ETO:Efficient Transformer-based Local Feature Matching by Organizing Multiple Homography Hypotheses

Junjie Ni[1] **Guofeng Zhang**[1*] **Guanglin Li**[1] **Yijin Li**[1]

Xinyang Liu[1] **Zhaoyang Huang**[2] **Hujun Bao**[1*]

[1]State Key Lab of CAD&CG, Zhejiang University [2]CUHK MMLab

Abstract

Recent developments have led to the emergence of transformer-based approaches for local feature matching, resulting in enhanced accuracy of matches. However, the time required for transformer-based feature enhancement is excessively long, which limits their practical application. In this paper, we propose methods to reduce the computational load of transformers during both the coarse matching and refinement stages. During the coarse matching phase, we organize multiple homography hypotheses to approximate continuous matches. Each hypothesis encompasses several features to be matched, significantly reducing the number of features that require enhancement via transformers. In the refinement stage, we reduce the bidirectional self-attention and cross-attention mechanisms to unidirectional cross-attention, thereby substantially decreasing the cost of computation. Overall, our method demonstrates at least 4 times faster compared to other transformer-based feature matching algorithms. Comprehensive evaluations on other open datasets such as Megadepth, YFCC100M, ScanNet, and HPatches demonstrate our method's efficacy, highlighting its potential to significantly enhance a wide array of downstream applications.

1 Introduction

Local feature matching [1, 2] is a fundamental problem in the field of computer vision and plays a significant role in downstream applications, including but not limited to SLAM [3–8], 3D reconstruction [9, 10], visual localization [11–13], and object pose estimation [14, 15]. However, traditional CNN-based methods [16, 17] often fail under extreme conditions due to the lack of a global receptive field, thus meeting failure under dramatic changes in scale, illumination, viewpoint, or weakly-textured scenes.

Recently, some methods [18–20] forgo traditional CNN-based approaches and base on Transformer [21] for better modeling the long-range dependencies. However, Transformer is widely known for its high computational complexity especially when it is applied in vision tasks where the computational complexity grows quadratically in the number of the input image tokens (i.e., patches). To reduce the inherent complexity associated with the Transformer, these methods generally adopt the coarse-to-fine strategy and incorporate more computationally efficient variants of the Transformer, such as the Linear Transformer [22]. Nevertheless, the computational overhead remains substantial and severely hinders the application demanding low-latency operations, such as tracking [23–25], or those requiring the processing of extensive datasets, such as large-scale mapping [26].

[*] Corresponding author

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

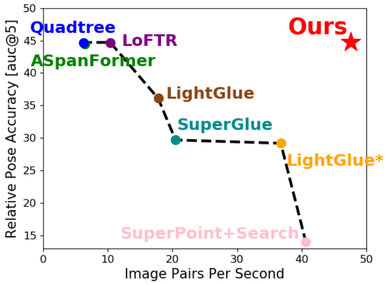

Figure 1: **ETO goes beyond the Pareto curve between accuracy and Efficiency.** This figure shows the performance of different state-of-the-art methods on YFCC100M. We take into account the time for the extraction of feature and their description. LightGlue and LightGlue* are different settings of LightGlue.

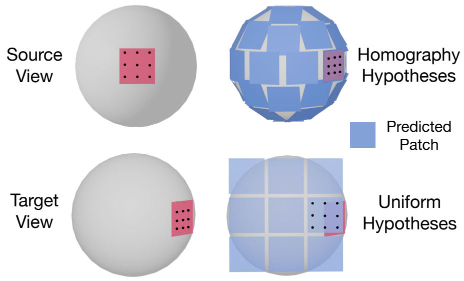

Figure 2: As demonstrated in the figure, there exists a correspondence between two red regions on the sphere. In contrast to uniform hypotheses, homography hypotheses approximate the correspondence function better, which allows for more precise matching results with fewer computational resources.

