setting from the original papers for all the baselines. For all the experiments on ScanNet, the image resolution is consistently 640×480. For evaluation on MegaDepth and YFCC100M, the default input sizes are different among the approaches. Thus, it is not reasonable to compare among the baselines as the input image size significantly impacts the final accuracy. We detail the default input sizes of different approaches in the supplementary material.

Metrics. Similar to the previous methods [49, 55], we report the Area Under the recall Curve (AUC) of the translation and rotation errors with multiple thresholds (5°, 10°, 20°). For each setting, we consider a pose to be correct if the maximum error from translation and rotation error is below the threshold. The estimated camera poses are recovered from the essential matrix estimated by RANSAC [17] with the inlier-outlier threshold set at 0.5 pixels. Besides, we report the matching precision and score [69, 12]. To calculate matching scores on detector-free matchers, we consider the number of coarse-level features as the number of detected keypoints, *i.e.*, $h * w$.

Results. We report the results of the CSE add-on on SuperGlue, LoFTR, and QuadTree in Tab. 1. For LoFTR and QuadTree, we report the results of directly adding the CSE module and adding CSE with fine-tuning. As the official training code for SuperGlue is not publicly available, we evaluated the SuperGlue + CSE without fine-tuning. However, we believe that fine-tuning would further improve the results of SuperGlue as well. We also notice that the MegaDepth-1500 is a part of the training set for SuperGlue. However, this does not invalidate the fact that the proposed method helps in finding better features. The proposed CSE improves the pose AUC scores of all methods on almost all datasets. Fine-tuning further increases the accuracy, and QuadTree combined with the proposed CSE method and fine-tuned on depth predictions achieves state-of-the-art AUC scores on *all* tested datasets.

As for matching score and precision, we also visualize the matching results of LoFTR and QuadTree in Fig. 4. By adding CSE, we obtain more correct matches and better pose accuracy on surface regions with distinctive geometry, while keeping similar matching precision.

4.2. Visual Localization

Datasets and Metrics. We select the AACHEN DAY-NIGHT v1.1 dataset [52] to demonstrate the effectiveness of our proposed approach on the visual localization task. AACHEN DAY-NIGHT is a challenging urban-scale outdoor dataset which is collected by handheld devices with 6,697 reference images and 1,015 queries, including 191 nighttime images. The ground truth 6DoF camera pose is obtained by COLMAP [53] and refined by [73]. We follow the evaluation protocols of the visual localization benchmark [52] reporting the translation and rotation error recalls at 0.25m/2°, 0.5m/5°, and 5m/10°.

Methods	**Day**	**Night**
	(0.25m, 2°) / (0.5m, 5°) / (5m,10°)	
SP [12] + SG [49]	89.8 / 96.1 / **99.4**	77.0 / 90.6 / **100.0**
SP + SG + CSE	**90.7 / 96.5 / 99.4**	75.9 / **91.1 / 100.0**

Table 2. **Visual localization.** We report the pose recall at (0.25m/2°, 0.5m/5°, 5m/10°) on the Aachen Day-Night [52] dataset with the HLoc [48] algorithm. The best results are bold.

Results. We add our CSE to the SuperPoint [12] + SuperGlue [49] pipeline in HLoc [48, 49] without fine-tuning any models. The matching pairs are provided by HLoc with top-50 candidate images from NetVLAD [2]. Tab. 2 reports the pose error recalls. We observe that the plug-in CSE improves the SP + SG pipeline on all thresholds on the Day sequences. On the Night sequence, it improves at threshold (0.5m/5°) and (5m/10°). We believe the slight decrease at (0.25m/1°) stems from the inaccuracies of depth prediction on nighttime images. Overall, by adding the CSE module, the localization accuracy is improved.

4.3. Ablation Study

We analyse the impact of different design components on our curvature similarity extractor in this section. We evaluate two main design choices: (1) multi-scale curvature extraction, (2) choice of depth predictors, and (3) mixing parameter value λ selection. All experiments are conducted with the Quadtree [58] matcher.

