EDM: Efficient Deep Feature Matching

Xi Li Tong Rao Cihui Pan
Realsee
{lixi042, raotong001, pancihui001}@realsee.com

Abstract

Recent feature matching methods have achieved remarkable performance but lack efficiency consideration. In this paper, we revisit the mainstream detector-free matching pipeline and improve all its stages considering both accuracy and efficiency. We propose an Efficient Deep feature Matching network, EDM. We first adopt a deeper CNN with fewer dimensions to extract multi-level features. Then we present a Correlation Injection Module that conducts feature transformation on high-level deep features, and progressively injects feature correlations from global to local for efficient multi-scale feature aggregation, improving both speed and performance. In the refinement stage, a novel lightweight bidirectional axis-based regression head is designed to directly predict subpixel-level correspondences from latent features, avoiding the significant computational cost of explicitly locating keypoints on high-resolution local feature heatmaps. Moreover, effective selection strategies are introduced to enhance matching accuracy. Extensive experiments show that our EDM achieves competitive matching accuracy on various benchmarks and exhibits excellent efficiency, offering valuable best practices for real-world applications. The code is available at https://github.com/chicleee/EDM.

1. Introduction

Image feature matching is a crucial task in the field of computer vision with a broad range of important applications, including structure from motion (SfM)[1, 21, 52], simultaneous localization and mapping (SLAM) [6, 36], visual tracking [53, 66], and visual localization [48, 50], etc. Traditional feature matching methods generally consist of several stages, including keypoint detection, feature description and matching[4, 5, 33, 47].

Benefiting from the powerful feature description capability of deep neural networks, many recent studies [3, 11, 42, 67] have adopted convolutional neural networks to extract local features, which significantly outperform the conventional handcrafted features. Besides, feature matching methods [32, 49] based on deep learning have also emerged and achieved remarkable results. Although these methods are generally effective, they still encounter difficulties due to various challenging factors, including illumination variations, scale changes, poor textures, and repetitive patterns.

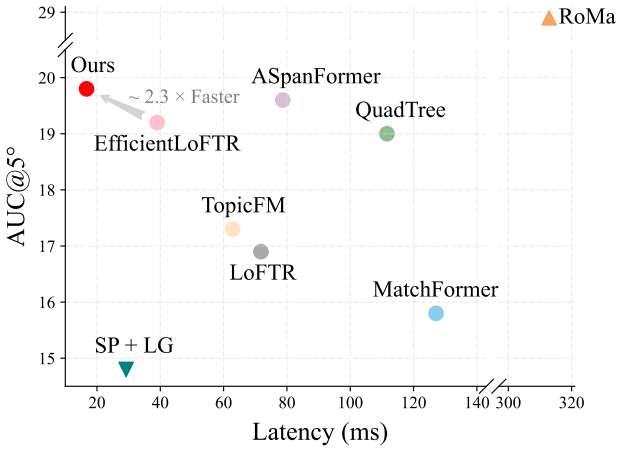


Figure 1. **Comparison of Matching Accuracy and Latency.** Our method achieves competitive accuracy with lower latency. Models are evaluated on the ScanNet dataset to get AUC@5° accuracy, while the latency for an image pair with 640×480 resolution is measured on a single NVIDIA 3090 GPU.

To address these limitations, end-to-end detector-free local feature matching methods are coming into existence. Early methods [17, 28, 43, 44, 68] typically used the cost volume and neighborhood consensus to generate matches. Given the powerful capability of modelling long-range global context information, some studies [8, 57, 63] have started using Transformer [61] to establish precise correspondences. To reduce computational complexity, most of these methods usually adopt a coarse-to-fine scheme. Specifically, coarse matches at the patch level are first obtained using the nearest neighbor criterion, then refined to the sub-pixel level for increased accuracy. More recently, some studies [15, 16] have explored methods for generating dense, pixel-wise matching, achieving impressive performance on mainstream datasets.

Although previous methods have constantly achieved breakthroughs in matching accuracy, few studies have focused on the ease of deployment and inference efficiency, which limits their application in real-time programs. Local feature matching is considered as a low-level computer vision task. Consequently, the current mainstream methods prioritize high-resolution local features for accurate matching, and their networks are designed to be typically shallow and wide, resulting in limited utilization of global high-level contextual information. While high-resolution local features offer superior localization accuracy and intuitive understanding, they come at a significant computational cost. A key insight is that focusing excessively on local details is computationally burdensome and superfluous.

In this work, we introduce EDM, an innovative and efficient deep feature matching network. By extracting high-level feature correlations between two images at deeper layers and implicitly estimating precise fine matches, EDM strikes an optimal balance between efficiency and performance. Fig. 1 highlights the impressive results of EDM.

In summary, the contributions of this paper include:

- A new detector-free matcher significantly improves efficiency while maintaining competitive accuracy by redesigning all the steps of the mainstream paradigm.
- A Correlation Injection Module models deep features correlations with high-level context information and integrates global and local features by hierarchical correlation injection to enhance performance and efficiency.
- A novel lightweight bidirectional axis-based regression head for estimating subpixel-level matches implicitly.
- Efficient matching selection strategies are proposed to improve accuracy for both coarse and fine stages.

2. Related Work

2.1. Feature Matching

Sparse Matching. Sparse matching methods are also known as detector-based methods. Classical methods utilize handcrafted keypoint detection, feature description and matching [4, 5, 33, 47]. Recently, learning-based keypoint detection [3, 46] , description [19, 41, 64, 67] and matching methods [7, 24, 26, 54] leverage the powerful expressive capabilities of deep neural networks to enhance their robustness and performance. Notably, SuperPoint (SP) [11] introduces a self-supervised network for both detection and description by leveraging homographic adaptation. Numerous subsequent methods [14, 35, 42, 60] follow this paradigm. SuperGlue (SG) [49] is the first to introduce the self- and cross-attention [61] to capture keypoint feature correlations, resulting in improved matching accuracy. To improve efficiency, LightGlue (LG) [32] finds that the computationally complex attention process can end earlier for most easy image pairs. For sparse methods, detecting repeatable keypoints is still challenging, particularly in low-texture areas.

Dense Matching. Dense matching methods aim to estimate all matchable pixel-level correspondences. Earlier methods NCNet [43] and its subsequent works [28, 44] achieved end-to-end dense matching by using 4D cost volume to represent features and possible matches. More recently, DKM [15] models the dense matches as probability functions with the Gaussian process and achieves impressive results. Similarly, RoMa [16] is a dense matcher that leverages a frozen pre-trained DINOv2 [40] model for extracting coarse features and a specialized VGG [55] model for further refinement. Dense matching methods exhibit significant matching capabilities, but they tend to be slower in practical applications due to excessive computational overhead.