In this paper, we propose to solve the efficiency problem of transformer-based local feature matching. Our insights are twofold. First, we propose to introduce the homography hypothesis in the pipeline. The homography hypothesis is a kind of piece-wise smooth prior to the scene that has long been explored in the vision tasks [27]. It allows us to create larger patches and reduce the tokens number that need to be processed in the Transformer. However, it is non-trivial since the regular shape introduced by the homography hypothesis can bring significant errors, especially along the boundary. Besides, how to supervise the training of multiple homography hypotheses with the absence of ground truth remains a problem. Second, We empirically find it is redundant that the previous methods employ multiple self- and cross-attention in their fine-level stage since the coarse-level stage has conducted sufficient propagation. As a result, the computation complexity can be further reduced.

Specifically, we propose ETO, the Efficient Transformer-based Local Feature Matching by Organizing Multiple Homography Hypotheses. ETO follows previous methods [20, 19] and employs a two-stage coarse-to-fine pipeline. It first establishes matches at the patch level and then refines the matches to the sub-pixel level. In the first stage, ETO learns to predict a set of hypotheses, each encompassing multiple patches to be matched. We approximately assume that each patch to be matched within one hypothesis is on the same plane, and thus describe these matches under the homography transformation, as illustrated in Fig. 2. The homography hypotheses allow us to reduce the image tokens (patches) that are fed to the Transformer. For a typical image with a resolution of 640 × 480, Previous methods feed 80 × 60 tokens to the transformer with 1/8 resolution, while we only need to feed 20 × 15 with 1/32 resolution, which brings a significant speed up. To reduce the possible error due to the regular shape of the homography hypotheses, ETO subdivides the patches into multiple sub-patches and re-selects the correct hypothesis for each sub-patch. We model the problem of re-selection as a segmentation problem [28]. After that, ETO refines the matches in the second stage. Unlike previous methods that employ multiple self- and cross-attention, ETO only conducts one cross-attention, and the size of query tokens it use is much smaller than previous methods. We call it uni-directional cross-attention. Empirically we find uni-directional cross attention converges significantly faster at training while providing much higher efficiency. As shown in Fig. 1, ETO outpaces existing methods, achieving 4-5 times faster than LoFTR [20] and 2-3 times more rapid than LightGlue [29] while maintaining a comparable accuracy with them.

Our contributions can be summarized as follows. 1) We introduce multiple homography hypotheses for the local feature matching problem, which can greatly compress the number of tokens involved in the transformer. 2) We introduce uni-directional cross-attention in the refinement stage. This structure provides fast inference efficiency while maintaining accuracy. 3) Our method not only matches the performance of other transformer-based approaches on diverse open-source datasets such as Megadepth [30], YFCC100M [31], ScanNet [32], and HPatches [33], but it also operates at a significantly higher speed, outpacing all compared methods.

2 Related Works

Transformer-based Local Feature Matching. To find sparse correspondence between two images under diverse viewpoint movement conditions, traditional hand-crafted [34, 1, 2] or early learning-based approaches methods [17, 35] usually match keypoints [36, 37] with their descriptors after detecting them [16]. These kinds of matching pipelines are highly dependent on the description of unique feature points, and they fail naturally at dramatic viewpoint changes or poorly textured scenes. Thanks to transformer [21], researchers now have a toolbox to enhance the feature descriptors with global information. Earlier researches [36] integrate the information that where other key points are to each descriptors of key points. Then, [38, 39] modeling the mapping relationship as a continuous 2-d function. Sun *et al.* [20] constructs a global matching pipeline for each unit on the feature map with Transformer [21, 40–42]. The following works refine this pipeline with optical flow [19] or more efficient attention structure [43]. There exists other attempts. [44–46] follow another technical route without transformer. They try to merge the gap between optical flow [47, 41] and local feature matching with the concept of confidence. Our approach is based on a similar feature extractor of [20], while we parameterize more information for the units on the feature map, and finally extend the correspondence relationship to the homography relationship between them. The concept of parameterized units on the feature map for local feature matching is introduced by Ni *et al.* [18], but they parameterize the units only with scale.