Multi-Scale	**ScanNet [9]**			**MegaDepth [27]**			**Time (ms)**
	AUC@5°/ AUC@10°/ AUC@20°						
4*4	24.4	44.6	61.9	52.7	69.6	82.0	113
6*6	25.3	**45.6**	**62.8**	53.3	69.9	81.8	115
8*8 (default)	25.2	45.5	62.5	**54.0**	**70.3**	**82.1**	**112**
4*4+6*6+8*8	**25.6**	45.2	62.4	**54.0**	70.0	**82.1**	338

Table 3. **Multi-scale and different grid sizes.** The AUC scores and average CSE run-times (ms) of Quadtree [58] + CSE with different grid sizes and on multi-scale estimation.

Multi-scale. As described in Sec. 3.3, we conduct multi-scale curvature similarity extraction on a $8 * 8$ patch grid with QuadTree by default. In Tab. 3, we show the results

Depth Predictors	**CSE**	**FT**	**AUC@5**	**@10**	**@20**	**Model**
Ground Truth	Quadric	✗	25.1	45.0	62.0	**153 MB**
MiDAS (Res101)	Quadric	✓	25.5	45.5	62.5	575 MB
MiDAS (Real-time)	Quadric	✓	25.5	45.4	62.9	239 MB
MiDAS (ViT)	Quadric	✓	**25.8**	**46.1**	**63.4**	1.6 GB
MiDAS (ViT)	Surface Fit	✗	24.8	44.7	61.6	1.6 GB

Table 4. **Depth study.** The AUC scores of QuadTree [58] with the proposed CSE using the GT depth or MiDaS [42] with different models. Quadric and surface fitting results are also shown.

λ (AUC@5°)	1.0	0.9	0.8	0.7	0.5	0.3
ScanNet [9]	24.9	25.8	**26.1**	23.8	21.0	17.8
MegaDepth [27]	53.5	**54.5**	52.8	47.0	45.5	39.8

Table 5. **Mixing Parameter Value Selection.** We finetune the Quadtree + SCE with different λ values and report the AUC@5°on ScanNet and MegaDepth datasets.

when using $4*4$, $6*6$, and multi-scale grids without finetuning. Even though the proposed CSE is robust to the grid size and the accuracy is similar, it slightly increases together with the grid size. Size $4*4$ leads to the worst solutions on average, while $8*8$ leads to the best ones. As expected, curvature similarity extraction with multi-scale shows the best performance. However, it increases the average run-time by three times.

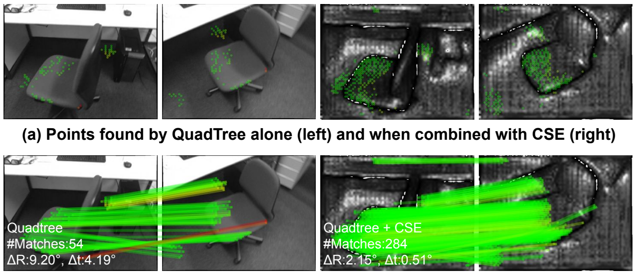
Depth Predictors and Curvature Extractors. We then explore the impact of different depth predictors on the final pose accuracy and model size. We select the ground truth depth, MiDAS with ResNet101 [22] architecture and MiDAS small (real-time) to compare with the default predictor MiDAS with ViT [64]. Tab. 4 reports the pose accuracy of these approaches on the ScanNet dataset. MiDAS with ViT, which shows the best performance, has the largest model size due to the heavy depth prediction network. When we employ lighter models for depth prediction, we observe that the performance has only a small drop, indicating that our method is still effective when working with lightweight networks. We also notice that the pose accuracy with ground truth depth is worse than the fine-tuned models. We believe the reason for this is the missing depth regions in the ground truth. We visualize the depth maps in different conditions at supplementary material.

The last two rows of the table show the results with the proposed ellipse and with quadratic surface fitting. The ellipse fitting performs more accurately in the experiments.

Mixing Parameter Value. We finetune our model with different values for λ in Eq. (10). The results are reported in Tab. 5. The proposed model performs best with $\lambda = 0.9$ for outdoor and $\lambda = 0.8$ for indoor scenes. This is expected since the proposed CSE is additional guidance for coarse-level matching that helps reduce the score of incorrect matches stemming from different surfaces. Assigning a large mixing weight could degrade the performance.

4.4. Curvature Similarity Understanding

In Fig. 5, we visualize how the curvature similarity score impacts the key points selection and matching results. We first show in the curvature images that similar surfaces share similar curvature values (more visualizations are presented in the supplementary). Then, we observe that, after adding CSE, more matches are detected from the same surface, as we increase the matching confidence of those matches by assigning a larger curvature similarity score. Besides, the curvature score helps in lowering the matching confidence of matches from different surfaces, *e.g*., the incorrect (red) matches in Fig. 5 (b) left are eliminated in the right.