Semi-Dense Matching. Semi-dense matchers[17, 68] adopt a coarse-to-fine manner, which not only fully utilizes the entire image space, but also avoids overly dense pixel-level calculations. Benefiting from the long-range modeling capability of the Transformer [61], LoFTR [57] and its follow-ups [8, 59, 62] apply the Transformer to enhance local features. TopicFM [18] attempts to model high-level contexts and latent semantic information as topics in deeper layer features, but it still uses the heavy fine-level network of LoFTR [57]. EfficientLoFTR (ELoFTR) [63] introduces an aggregated attention network to reduce local feature tokens for efficient transformation and a correlation refinement module for fine-level correspondences location in high-resolution features, achieves comparable performance with lower latency. ETO [38] introduces multiple homography hypotheses for local feature matching, achieves comparable efficiency but displays a performance gap.

2.2. Keypoints Estimation in Related Tasks

Keypoints estimation is an important component of feature matching and also plays a significant role in various other computer vision tasks, such as object detection [13, 69], human pose estimation [27, 29, 34, 39], hand and facial keypoints detection [9, 25], etc. DSNT [39] introduces the soft-argmax method to calculate the approximate maximum response point from the feature maps, enabling the model to directly regress the coordinate values. RLE [27] proposes an effective regression paradigm, namely residual log-likelihood estimation, which improves regression performance by utilizing normalized flows [12] to estimate latent distributions and facilitate the training process. SimCC [29] divides each pixel into several bins and classifies the coordinates of each region to achieve subpixel-level positioning accuracy. We design our fine matching network based on these methods to avoid the heavy computational burden of upsampling and high-resolution heatmaps.

3. Methods

An overview of our pipeline is shown in Fig. 2.

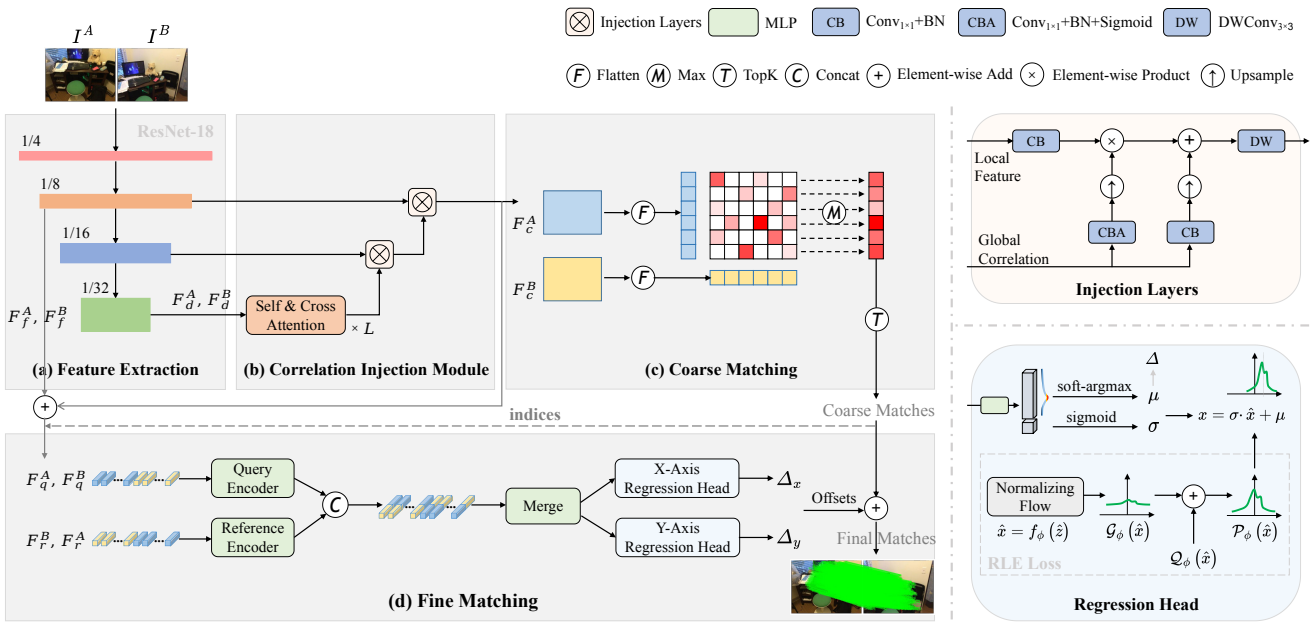


Figure 2. **Pipeline Overview**. **(a)** A deeper CNN backbone is adopted to extract multi-level feature maps. **(b)** In the Correlation Injection Module, we alternately apply self-attention and cross-attention a total of L times to capture and transform the correlations between deep feature F_d^A and F_d^B. Subsequently, two Injection Layers are employed to progressively inject feature correlations from deep to local levels. **(c)** After the CIM, the coarse features F_c^A and F_c^B are flattened and then correlated to produce the similarity matrix. To establish coarse matches, we determine the row-wise maxima in the probability matrix and select the top K values among them. **(d)** For fine-level matching, the corresponding fine features are extracted by the indices obtained from the coarse matching process. We treat the fine features F_q^A and F_q^B as queries, while considering the same features but in reversed order, F_r^B and F_r^A, as references. The query and reference features are encoded separately and then merged together. Then, a lightweight regression head is designed to estimate the reference offsets on the X and Y axes, respectively. The final matches are obtained by adding the coarse matches to their corresponding offsets.

3.1. Feature Extraction

Unlike previous detector-free methods [57, 63] using a shallow-wide CNN to extract features at $\frac{1}{8}$ scale of the original image resolution for feature transformation and coarse-level matching, and then employing the Feature Pyramid Network (FPN) [31] to upsample the features to $\frac{1}{2}$ or a higher scale for fine-level matching, our feature extractor is a ResNet-18 [20] with fewer channels and deeper layers. In order to achieve higher efficiency and capture more comprehensive high-level context information such as semantics and geometries, we utilize low-resolution deep feature maps F_d^A and F_d^B at $\frac{1}{32}$ scale for feature transformation and F_f^A and F_f^B at $\frac{1}{8}$ scale for fine matching regression.

3.2. Correlation Injection Module

Inspired by [31, 65], the Correlation Injection Module (CIM) is introduced to aggregate the multi-scale features before coarse matching. The CIM is composed of stacked Transformers and two Injection Layers (ILs) as a whole.