Paramerterization in Local Feature Matching. Conventional techniques, as demonstrated in previous works [34, 48, 35], when confronted with appearance differences due to changes in viewpoint, try to construct feature descriptors which are invariant to these changes, involving scale, normal and rotation, etc. For rule-based methods [34], they create hand-crafted descriptors with scale space analysis [49]. For learning-based methods [16], they input many image pairs with different viewpoints, letting the neural network learn about the invariance of these changes in appearance. Nevertheless, to fully mitigate the impact of these appearance changes, it is essential to accurately estimate their effects. Recent efforts [38, 18, 50] have attempted to directly estimate the scale differences between images and resize the images to enhance the refinement stage of feature matching. In a similar route, [51] has focused on directly estimating rotation to calibrate the local features. Unlike the methods aimed solely at discovering more invariant feature descriptors, we argue that parameterized local appearance changes not only pose a challenge in finding accurate matches but also provide a direct avenue to achieve more accurate and more efficient matches over a broader spatial extent. It allows us to locally parameterize the image to many planes, thus creating multiple homography relationships, not just point-to-point correspondence.

Acceleration in Local Feature Matching. In the realm of local feature matching based on the Transformer architecture, three predominant strategies have been mainly employed to enhance computational efficiency in the past. The first approach involves substituting the Softmax function for the Optimal Transport algorithm introduced by [36]. The second approach seeks to replace the full Transformer with a linear Transformer [22]. LoFTR [20] incorporates both of these strategies, however, it introduces a larger number of matching units compared to SuperGlue [36], resulting in a considerably slower performance than [36]. The third strategy focuses on reducing the number of layers and units within the Transformer. LightGlue [29] introduces early termination [52, 53] and progressive unit selection strategies to accelerate computation, yielding significant improvements in speed. However, the performance of LightGlue heavily relies on SuperPoint [16], which puts a ceiling on its acceleration. In contrast, ETO relies on a more precise parameterized model, achieving higher coarse matching accuracy with a feature map whose resolution is 16 times smaller than LoFTR [20], which distinguishes ETO from another concurrent work Efficient LoFTR [54] that accelerates the algorithm with adaptive token selection on the feature map of the same resolution. Furthermore, during the fine matching stage, we introduce a uni-directional cross-attention mechanism, allowing us to achieve higher matching speed while sacrificing only a minimal amount of fine stage accuracy.

3 Method

Fig. 3 presents our comprehensive feature matching process, organized into three structured modules. These modules are interconnected through feature extractors inspired by U-Net [55] and local attributes generated by neural networks. For each unit i on a H/32 × W/32 resolution feature map M_1, we estimate the attributes of homography hypotheses H_i. For each unit j on H/8×W/8 resolution feature map M_2, it re-selects the optimal homography hypotheses $\hat{H}_j$ from nearby 9 hypotheses to minimize the projection errors. For the chosen unit k_j on a H/2×W/2 resolution feature map M_3, we fix its center point P_j^s at the source image and refine the coordinates of its projected point P_j^t at target image according to $\hat{H}_j$, then get the final matches P_j^{t*}. We introduce the feature extractor in Sec. 3.1, Sec. 3.2 details the estimation of the hypotheses. In Sec. 3.3 we describe the segmentation