(a) Points found by QuadTree alone (left) and when combined with CSE (right)

(b) Matches found by QuadTree alone (left) and when combined with CSE (right)

Figure 5. Keypoints (top row) and matches (bottom) found by the QuadTree matcher [58] with and without the proposed CSE, overlayed on the RGB (left) and extracted curvature images (right).

5. Conclusions

We present a curvature similarity extractor (CSE), a new algorithm to leverage 3D geometric cues in local feature matching. By fitting quadrics to depth maps obtained from off-the-shelf monocular depth predictors, we extract curvature similarity, which is invariant to translation, rotation, and scaling. CSE can be seamlessly integrated with matchers to guide the match selection. Also, it allows training feature matches and monocular depth networks jointly minimizing the matching loss in an end-to-end manner to further improve the accuracy. Our experiments in Sec. 4 demonstrate the effectiveness of the proposed CSE module, which consistently improves the pose AUC scores of all methods on almost all datasets, achieving state-of-the-art performance on all benchmarks when combined with QuadTree.

Limitations and Future Work. First, the integration of depth prediction and curvature similarity extraction into feature matching leads to increased memory consumption and inference time. An alternative solution for time-sensitive applications is to use a lightweight depth predictor. As shown in Tab. 4, such an approach still increases the accuracy. Second, the discussed invariance properties do not hold at depth discontinuities, and limitations may arise in scenes lacking accurate depth or clear surfaces. Even though this did not pose an issue in our experiments, this is a potential future direction that might lead to further improvements.

Acknowledgements. This work is supported by the Academy of Finland (grant No. 327911, No. 353138). Daniel Barath is supported by ETH Postdoc fellowship. We acknowledge the computational resources provided by the Aalto Science-IT project and CSC-IT Center for Science, Finland.

References

[1] Pablo Fernández Alcantarilla, Adrien Bartoli, and Andrew J Davison. Kaze features. In *Proceedings of European Conference on Computer Vision*, pages 214–227, 2012. 1

[2] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5297–5307, 2016. 7

[3] Relja Arandjelović and Andrew Zisserman. Three things everyone should know to improve object retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2911–2918, 2012. 1

[4] Daniel Barath and Jiří Matas. Graph-cut ransac. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6733–6741, 2018. 2

[5] Daniel Barath, Jana Noskova, Maksym Ivashechkin, and Jiri Matas. Magsac++, a fast, reliable and accurate robust estimator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1304–1312, 2020. 2

[6] Axel Barroso-Laguna, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Key. net: Keypoint detection by handcrafted and learned cnn filters. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5836–5844, 2019. 1, 2

[7] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European Conference on Computer Vision*, pages 404–417. Springer, 2006. 1, 2

[8] Hongkai Chen, Zixin Luo, Lei Zhou, Yurun Tian, Mingmin Zhen, Tian Fang, David Mckinnon, Yanghai Tsin, and Long Quan. Aspanformer: Detector-free image matching with adaptive span transformer. In *Proceedings of the European Conference on Computer Vision*, 2022. 1, 2, 5

[9] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017. 6, 7, 8

[10] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics*, 36(4):1, 2017. 1

[11] Luanyuan Dai, Yizhang Liu, Jiayi Ma, Lifang Wei, Taotao Lai, Changcai Yang, and Riqing Chen. Ms2dg-net: Progressive correspondence learning via multiple sparse semantics dynamic graph. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8973–8982, 2022. 2

[12] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshop*, pages 224–236, 2018. 1, 2, 7

[13] M.P. do Carmo. *Differential Geometry of Curves and Surfaces: Revised and Updated Second Edition*. Dover Books on Mathematics. Dover Publications, 2016. 3

[14] Siyan Dong, Shuzhe Wang, Yixin Zhuang, Juho Kannala, Marc Pollefeys, and Baoquan Chen. Visual localization via few-shot scene region classification. *International Conference on 3D Vision*, 2022. 1

[15] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable cnn for joint description and detection of local features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8092–8101, 2019. 1, 2

[16] Ainaz Eftekhar, Alexander Sax, Jitendra Malik, and Amir Zamir. Omnidata: A scalable pipeline for making multitask mid-level vision datasets from 3d scans. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10786–10796, 2021. 2