Feature Transform. The deep feature maps F_d^A and F_d^B at $\frac{1}{32}$ scale are transformed by interleaving self- and cross-attention L times to obtain the correlations between the features of two images. This design significantly reduces the token sequence length and computational overhead in the Transformer. Following [63], the 2D rotary positional embedding (RoPE) [56] is inserted to each self-attention layers to capture the relative spatial information.

Query-Key Normalized Attention. Attention mechanism is a core component in the Transformer, characterized by query Q, key K, and value V. The attention weight, determined by Q and K, results in an output that is a weighted sum of V. To enhance the correlation modeling capability, we replace the vanilla attention [61] with Query-Key Normalized Attention (QKNA) [22], which is defined as:

$$QKNormAtt(Q, K, V) = softmax(s \cdot \hat{Q}\hat{K}^T)V \quad (1)$$

where s is a manual scale factor, $\hat{Q}$ and $\hat{K}$ are obtained by applying L2 normalization in the head dimensions.

Injection Layers. After modeling feature correlations, two cascaded Injection Layers (ILs) are used to upsample features to $\frac{1}{8}$ scale. As illustrated in the top-right of the Fig. 2, the ILs take the backbone local features and the deep features containing global correlations as inputs. The local features are passed through a 1×1 convolution layer and a batch normalization layer in sequence (CB) to increase the

number of channels to match the global features. The low-resolution deep features, which have a larger receptive field and contain global correlations and rich context information, are first fed into the 1×1 convolution, batch normalization and a sigmoid activation function (CBA) to generate weights to determine how much detail to retain for the local features. Then, the output is upsampled to match the size of the local features and injected into the local features by element-wise product. Meanwhile, the global features are passed through another CB block and bilinear interpolation upsampling, and then element-wise added to the features after injection. Additionally, a 3×3 depthwise convolution (DW) [23] is used to alleviate the aliasing effect of upsampling. Finally, after two consecutive ILs, the multi-scale features from two views are efficiently aggregated, and coarse features F_c^A and F_c^B for the subsequent matching process are obtained.

3.3. Coarse Matching

We establish coarse-level matches based on the coarse feature maps F_c^A and F_c^B after correlation injection. Each pixel on the feature maps F_c^A and F_c^B represents an 8×8 grid region in original images. So coarse matches indicate rough local window correspondences between two images. Firstly, the coarse feature maps F_c^A and F_c^B are flattened to 1-D vectors $\tilde{F}_c^A$ and $\tilde{F}_c^B$. Then we utilize the inner product to build a similarity matrix $\mathcal{S}$ as follows:

$$\mathcal{S} = \frac{\left\langle \tilde{F}_c^A(i), \tilde{F}_c^B(j) \right\rangle}{\tau} \tag{2}$$

where τ means the temperature parameter.

Following [57], the matching probability matrix $\mathcal{P}_c$ is obtained by a dual-softmax [43] operator on both dimensions of $\mathcal{S}$:

$$\mathcal{P}_c = softmax(\mathcal{S}(i,\cdots))_j \cdot softmax(\mathcal{S}(\cdots,j))_i \tag{3}$$

Efficient Implementation. We note that the above Eq. (3) can also be implemented by first calculating the exponential function as $\mathcal{Z} = e^{\mathcal{S}}$ only once, and then taking the product of its row-wise and column-wise L1 normalizations, so as to reduce redundant computations and improve inference efficiency. This implementation can be defined as:

$$\mathcal{P}_c = \frac{\mathcal{Z}}{\|\mathcal{Z}(i,\cdots)_j\|_1} \cdot \frac{\mathcal{Z}}{\|\mathcal{Z}(\cdots,j)_i\|_1} \tag{4}$$

Coarse Matching Selection. Contrary to the previous methods of selecting matches using Mutual Nearest Neighbor (MNN), we first obtain the maximum values from each row of the probability matrix $\mathcal{P}_c$, and then select the Top-K scoring values to control the number of coarse matches. Besides, the selected matching probabilities should be higher than the coarse-level threshold θ_c. Such a matching selection strategy drastically reduces the time complexity, and the elimination of dynamic tensor shapes facilitates the formation of mini-batches for efficient inference.

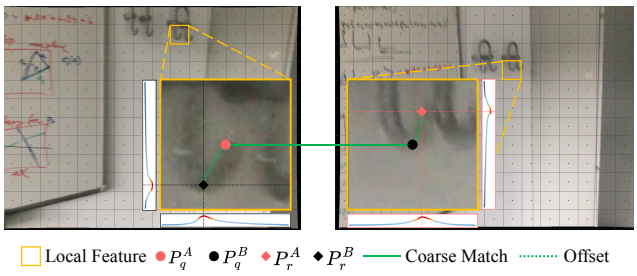


Figure 3. **Bidirectional Refinement.** For a coarse matching pair, the center point of one grid serves as query for fine matching, and its corresponding reference point is offset from the center point in another grid, exhibiting duality.

3.4. Fine Matching

For higher efficiency, we regress fine-level matching offsets directly from latent features, abandoning explicit pixel-level keypoint localization from high-resolution features. Firstly, we take the element-wise sum of backbone features F_f^A , F_f^B and coarse-level features F_c^A , F_c^B as inputs. Then we extract fine-level corresponding features using coarse matching indices and flatten them to 1-D vectors F^A , F^B.

Bidirectional Refinement. We consider the central pixel of grids as keypoints P^A, P^B with descriptors F^A , F^B. As show in Fig. 3, we define the grid center P_q^A as the query point, and its coarse corresponding keypoint on the reference image grid is P_q^B. Due to quantization errors during supervision, for the query points P_q^A, there is an offset between the ground truth keypoint P_r^A and the coarse corresponding keypoint P_q^B. Similarly, we found that using point P_q^B as the query point is dual. So we propose a bidirectional refinement strategy to obtain double fine matches with a single slight inference. We concatinate F^A , F^B in sequence as query features F_q^A , F_q^B, and reference features F_r^B , F_r^A in the reverse order. Then, they are passed through their respective query and reference encoders, each consisting of a lightweight Multi-Layer Perceptron (MLP). Subsequently, the corresponding features are concatenated along the descriptive dimension and then merged through another MLP.