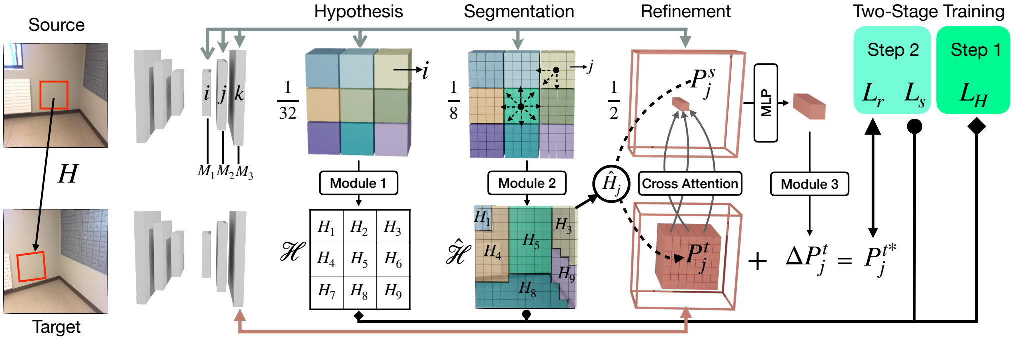

Figure 3: Given the source image $\mathcal{S}$ and target image $\mathcal{T}$, we first use a U-Net like feature extractor to get images' feature map at different resolution: M_1 (H/32 × W/32), M_2 (H/8 × W/8) and M_3 (H/2 × W/2). We use local 3 × 3 patches to illustrate our method: **(a)** We estimate homography hypotheses H_i for every feature after performing transformer. **(b)** We segment the map from these hypotheses to minimize projection errors. With the applied homography matrix $\hat{H}_j$, we can project the chosen source point P^s_j to target point P^t_j . **(c)** We update the P^t_j to P^{t*}_j after a uni-directional cross attention. The training process is split into two parts, the coarse and the fine. We train the coarse part with L_H, while training the fine part with L_s and L_r.

of the feature map, Sec. 3.4 delves into refining the matches, while Sec. 3.5 states our supervision methodology.

3.1 Feature Extraction

Following [20], we use ResNet-18 [56] as the basic feature extractor to get the feature map with the resolution of $N = \mathrm{H}/32 \times \mathrm{W}/32$, while we assume that the resolution of the source image and target image is the same. Here we get $N = \mathrm{H}/32 \times \mathrm{W}/32$ features and then perform stacked self-attention and cross attention layers between these N tokens to compute the feature map M_1. Although there will be N (more than 9) possible patches, our method is mainly performed on locally adjacent patches, so we will omit N in the future and take the local 3×3 patches to illustrate our method (as shown in Fig. 3). Then, we follow [18] to upsample the feature map M_1 with a U-Net [55] like structure. So, we can obtain the feature maps M_2 and M_3 at $1/8$ and $1/2$ scale, respectively.

3.2 Hypothesis Estimation

Traditional semi-dense feature matching methods [20, 19] often divide an image into thousands of units. For each unit, they perform bipartite graph matching [57]. Contrarily, we argue that bipartite graph matching can be extended to the local homography transformation as hypotheses that cover multiple units to be matched. This approach's merit lies in two folds: achieving more precise matches estimated during the first stage and reducing the number of units which are involved in transformer.

For each unit i on M_1 of source images, it is equipped with a feature f^1_i, a confidence score c_i, and a set of hypothesis homography parameters H_i (including the source positions $p^s_i \in \mathbb{R}^2$, target positions $p^t_i \in \mathbb{R}^2$, rotation $r_i \in \mathbb{R}^1$, scale $s_i \in \mathbb{R}^1$ and perspective $q_i \in \mathbb{R}^4$). And the unit on target images is indicated by a.

Homography Matrix. Initially, we outline the methodology for estimating each unit's local attributes $(p^s_i, p^t_i, r_i, s_i, q_i, c_i)$ and subsequently use these attributes to formulate the local homography matrix. Among these local attributes, the scale s_i, rotation r_i, and perspective q_i are more related to the feature itself and are regressed directly from features f_i on the source image through an MLP network. In contrast, target coordinates p^t_i, and confidence scores c_i are more related with the feature map of target images. They are acquired by first identifying the unit a^*_i with maximal similarity among all target units, and the similarity is defined as the cosine similarity of initial features f on the source image and target image. Following this, we construct new features $\hat{f}$ by executing the group-wise correlation [58] within the neighborhood of the target units on M_1.