[17] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 2, 7

[18] Per-Erik Forssén and David G Lowe. Shape descriptors for maximally stable extremal regions. In *International Conference on Computer Vision*, pages 1–8. IEEE, 2007. 3

[19] Khang Truong Giang, Soohwan Song, and Sungho Jo. Curvature-guided dynamic scale networks for multi-view stereo. In *International Conference on Learning Representations*, 2022. 3

[20] Jack Goldfeather and Victoria Interrante. A novel cubic-order algorithm for approximating principal direction vectors. *ACM Trans. Graph.*, 23(1):45–63, 2004. 4

[21] Chris Harris, Mike Stephens, et al. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988. 1

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 8

[23] Ji Hou, Saining Xie, Benjamin Graham, Angela Dai, and Matthias Nießner. Pri3d: Can 3d priors help 2d representation learning? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5693–5702, 2021. 3

[24] Wei Jiang, Eduard Trulls, Jan Hosang, Andrea Tagliasacchi, and Kwang Moo Yi. Cotr: Correspondence transformer for matching across images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6207–6217, 2021. 2

[25] Qingde Li and J.G. Griffiths. Least squares ellipsoid specific fitting. In *Geometric Modeling and Processing, 2004. Proceedings*, pages 335–340, 2004. 4, 5

[26] Xinghui Li, Kai Han, Shuda Li, and Victor Prisacariu. Dual-resolution correspondence networks. *Advances in Neural Information Processing Systems*, 33:17346–17357, 2020. 2

[27] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2041–2050, 2018. 6, 7, 8

[28] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017. 3

[29] Philipp Lindenberger, Paul-Edouard Sarlin, Viktor Larsson, and Marc Pollefeys. Pixel-perfect structure-from-motion with featuremetric refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5987–5997, 2021. 1

[30] Xiaotao Liu, Chen Meng, Fei-Peng Tian, and Wei Feng. Dgd-net: Local descriptor guided keypoint detection network. In *2021 IEEE International Conference on Multimedia and Expo*, pages 1–6. IEEE, 2021. 2

[31] Yuan Liu, Lingjie Liu, Cheng Lin, Zhen Dong, and Wenping Wang. Learnable motion coherence for correspondence pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3237–3246, 2021. 2

[32] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 1, 2, 5

[33] Jiayi Ma, Ji Zhao, Junjun Jiang, Huabing Zhou, and Xiaojie Guo. Locality preserving matching. *International Journal of Computer Vision*, 127(5):512–531, 2019. 2

[34] Jiri Matas, Ondrej Chum, Martin Urban, and Tomás Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 22(10):761–767, 2004. 1

[35] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004. 1

[36] Krystian Mikolajczyk, Tinne Tuytelaars, Cordelia Schmid, Andrew Zisserman, Jiri Matas, Frederik Schaffalitzky, Timor Kadir, and L Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1):43–72, 2005. 1

[37] Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Repeatability is not enough: Learning affine regions via discriminability. In *Proceedings of the European Conference on Computer Vision*, pages 284–300, 2018. 3

[38] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015. 1

[39] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017. 1

[40] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3456–3465, 2017. 1

[41] Rémi Pautrat, Viktor Larsson, Martin R Oswald, and Marc Pollefeys. Online invariance selection for local feature descriptors. In *European Conference on Computer Vision*, pages 707–724. Springer, 2020. 1

[42] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12179–12188, 2021. 2, 3, 4, 6, 8

[43] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3), 2022. 2, 3

[44] Jerome Revaud, Cesar De Souza, Martin Humenberger, and Philippe Weinzaepfel. R2d2: Reliable and repeatable detector and descriptor. *Advances in Neural Information Processing Systems*, 32, 2019. 1, 2

[45] Ignacio Rocco, Relja Arandjelović, and Josef Sivic. Efficient neighbourhood consensus networks via submanifold sparse convolutions. In *European Conference on Computer Vision*, pages 605–621. Springer, 2020. 2

[46] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Neighbourhood consensus networks. *Advances in Neural Information Processing Systems*, 31, 2018. 2

[47] John Rugis and Reinhard Klette. A scale invariant surface curvature estimator. In *Pacific-Rim Symposium on Image and Video Technology*, pages 138–147. Springer, 2006. 3