Axis-Based Regression Head. Inspired by [27, 29, 39], regressing numerical coordinates directly from latent features is extremely fast, yet it lacks spatial generalization and robustness. To facilitate model learning, we design a lightweight Axis-Based Regression Head (ABRHead) with Soft Coordinate Classification (SCC) as shown in the bottom-right of the Fig. 2. Taking the X-axis as an example, the merged feature first passed through linear layers to

Category	Method	MegaDepth			ScanNet			Time (ms)
		AUC@5°	AUC@10°	AUC@20°	AUC@5°	AUC@10°	AUC@20°	
Sparse	SP [11] + SG [49]	49.7	67.1	80.6	16.2	32.8	49.7	48.4
	SP [11] + LG [32]	49.9	67.0	80.1	14.8	30.8	47.5	29.2
Dense	DKM [15]	60.4	74.9	85.1	26.6	47.1	64.2	186.2
	ROMA [16]	62.6	76.7	86.3	28.9	50.4	68.3	312.9
Semi-Dense	LoFTR [57]	52.8	69.2	81.2	16.9	33.6	50.6	71.8
	QuadTree [59]	54.6	70.5	82.2	19.0	37.3	53.5	111.6
	MatchFormer [62]	53.3	69.7	81.8	15.8	32.0	48.0	127.1
	ASpanFormer [8]	55.3	71.5	83.1	19.6	**37.7**	**54.4**	78.7
	TopicFM [18]	54.1	70.1	81.6	17.3	35.5	50.9	62.9
	EfficientLoFTR [63]	56.4	72.2	83.5	19.2	37.0	53.6	39.0
	Ours	**57.5**	**73.2**	**84.2**	**19.8**	37.5	**54.4**	**16.7**

Table 1. **Results of Relative Pose Estimation on the MegaDepth Dataset and ScanNet Dataset.** The models are trained on the MegaDepth dataset to evaluate all methods on both datasets. The AUC of relative pose error at multiple thresholds, and the average inference time on the ScanNet dataset for pairwise image of 640×480 resolution is provided.

reduce the number of output dimension to N+1. The N-D tensor is passed through soft-argmax [39] to predict a location parameter μ, which indicates the index of the maximum response in continuous coordinate space from a classification view. The another 1-D tensor is passed through a sigmoid activation function to predict a scale parameter σ. The output μ and σ are used to shift and scale the distribution generated by a flow model [12], respectively. SCC, which utilizes N bins, implicitly encodes local coordinate information on the one hand, thereby reducing the learning difficulty. On the other hand, it avoids the issue of regression method values exceeding the local window boundary.

Besides, we use RLE Loss [27] to supervise the prediction results of the network (refer to Sec. 3.5). The predict μ is equivalent to the normalized offset Δ, which represents the distance from the predicted keypoint coordinate to the center of the grid along the current axis. Additionally, ϕ represents the parameters of the flow model, which is not required during the inference process, thus avoiding additional overhead during testing.

Fine Matching Selection. The scale parameter σ reflects the standard deviation of the predict distribution. The model will output a larger σ for a more uncertain result. Therefore, the prediction confidence can be obtained by:

$$\mathcal{P}_f = 1 - \frac{\sigma_x + \sigma_y}{2} \tag{5}$$

where σ_x and σ_x represent the σ on X- and Y-axis respectively. For each bidirectional matching pair, we keep the more confident one while requiring it to be above the fine-level threshold θ_f to enhance the matching precision.

3.5. Loss Function

Coarse-Level Loss Function. To generate the coarse-level ground truth matches $\mathcal{M}_c$, we warp the grid centroids from input image I^A to I^B using relative camera poses and depth maps at $\frac{1}{8}$ scale following previous works [49, 57, 63]. The matching probability matrix $\mathcal{P}_c$ produced by dual-softmax is supervised by minimizing the focal loss [45]:

$$\mathcal{L}_c = -\frac{1}{|\mathcal{M}_c|} \sum_{\langle i,j\rangle \in \mathcal{M}_c} \alpha \left(1 - \mathcal{P}_c\langle i,j\rangle\right)^{\gamma} \log \mathcal{P}_c\langle i,j\rangle \tag{6}$$

where α and γ are respectively defined as weighting factor and focusing parameter.

Fine-Level Loss Function. We employ the residual log-likelihood estimation (RLE) [27] loss to improve the offset regression performance, which can be defined as follows:

$$\mathcal{L}_f = -\log \mathcal{G}_\phi\left(\hat{x}\right) - \log \mathcal{Q}_\phi\left(\hat{x}\right) + \log \sigma \tag{7}$$

where $\mathcal{G}_\phi\left(\hat{x}\right)$ is the distribution learned by the normalizing flow model ϕ, $\mathcal{Q}_\phi\left(\hat{x}\right)$ is a simple Laplace distribution, and σ is the prediction scale parameter. Specifically, the Laplace distribution loss item about $\mathcal{Q}_\phi\left(\hat{x}\right)$ can be defined as:

$$\mathcal{Q}_\phi\left(\hat{x}\right) = \sum_{\mathcal{M}_f} \frac{1}{\sigma} e^{-\frac{|\mu^{gt} - \mu|}{2\sigma}} \tag{8}$$

where $\mathcal{M}_f$ is the ground truth fine-level matches, which is a subset of correctly predicted coarse-level matches $\tilde{\mathcal{M}}_c$. The μ^{gt} is the corresponding ground truth offsets.

The total loss is the weighted sum of coarse-level and fine-level matching loss as follows:

$$\mathcal{L} = \lambda_c \mathcal{L}_c + \lambda_f \mathcal{L}_f \tag{9}$$

3.6. Implementation Details

The backbone feature widths from $\frac{1}{2}$ scale to $\frac{1}{32}$ scale are [32, 64, 128, 256, 256]. We set L to 2 in the CIM to transform deep correlations. The coordinate bins number N in ABRHead is 16. The attention scale factor s is set to 20.

Following [63], to demonstrate the generalization ability of EDM, we trained it on the outdoor MegaDepth dataset and evaluated it on all tasks and datasets in our experiments. During the training phase, images are resized and padded to the size of 832×832. The training process utilizes the AdamW optimizer with an initial learning rate of $2e$-3 and a batch size of 32 on 8 NVIDIA 3090 GPU. The model converges in 6 hours, which is extremely fast compared to other methods. For loss function, the focal loss parameters α and γ are set to 0.25 and 2 respectively. Then we set λ_c to 1 for coarse-level loss weight and λ_f to 0.2 for fine-level loss weight. The coarse-level threshold θ_c is usually set to $5e$-2, while fine-level threshold θ_f is set to $1e$-6.

4. Experiments

4.1. Relative Pose Estimation

Datasets. We follow the test settings of the previous methods [49, 57, 63], selecting 1500 image pairs from the indoor ScanNet [10] dataset and outdoor MegaDepth [30] dataset, respectively. For ScanNet, we resize all images to 640×480 resolution. For MegaDepth, images are resized to 832×832 for training and 1152×1152 for validation.