$$\hat{f}_i = \underset{\delta \in Neighbor(a^*_i)}{\oplus} (< f^1_i, f^1_\delta >_g). \tag{1}$$

where $< *, * >_g$ is the group-wise correlation [58], and the group size in our method is 8, $Neighbor$ represents the 5×5 neighborhood of unit a^*_i, $\oplus$ indicates the operation of concatenation. With the new features $\hat{f}$, we use an MLP to process it to get the target position p^t_i and confidence c_i. With these attributes, in order to compute the homography matrix, we establish four target points B^t_i that correspond to four predetermined reference points:

$$B^t_i = p^t_i + \mathscr{R}(\mathscr{P}(B^s_i, q_i), r_i) * s_i. \tag{2}$$

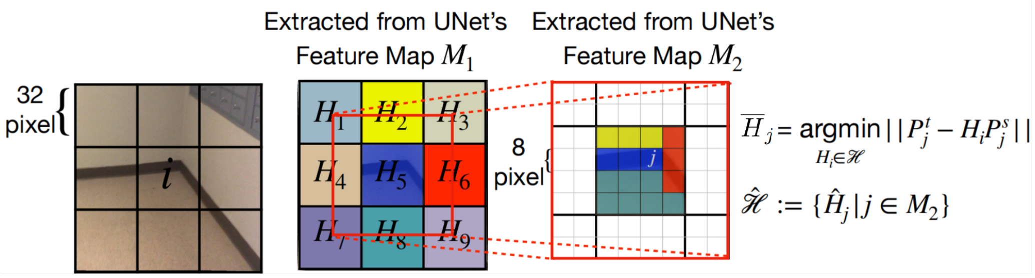
Extracted from UNet's Feature Map M_1
Extracted from UNet's Feature Map M_2
32 pixel
i
H_1
H_2
H_3
H_4
H_5
H_6
H_7
H_8
H_9
8 pixel
j
$\overline{H}_j = \underset{H_i \in \mathscr{H}}{\text{argmin}} ||P_j^t - H_i P_j^s||$
$\hat{\mathscr{H}} := \{\hat{H}_j | j \in M_2\}$

Figure 4: Any unit j on M_2 should be classified for a hypotheses in $\mathcal{H}$ to minimize projection error. Each H_i describes a plane.

Here B_i^s are four imaginary points on source image, while B_i^t are the corresponding target points of B_i^s on the target image. B_i^t is computed from B_i^s by following operation: $\mathscr{R}$ is the operation of rotation with parameter r_i, $\mathscr{P}$ is the perspective transformation operation with parameter q_i. These operations allow each variable within the homography matrix H_i to be deduced from 8 projection equations of 4 correspondence $B_i^t = H_i B_i^s$. Details regarding the specific implementation methods for rotation and perspective transformations will be included in the supplementary materials.

3.3 Segmentation

To propagate the homography hypotheses predicted by M_1 to a more detailed resolution. We introduce a segmentation operation at the feature map M_2 with the resolution H/8 × W/8. Segmentation is a per-unit classification task, and we predict a class for each unit j on M_2. Here, we only consider locally adjacent 3×3 patches, and all possible classes is defined as $\mathcal{H} = \{H_i | i = 1...9\}$. This classification (segmentation) involves that, for each unit j, selecting a hypothesis from $\mathcal{H}$ that minimizes the projection error at the center of unit j. After our proposed segmentation stage, each sub-unit j can find the hypotheses $\hat{H}_j$ that make its error smallest in all possible hypotheses $\mathcal{H}$. We illustrate the intuitive process of segmentation at Fig. 4.

Our proposed **segmentation** differs from traditional semantic segmentation. Instead of aiming for a specific semantic category, it targets a dynamic geometric relationship. To find the relationship, we introduce a new cosine similarity matrix C_j between the local feature f_j^2 on M_2 and all features f_i^1 on M_1, Positional encoding is employed during this phase to enhance local features, which is indispensable here because the hypothesis in $\mathcal{H}$ are not equivalent. To predicting the class $\hat{H}_j$ by finding the maximum C_j, we generate the computed groundtruth $\overline{H}_j$ as follows:

$$\overline{H}_j = \underset{H_i \in \mathcal{H}}{argmin} ||P_j^t - H_i P_j^s||. \tag{3}$$

where $\overline{H}_j$ is the optimal hypothesis that minimize the projection error. Then, we use focal loss [59] L_s to minimize the segmentation error between predicted $\hat{H}_j$ and the computed groundtruth $\overline{H}_j$. The probability of focal loss is set to C_j.