[48] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12716–12725, 2019. 1, 7

[49] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4938–4947, 2020. 1, 2, 5, 6, 7

[50] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Improving image-based localization by active correspondence search. In *European conference on computer vision*, pages 752–765. Springer, 2012. 1

[51] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1744–1756, 2017. 1

[52] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, et al. Benchmarking 6dof outdoor visual localization in changing conditions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8601–8610, 2018. 1, 7

[53] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016. 1, 7

[54] Yan Shi, Jun-Xiong Cai, Yoli Shavit, Tai-Jiang Mu, Wensen Feng, and Kai Zhang. Clustergnn: Cluster-based coarse-to-fine graph neural network for efficient feature matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12517–12526, 2022. 2

[55] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 1, 2, 3, 4, 5, 6, 7

[56] Weiwei Sun, Wei Jiang, Eduard Trulls, Andrea Tagliasacchi, and Kwang Moo Yi. Acne: Attentive context normalization for robust permutation-equivariant learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11286–11295, 2020. 2

[57] Dongli Tan, Jiang-Jiang Liu, Xingyu Chen, Chao Chen, Ruixin Zhang, Yunhang Shen, Shouhong Ding, and Rongrong Ji. Eco-tr: Efficient correspondences finding via coarse-to-fine refinement. *Proceedings of the European Conference on Computer Vision*, 2022. 2

[58] Shitao Tang, Jiahui Zhang, Siyu Zhu, and Ping Tan. Quadtree attention for vision transformers. In *International Conference on Learning Representations*, 2022. 1, 2, 5, 6, 7, 8

[59] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016. 6

[60] Yurun Tian, Vassileios Balntas, Tony Ng, Axel Barroso-Laguna, Yiannis Demiris, and Krystian Mikolajczyk. D2d: Keypoint extraction with describe to detect approach. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 2

[61] Carl Toft, Daniyar Turmukhambetov, Torsten Sattler, Fredrik Kahl, and Gabriel J Brostow. Single-image depth prediction makes feature matching easier. In *European Conference on Computer Vision*, pages 473–492. Springer, 2020. 2, 3

[62] Lijing Tong and Xiang Ying. 3d point cloud initial registration using surface curvature and surf matching. *3D Research*, 9:1–16, 2018. 3

[63] Michał Tyszkiewicz, Pascal Fua, and Eduard Trulls. Disk: Learning local features with policy gradient. *Advances in Neural Information Processing Systems*, 33:14254–14265, 2020. 1, 2

[64] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017. 1, 2, 3, 8

[65] Qing Wang, Jiaming Zhang, Kailun Yang, Kunyu Peng, and Rainer Stiefelhagen. Matchformer: Interleaving attention in transformers for feature matching. In *Proceedings of the Asian Conference on Computer Vision*, 2022. 1, 2

[66] Shuzhe Wang, Zakaria Laskar, Iaroslav Melekhov, Xiaotian Li, and Juho Kannala. Continual learning for image-based camera localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3252–3262, 2021. 1

[67] Changchang Wu. Towards linear-time incremental structure from motion. In *International Conference on 3D Vision*, pages 127–134. IEEE, 2013. 1

[68] Zhenyu Xu, Yiguang Liu, Xuelei Shi, Ying Wang, and Yunan Zheng. Marmvs: Matching ambiguity reduced multiple view stereo for efficient large scale scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2020. 3

[69] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *European Conference on Computer Vision*, pages 467–483. Springer, 2016. 1, 2, 7

[70] Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to find good correspondences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2666–2674, 2018. 2

[71] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *Advances in Neural Information Processing Systems*, 2022. 3

[72] Jiahui Zhang, Dawei Sun, Zixin Luo, Anbang Yao, Lei Zhou, Tianwei Shen, Yurong Chen, Long Quan, and Hongen Liao. Learning two-view correspondences and geometry using order-aware network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5845–5854, 2019. 2

[73] Zichao Zhang, Torsten Sattler, and Davide Scaramuzza. Reference pose generation for long-term visual localization via learned features and view synthesis. *International Journal of Computer Vision*, 129(4):821–844, 2021. 7

[74] Chen Zhao, Yixiao Ge, Feng Zhu, Rui Zhao, Hongsheng Li, and Mathieu Salzmann. Progressive correspondence pruning by consensus learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6464–6473, 2021. 2