Evaluation Protocol. Following SuperGlue (SG) [49] and LoFTR [57], the relative pose error is defined as the maximum of angular errors in rotation and translation. We report the area under the cumulative curve (AUC) of the relative pose error under multiple thresholds, including 5°, 10°, and 20°. In addition, the pairwise matching runtime on the ScanNet dataset is reported to explain the accuracy-efficiency tradeoffs. Specifically, a single NVIDIA 3090 GPU is used to measure the latency of all methods.

Results. As shown in Tab. 1, EDM shows superior performance compared with sparse and semi-dense methods on both datasets, with the exception of a slightly lower AUC@10° on the ScanNet dataset compared to ASpanFormer [8]. Specifically, our method surpasses the recent semi-dense baseline ELoFTR [63] on all metrics, with a significant speed improvement.

4.2. Homography Estimation

Dataset. We evaluate our method on the widely adopted HPatches dataset [2] for homography estimation.

Evaluation Protocol. Following [18, 49, 57], we resize input images to 480px in the smallest dimension and select the top 1000 matches. We compute the mean reprojection error for the four corners and report the AUC values under 3, 5, and 10-pixel thresholds. For fairness, we use the same OpenCV RANSAC with identical parameters to estimate homography for all comparative methods.

Results. As presented in Tab. 2, Our EDM notably outperforms other methods under all thresholds, demonstrating its effectiveness for homography estimation.

Category	Method	Homography est. AUC		
		@3px	@5px	@10px
Sparse	DISK [60] + NN	52.3	64.9	78.9
	SP [11] + SG [49]	53.9	68.3	81.7
	SP [11] + LG [32]	54.2	68.3	81.5
Semi-Dense	DRC-Net [28]	50.6	56.2	68.3
	Patch2Pix [68]	59.3	70.6	81.2
	LoFTR [57]	65.9	75.6	84.6
	TopicFM [18]	67.3	77.0	85.7
	ASpanFormer [8]	67.4	76.9	85.6
	EfficientLoFTR [63]	66.5	76.4	85.5
	Ours	**68.5**	**78.1**	**86.6**

Table 2. Homography estimation on HPatches.

Method	DUC1	DUC2
	(0.25m,2°) / (0.5m,5°) / (1.0m,10°)	
SP [11] + SG [49]	47.0 / 69.2 / 79.8	53.4 / 77.1 / 80.9
SP [11] + LG [32]	49.0 / 68.2 / 79.3	55.0 / 74.8 / 79.4
LoFTR [57]	47.5 / 72.2 / 84.8	54.2 / 74.8 / 85.5
TopicFM [18]	52.0 / **74.7** / **87.4**	53.4 / 74.8 / 83.2
ASpanFormer [8]	51.5 /73.7 / 86.0	55.0 / 74.0 / 81.7
PATS [37]	**55.6** / 71.2 / 81.0	58.8 / 80.9 / 85.5
EfficientLoFTR [63]	52.0 / **74.7** / 86.9	58.0 / 80.9 / **89.3**
Ours	51.5 / 72.7 / 85.9	**59.5** / **82.4** / 88.5

Table 3. Results of visual localization on InLoc dataset.

Method	Day	Night
	(0.25m,2°) / (0.5m,5°) / (1.0m,10°)	
SP [11] + SG [49]	89.8 / 96.1 / **99.4**	77.0 / 90.6 / **100.0**
SP [11] + LG [32]	**90.2** / 96.0 / **99.4**	77.0 / 91.1 / **100.0**
LoFTR [57]	88.7 / 95.6 / 99.0	**78.5** / 90.6 / 99.0
TopicFM [18]	**90.2** / 95.9 / 98.9	77.5 / 91.1 / 99.5
ASpanFormer [8]	89.4 / 95.6 / 99.0	77.5 / 91.6 / 99.5
PATS [37]	89.6 / 95.8 / 99.3	73.8 / **92.1** / 99.5
EfficientLoFTR [63]	89.6 / **96.2** / 99.0	77.0 / 91.1 / 99.5
Ours	89.1 / **96.2** / 98.8	77.0 / **92.1** / 99.5

Table 4. Results of visual localization on Aachen v1.1 dataset.

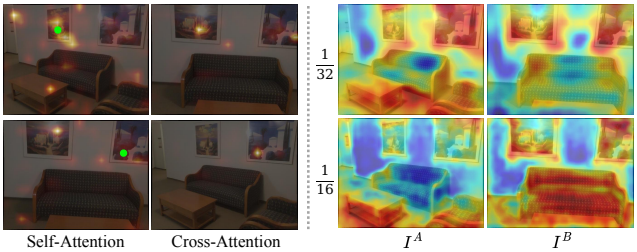
4.3. Visual Localization

Datasets and Evaluation Protocols. Following [49, 57], We assess our method on the InLoc [58] dataset and Aachen v1.1 [51] dataset for visual localization, within the open-sourced localization pipeline HLoc [48].

Results. As shown in Tab. 3 and Tab. 4, EDM performs comparably to sparse and semi-dense methods on the InLoc dataset and Aachen v1.1 dataset, demonstrating robust generalization in visual localization.

4.4. Understanding EDM

Weight Analysis. In the CIM, we employ self- and cross-attention alternately at deeper layers to capture feature correlations enriched with high-level context information, such



(a) Deep Correlations (b) Injection Weights

Figure 4. **Attention Visualization.** (a) Deep correlations.The green dots represent the query points. (b) Injection weights. Significant response values usually located in detail-rich regions.

as semantics and structures. To explain this process, we selected several query points and visualized the outcomes of self- and cross-attention separately. Specifically, we summed and normalized the weight maps corresponding to the same input image and the same type of attention, upsampled and overlaid them on the original image. As depicted in Fig. 4 (a), in the context of self-attention, the larger response points are more dispersed across different semantic regions. Conversely, in cross-attention, the significant response points are more concentrated in proximity to the potential matching points.

In the ILs after modeling feature correlations, the low-resolution global features, characterized by a larger receptive field and rich context information, are fed into a CBA block to generate weights that determine the level of detail retention for the local features. As shown in Fig. 4 (b), we overlay two layers of weight maps onto the input images. The weight maps at $\frac{1}{32}$ and $\frac{1}{16}$ scale layers exhibit different focuses, but the more prominent response values generally cluster in regions with distinct details.

Qualitative Results Visualization. As shown in 5. Our approach is able to extract more adequate and accurate matches compared to LoFTR [57] and ELoFTR [63], even in challenging scenes characterized by wide viewpoints, repetitive patterns and textureless regions. Previous methods primarily focused on low-level local features, often struggle with strong repetitive structures in indoor environments, like similar paintings or sofas. By leveraging CIM, EDM correlates high-level context information across views, thus enhancing matching capability.