3.4 Refinement

Following [20], to enhance efficiency, only one of the points within each unit j is selected for refinement, which we denote the source and target point as P_j^s and P_j^t. Given $P_j^t = \hat{H}_j P_j^s$, the refinement stage finds the offset ΔP_j^t of each target point P_j^t relative to a fixed source point P_j^s. With the feature f^3 from M_3, conventional techniques unfold features f_k^3 from local regions in both source image and target image, followed by self-attention and cross-attention. We claim that this process is unnecessarily slow. Here we eliminate self-attention and reduce cross-attention from a bi-directional process to a uni-directional one. Specifically, the feature $\hat{f}_j^3$ is computed by querying the features f_k^3 within the neighborhood of the original target point P_j^t, and become the final feature vectors $\hat{f}_j^3$ for the point P_j on the images. We illustrate this process in supplementary materials. With the proof of experiments in Sec. 4.4, we find that refining a single feature in the local region of M_2 is enough to get expected results. Our findings indicate that this approach can largely diminish the computational load of attention mechanisms while still preserving highly accurate matching outcomes.

Finally, following [28], we process one fixed element of the final feature vector $\hat{f}_j^3$ via cross-attention as the confidence score c_j for the corresponding set of matches. It is supervised by if the error final match is larger than a threshold. The coordinates for the matched pairs on the source and target images are (P_j^s, P_j^{t*}). Therefore we can define the supervision of refinement as:

$$L_r = |P_j^{t*} - \overline{P}_j^t|_2 + BCE(c_j). \tag{4}$$

where BCE is binary cross entropy [60], which is a commonly utilized loss function for binary classification problems. Here we use BCE to recognize reliable matches. Here $\overline{P}_j^t$ is the ground truth value of P_j^{t*}.

3.5 Supervision

Indirect supervision for the homography hypothesis. Instead of supervising the attributes of the hypothesis directly, our approach employs indirect supervision by monitoring the correspondences of sampling points that are linked via the homography transformation. This design offers the advantage of leveraging an excessive number of ground truth matches to efficiently train a network focused on estimating a set of homography parameters. Using the ground truth camera pose and depth from datasets, we can get real matched points $\overline{P}^s$ in source and $\overline{P}^t$ in target images. We sample the matched points to train our method. For each points p in $\overline{P}^s$, we select only 3×3 adjacent hypotheses $\mathcal{H} = \{H_1, ..., H_9\}$ around it on M_1. where H_5 is in the hypothesis region's center and represents the hypothesis of the region containing point p. Similar to the necessity of segmentation stated in Sec. 3.3, direct supervision which applies H_5 to every p could result in avoidable errors, which arise from the mismatch between the irregular boundaries of planes in the real world and the grid-structured unit i on the source image. However, given that these sampling points are merely an auxiliary tool for the loss function, we can directly utilize the ground truth coordinates of the matches to supervise and eliminate classifying each sampling point at this stage. For each point p, we assume that it satisfies a certain homography transformation $\overline{H}_p$, and $\overline{H}_p$ satisfies the following defination:

$$\overline{H}_p = \underset{H_i \in \mathcal{H}}{argmin} \, |H_i p^s - p^t|_1. \tag{5}$$

We denote p in the source image as p^s and target image as p^t, we use the following error to optimize our method:

$$e_p = |\overline{H}_p p^s - p^t|_1. \tag{6}$$

where $| * |_1$ is the L1 norm error.

Classification or Correspondence Loss for Hypotheses. H_i is calculated on the base of identifying the matched unit a_i^* for unit i in target image. Therefore, if the estimated a_i^* significantly deviates from the ground truth $\overline{a_i^*}$, H_i would be entirely incorrect. In such conditions, we use classification loss to enhance the feature similarity between a_i^* and $\overline{a_i^*}$. In the opposite case, directly supervising the point correspondences calculated through H_i yields better results. The methodology is detailed as follows:

$$\begin{aligned} q_1 &= \{i | \theta_1 < |a_i^* - \hat{a}_i^*|_\infty\}, \\ q_2 &= \{i | \theta_1 \geq |a_i^* - \hat{a}_i^*|_\infty\}, \\ L_H &= \begin{cases} 1 - CosSim(f_i^1, f_{a_i^*}^1) & , i \in q_1, \\ \sum_{p \in P_i} e_p & , i \in q_2, \end{cases} \end{aligned} \tag{7}$$

where $| * |_\infty$ denotes the computation of infinity norm, $CosSim$ is the cosine similarity of two features in the feature map M_1. P_i is the set of sampled points p that apply H_i as $\overline{H}_p$.

Two-stage Training Process. In the entire feature matching process, we divide the training process into two stages, the coarse stage and the fine stage. The fine stage will freeze all the parameters of the coarse stage during training. The coarse stage includes the homography hypotheses estimation, while the fine stage includes segmentation and refinement. The losses used in these two parts are:

$$\begin{aligned} L_{coarse} &= L_H, \\ L_{fine} &= L_s + L_r. \end{aligned} \tag{8}$$

Methods	Average Corner Error			Point Accuracy	time
	1px(%)	3px(%)	5px(%)	1px(%)	ms
LoFTR [20]	46	77	86	63	218
SP [16]+LG [29]	44	73	85	51	101
SP [16]+SG [36]	41	72	82	47	79
SP [16]+search	38	68	81	32	81
Ours	42	72	82	52	**53**

Table 1: Evaluation on HPatches [33] for homography estimation.

3.6 Implementation Details

For feature extracting, we use Resnet-18 [56], then we perform transformer [21] five times at M_1. We implement uni-directional cross-attention once in the process of refinement. We train our outdoor model and indoor model respectively. The outdoor model is trained on the Megadepth [30] dataset, while the indoor model is trained on a mixed dataset of Megadepth and ScanNet [32]. The training process is divided into three stages: the first stage is training on data of 640x480 resolution; in the second stage, the longer side is scaled to 640, and some images are rotated by 90 degrees for adaptation training for the coarse; the third stage involves training the fine on 640x480 data. The learning rate used in the first stage is 1e-4. In the second stage is 5e-5, and in the third stage is 3e-4. Both models are trained using three RTX 3090 for 80 hours, with a batch size of 24 in the first stage and 16 in both the second and third stages. We perform all inferences using PyTorch, merely following the implementation of LightGlue to pre-compile the transformer in the coarse stage with PyTorch.

4 Experiment

We conducted these evaluations on four different datasets for outdoor and indoor relative pose estimation and homography estimation. These experiments demonstrate superior performance on various downstream tasks.

4.1 Homography Estimation

As our first experiment, we evaluate our quality of correspondences and the ability to fit the homography matrix for planar scenes on the HPatches [33] dataset.

Experimental Setup. We conducted comparative experiments using the image matching toolbox proposed by [61]. Our experiments were configured to replicate the settings outlined for SuperPoint[16], SuperGlue [36], and LoFTR [20] as shown in this toolbox. For LightGlue [29], we follow their open-source code settings. To estimate the homography, we employed the RANSAC algorithm with a threshold of 0.25 pixels, leveraging the OpenCV library. To comprehensively assess the performance of each method, we considered three key metrics: the proportion of matched points with an error within a 1-pixel threshold, the average corner distance for estimated homography matrices measuring less than 1/3/5 pixels, and the average computational time. These metrics were chosen to simultaneously evaluate the matching accuracy, homography estimation precision, and computational efficiency of the methods. We perform this experiment on a RTX2070 GPU, and we turn off all acceleration options for pytorch implementations, such as flash attention and precompilation. In order to get as close as possible to a real usage scenario, here, we do not use a warm-up operation when measuring the computing speed.