Image Size Analysis. As shown in Tab. 5, we evaluate the performance and efficiency variations of our method and ELoFTR [63] across different image sizes. Employing a larger image size leads to an accuracy enhancement, albeit at the expense of a slower speed. Our method significantly outperforms ELoFTR [63] at all resolutions under both Mixed-Precision and FP32 configurations.

Stage Analysis. We evaluated the running time of each stage of our method on the ScanNet dataset at 640×480 resolution, and benchmarked it against the leading semi-dense matchers, LoFTR [57] and ELoFTR [63]. As presented in 6, our method achieves higher efficiency in all matching stages. Specifically, compared to ELoFTR [63], our method reduces the time consumption by 56.1% in feature extraction, 34.5% in feature transformation, 70.4% in coarse matching, and 76.6% in fine matching. Finally, in terms of overall time, it is 2.3 times faster than ELoFTR.

Resolution	Method	Pose Est. AUC @5°/@10°/@20°	Runtime (ms) Mixed-Precision / FP32
640×640	ELoFTR [63]	51.0/67.4/79.8	46.6 / 52.1
	Ours	52.2/68.9/80.9	**23.0 / 23.8 (-54.3%)**
800×800	ELoFTR [63]	53.4/70.0/81.9	63.0 / 75.7
	Ours	54.3/70.8/82.4	30.7 / 34.7 (-54.2%)
960×960	ELoFTR [63]	54.7/70.7/82.4	90.2 / 114.9
	Ours	55.6/71.4/82.8	44.9 / 52.8 (-54.0%)
1152×1152	ELoFTR [63]	56.4/72.2/83.5	142.1 / 185.0
	Ours	57.5/**73.2/84.2**	72.3 / 86.0 (-53.5%)
1408×1408	ELoFTR [63]	56.2/73.1/83.4	257.4 / 327.8
	Ours	**57.6/73.2**/84.1	136.4 / 162.7 (-50.4%)

Table 5. Comparison of image size on the MegaDepth dataset.

Stage	Runtime (ms) LoFTR [57]	ELoFTR [63]	Ours
(a) Feature Extraction	28.01	9.12	**4.00 (-56.1%)**
(b) Feature Transform	17.77	12.52	**8.20 (-34.5%)**
(c) Coarse Matching	7.80	7.71	**2.28 (-70.4%)**
(d) Fine Matching	18.23	9.67	**2.26 (-76.6%)**
Overall	71.81	39.02	**16.74 (-57.1%)**

Table 6. Runtime comparisons for each stage on ScanNet dataset.

Ablation Study. For a comprehensive understanding of our method, we conduct ablation studies at different stages on the MegaDepth dataset. The results are shown in Tab. 7. (a) Feature Extraction. (1) Following ELoFTR's shadow-wide network design results in decreased matching accuracy and a substantial increase in running time. (b) Feature Transform. (2) Adopting QKNA can improve evaluation metrics, especially for AUC@5°. (3-5) Setting $L = 2$ achieves an optimal balance between performance and efficiency. (6) Replacing ILs with a naive element-wise sum for multi-scale feature integration leads to a substantial performance drop. (c) Coarse Matching. (7) Our implementation of dual-softmax saves significant inference time compared to previous methods. (8) Compared to MNN, our coarse matching selection strategy offers higher efficiency and precision. (9) Focal loss improves performance compared to the negative log-likelihood loss in coarse matching supervised learning. (d) Fine Matching. (10) Replacing the entire stage with a high-resolution implementation of ELoFTR, leading to notable time overhead and a decline in performance. (11) Bidirectional refinement yields significant performance gains with only a minor increase in time cost. (12)

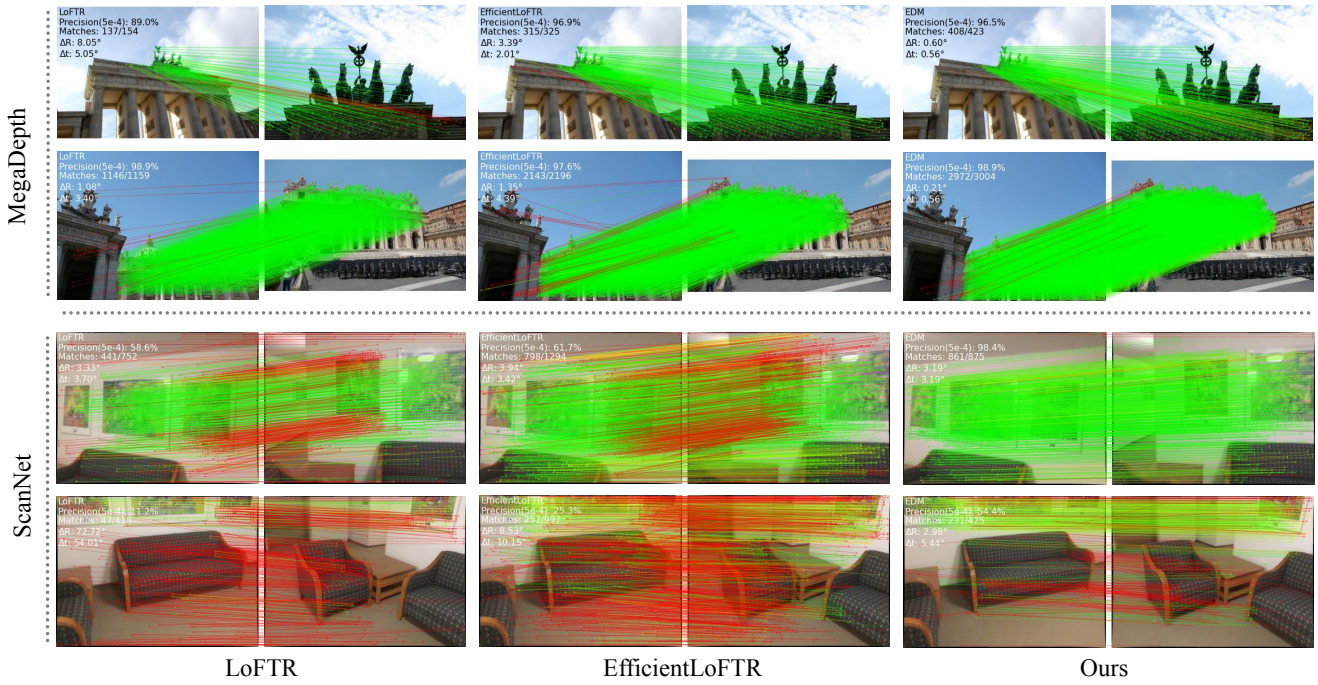


Figure 5. **Qualitative Comparisons.** Compared with LoFTR [57] and EfficientLoFTR [63], our method is more robust in scenarios with large viewpoint changes and repetitive semantics. The red color indicates epipolar error beyond $5e$-4 in the normalized image coordinates.

Method	Pose est. AUC @5°/@10°/@20°	Time(ms)
Ours Full	57.5/73.2/84.2	86.0
(a) Feature Extraction		
(1) shadow-wide design (to $\frac{1}{8}$ scale)	56.9/72.6/83.5	109.8
(b) Feature Transform (CIM)		
(2) replace QKNA with vanilla Attn.	56.7/72.9/84.0	85.7
(3) $L = 0$	51.8/67.3/78.8	70.1
(4) $L = 1$	55.9/71.5/82.9	77.5
(5) $L = 4$	57.6/73.2/84.1	101.9
(6) replace ILs with element-wise sum	55.9/71.7/82.9	85.0
(c) Coarse Matching		
(7) replace dual-softmax with ELoFTR's	57.3/73.0/84.0	100.6
(8) replace coarse selection with MNN	57.2/72.7/83.9	101.2
(9) negative log-likelihood loss	56.3/71.9/83.2	86.0
(d) Fine Matching		
(10) replace fine matching with ELoFTR's	56.2/71.9/83.0	113.8
(11) w/o bidirectional refinement	55.7/71.9/83.4	84.0
(12) w/o fine selection by σ	57.0/72.7/83.8	86.1
(13) replace $\mathcal{Q}_\phi(\hat{x})$ with Gussian dist.	57.1/72.9/84.0	86.0
(14) w/o Soft Coordinates Classification	56.5/72.2/83.4	85.2
(15) replace RLE loss with L1 loss	53.9/70.6/82.4	86.7
(16) replace RLE loss with L2 loss	53.6/70.1/82.2	86.7

Table 7. Ablation studies on the MegaDepth dataset at all steps, with average running times measured at 1152×1152 resolution.

Using σ to select bidirectional fine matches for retaining more confident results can innocuously boost matching accuracy. (13) Laplace distribution is a more suitable initial distribution for local feature matching than Gaussian distribution. (14) SCC simplifies fine matching local offset regression. (15-16) Compared to supervising regression results with L1 or L2 loss, the RLE loss significantly enhances regression accuracy without additional inference overhead.

Limitations. EDM's significant relative efficiency advantage declines moderately with increasing image resolution due to deeper feature extraction layers. However, semi-dense matchers generally achieve optimal performance without requiring extremely high resolutions.

5. Conclusions

Depart from the prevailing shallow-wide network design paradigm, this paper introduces EDM, an efficient deep feature matching network. By alternately applying self- and cross-attention on low-resolution deep layers to model feature correlations, and integrating global and local features through progressive correlation injection, the proposed CIM notably reduces the number of tokens while capturing enriched high-level contextual information, thereby enhancing both the matching accuracy and efficiency. Besides, we design a novel lightweight bidirectional axis-based regression head to implicitly refine the coarse matches by estimating local coordinate offsets. We also propose deployment-friendly matching selection strategies to filter accurate matches effectively at both coarse and fine matching stages. As a result, EDM attains competitive performance in multiple benchmarks with superb efficiency by redesigning all the steps of the mainstream semi-dense matching pipeline, opening up new prospects for time-sensitive image matching applications.

References

[1] Sameer Agarwal, Noah Snavely, Ian Simon, Steven M Seitz, and Richard Szeliski. Building rome in a day. In *ICCV*, 2009. 1

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

[3] Axel Barroso-Laguna, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Key. net: Keypoint detection by handcrafted and learned cnn filters. In *ICCV*, 2019. 1, 2

[4] Herbert Bay. Surf: Speeded up robust features. In *ECCV*, 2006. 1, 2

[5] Michael Calonder, Vincent Lepetit, Mustafa Ozuysal, Tomasz Trzcinski, Christoph Strecha, and Pascal Fua. Brief: Computing a local binary descriptor very fast. *PAMI*, pages 1281–1298, 2011. 1, 2

[6] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 2021. 1

[7] Hongkai Chen, Zixin Luo, Jiahui Zhang, Lei Zhou, Xuyang Bai, Zeyu Hu, Chiew-Lan Tai, and Long Quan. Learning to match features with seeded graph matching network. In *ICCV*, pages 6301–6310, 2021. 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 *ECCV*, 2022. 1, 2, 5, 6

[9] Xingyu Chen, Yufeng Liu, Yajiao Dong, Xiong Zhang, Chongyang Ma, Yanmin Xiong, Yuan Zhang, and Xiaoyan Guo. Mobrecon: Mobile-friendly hand mesh reconstruction from monocular image. In *CVPR*, 2022. 2

[10] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 6

[11] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPRW*, 2018. 1, 2, 5, 6

[12] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016. 2, 5

[13] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *ICCV*, pages 6569–6578, 2019. 2

[14] 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 *CVPR*, 2019. 2

[15] Johan Edstedt, Ioannis Athanasiadis, Mårten Wadenbäck, and Michael Felsberg. Dkm: Dense kernelized feature matching for geometry estimation. In *CVPR*, 2023. 1, 2, 5

[16] Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. Roma: Robust dense feature matching. In *CVPR*, 2024. 1, 2, 5

[17] Ufuk Efe, Kutalmis Gokalp Ince, and Aydin Alatan. Dfm: A performance baseline for deep feature matching. In *CVPR*, pages 4284–4293, 2021. 1, 2

[18] Khang Truong Giang, Soohwan Song, and Sungho Jo. Topicfm: Robust and interpretable topic-assisted feature matching. In *AAAI*, 2023. 2, 5, 6

[19] Pierre Gleize, Weiyao Wang, and Matt Feiszli. Silk: Simple learned keypoints. In *ICCV*, pages 22499–22508, 2023. 2

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3

[21] Xingyi He, Jiaming Sun, Yifan Wang, Sida Peng, Qixing Huang, Hujun Bao, and Xiaowei Zhou. Detector-free structure from motion. In *CVPR*, 2024. 1

[22] Alex Henry, Prudhvi Raj Dachapally, Shubham Pawar, and Yuxuan Chen. Query-key normalization for transformers. *arXiv preprint arXiv:2010.04245*, 2020. 3

[23] Andrew G Howard. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 4

[24] Hanwen Jiang, Arjun Karpur, Bingyi Cao, Qixing Huang, and André Araujo. Omniglue: Generalizable feature matching with foundation model guidance. In *CVPR*, 2024. 2

[25] Haibo Jin, Shengcai Liao, and Ling Shao. Pixel-in-pixel net: Towards efficient facial landmark detection in the wild. *IJCV*, pages 3174–3194, 2021. 2

[26] Shinjeong Kim, Marc Pollefeys, and Daniel Barath. Learning to make keypoints sub-pixel accurate. In *ECCV*, 2024. 2

[27] Jiefeng Li, Siyuan Bian, Ailing Zeng, Can Wang, Bo Pang, Wentao Liu, and Cewu Lu. Human pose regression with residual log-likelihood estimation. In *ICCV*, 2021. 2, 4, 5

[28] Xinghui Li, Kai Han, Shuda Li, and Victor Prisacariu. Dual-resolution correspondence networks. In *NIPS*, 2020. 1, 2, 6

[29] Yanjie Li, Sen Yang, Peidong Liu, Shoukui Zhang, Yunxiao Wang, Zhicheng Wang, Wankou Yang, and Shu-Tao Xia. Simcc: A simple coordinate classification perspective for human pose estimation. In *ECCV*, 2022. 2, 4

[30] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, 2018. 6

[31] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 3

[32] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. Lightglue: Local feature matching at light speed. In *ICCV*, 2023. 1, 2, 5, 6

[33] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 1, 2

[34] Peng Lu, Tao Jiang, Yining Li, Xiangtai Li, Kai Chen, and Wenming Yang. Rtmo: Towards high-performance one-stage real-time multi-person pose estimation. In *CVPR*, pages 1491–1500, 2024. 2

[35] Zixin Luo, Lei Zhou, Xuyang Bai, Hongkai Chen, Jiahui Zhang, Yao Yao, Shiwei Li, Tian Fang, and Long Quan. Aslfeat: Learning local features of accurate shape and localization. In *CVPR*, 2020. 2

[36] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 2015. 1

[37] Junjie Ni, Yijin Li, Zhaoyang Huang, Hongsheng Li, Hujun Bao, Zhaopeng Cui, and Guofeng Zhang. Pats: Patch area transportation with subdivision for local feature matching. In *CVPR*, pages 17776–17786, 2023. 6

[38] Junjie Ni, Guofeng Zhang, Guanglin Li, Yijin Li, Xinyang Liu, Zhaoyang Huang, and Hujun Bao. Eto: Efficient transformer-based local feature matching by organizing multiple homography hypotheses. *arXiv preprint arXiv:2410.22733*, 2024. 2

[39] Aiden Nibali, Zhen He, Stuart Morgan, and Luke Prendergast. Numerical coordinate regression with convolutional neural networks. *arXiv preprint arXiv:1801.07372*, 2018. 2, 4, 5

[40] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *arXiv:2304.07193*, 2023. 2

[41] Guilherme Potje, Felipe Cadar, André Araujo, Renato Martins, and Erickson R Nascimento. Xfeat: Accelerated features for lightweight image matching. In *CVPR*, 2024. 2

[42] Jerome Revaud, Cesar De Souza, Martin Humenberger, and Philippe Weinzaepfel. R2d2: Reliable and repeatable detector and descriptor. In *NIPS*, 2019. 1, 2

[43] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Neighbourhood consensus networks. In *NIPS*, 2018. 1, 2, 4

[44] Ignacio Rocco, Relja Arandjelović, and Josef Sivic. Efficient neighbourhood consensus networks via submanifold sparse convolutions. In *ECCV*, pages 605–621, 2020. 1, 2

[45] T-YLPG Ross and GKHP Dollár. Focal loss for dense object detection. In *CVPR*, pages 2980–2988, 2017. 5

[46] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *ECCV*, 2006. 2

[47] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *ICCV*, 2011. 1, 2

[48] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019. 1, 6

[49] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 1, 2, 5, 6

[50] Paul-Edouard Sarlin, Ajaykumar Unagar, Mans Larsson, Hugo Germain, Carl Toft, Viktor Larsson, Marc Pollefeys, Vincent Lepetit, Lars Hammarstrand, Fredrik Kahl, et al. Back to the feature: Learning robust camera localization from pixels to pose. In *CVPR*, 2021. 1

[51] 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 *CVPR*, 2018. 6

[52] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 1

[53] Jianbo Shi et al. Good features to track. In *CVPR*, 1994. 1

[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 *CVPR*, 2022. 2

[55] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[56] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021. 3

[57] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *CVPR*, 2021. 1, 2, 3, 4, 5, 6, 7, 8

[58] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. Inloc: Indoor visual localization with dense matching and view synthesis. In *CVPR*, 2018. 6

[59] Shitao Tang, Jiahui Zhang, Siyu Zhu, and Ping Tan. Quadtree attention for vision transformers. In *ICLR*, 2022. 2, 5

[60] Michał Tyszkiewicz, Pascal Fua, and Eduard Trulls. Disk: Learning local features with policy gradient. In *NIPS*, 2020. 2, 6

[61] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017. 1, 2, 3

[62] Qing Wang, Jiaming Zhang, Kailun Yang, Kunyu Peng, and Rainer Stiefelhagen. Matchformer: Interleaving attention in transformers for feature matching. In *ACCV*, 2022. 2, 5

[63] Yifan Wang, Xingyi He, Sida Peng, Dongli Tan, and Xiaowei Zhou. Efficient loftr: Semi-dense local feature matching with sparse-like speed. In *CVPR*, 2024. 1, 2, 3, 5, 6, 7, 8

[64] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *ECCV*, 2016. 2

[65] Wenqiang Zhang, Zilong Huang, Guozhong Luo, Tao Chen, Xinggang Wang, Wenyu Liu, Gang Yu, and Chunhua Shen. Topformer: Token pyramid transformer for mobile semantic segmentation. In *CVPR*, 2022. 3

[66] Qi Zhao, Zhi Yang, and Hai Tao. Differential earth mover's distance with its applications to visual tracking. *PAMI*, pages 274–287, 2008. 1

[67] Xiaoming Zhao, Xingming Wu, Jinyu Miao, Weihai Chen, Peter CY Chen, and Zhengguo Li. Alike: Accurate and lightweight keypoint detection and descriptor extraction. *TMM*, pages 3101–3112, 2022. 1, 2

[68] Qunjie Zhou, Torsten Sattler, and Laura Leal-Taixe. Patch2pix: Epipolar-guided pixel-level correspondences. In *CVPR*, 2021. 1, 2, 6

[69] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 2