Dataset. HPatches [33] contains 52 sequences under significant illumination changes and 56 sequences that exhibit large variations in viewpoints. All images are resized with longer dimensions equal to 640.

Results. We compare ETO with SuperPoint [16], SuperGlue [36], LightGlue [29] and LoFTR [20]. According to Table. 1, our experimental results demonstrate that our method excels in homography estimation accuracy compared to SuperGlue, achieving lower errors within a 1-pixel threshold when compared to both SuperGlue and LightGlue. Furthermore, our approach is significantly faster, outpacing all other methods several times in the evaluation.

4.2 Outdoor Pose Estimation

We assess the efficacy of our approach for relative pose estimation in the same setting using two distinct datasets: YFCC100M [31] and Megadepth [30] for outdoor scenes.

Methods	Megadepth				YFCC100M			
	@5°	@10°	@20°	ms	@5°	@10°	@20°	ms
ASpanFormer [19]	58.6	72.2	81.7	158.5	44.5	63.5	78.1	155.5
Quadtree [43]	58.6	72.1	81.5	147.6	44.7	63.9	78.2	159.4
LoFTR [20]	57.5	71.2	80.8	93.2	44.7	63.6	78.3	96.3
SP [16]+LG [29]	51.5	67.7	78.9	64.2	36.1	56.2	73.1	60.8
SP [16]+LG* [29]	47.1	64.0	77.3	26.9	29.2	48.8	67.0	27.2
SP [16]+SG [36]	43.2	60.0	72.8	43.9	29.7	49.6	67.9	48.7
SP [16]+search	28.8	43.4	56.6	23.7	14.0	27.0	42.2	24.6
RoMa [62]	64.8	77.4	86.1	689	*	*	*	*
Tiny-RoMa [62]	36.2	53.6	67.5	29.0	*	*	*	*
Ours	51.7	66.6	77.4	**21.0**	**44.8**	**64.0**	**78.8**	**22.1**

Table 2: Evaluation on Megadepth [30] and YFCC100M [31] for outdoor pose estimation.

Experimental setup. We report the pose accuracy in terms of AUC metric at multiple thresholds (5°,10°,20°) and runtime for every approach, and the RANSAC threshold here is set as 0.25 pixel for all methods. All of the evaluations here are conducted on a RTX2080ti. We turn on flash-attention for LightGlue and turn on the pre-compilation to accelerate the transformer for LightGlue and ETO. Here LightGlue [29] is slower than SuperGlue [36] for the reason that following the default configuration LightGlue extracts 2048 keypoints and resizes the resolution of images to 1024, while SuperGlue extracts only 1024 keypoints and keep the resolution of images as the same. LightGlue* apply the setting of SuperGlue. It is imperative to highlight that our method encompasses 4800 points to be matched here, which is the same as LoFTR [20], ASpanFormer [19], and Quadtree [43]. To ensure an accurate representation of the actual computation speed, we initiate a warm-up phase for each method, consisting of 10 iterations, prior to conducting measurements.

Dataset. YFCC100M [31] encompasses an extensive repository comprising 100 million media assets. For our evaluation, we following [36] and focus on a subset of YFCC100M, specifically four handpicked image collections featuring prominent landmarks, in accordance with the criteria outlined in [36] and [20]. MegaDepth, on the other hand, comprises a dataset containing one million Internet-sourced images depicting 196 distinct outdoor scenes. To ensure the integrity of our evaluation protocol, in line with the guidelines presented in [18], we randomly select 1000 image pairs, guaranteeing that none of these pairs have been used in the training processes of any existing methods. All images in Megadepth and YFCC100M are resized with a resolution equal to 640*480.

Results. We compare ETO with SuperPoint [16], SuperGlue [36], LightGlue [29], LoFTR [20], ASpanFormer [19] and Quadtree [43]. According to the results shown in Table 2, on the easier outdoor cases in MegaDepth, the accuracy of our method for pose estimation is lower than advanced detector-free method but is higher than any detector-based approaches, while our runtime is at most 23% of the detector-free methods, 81% of the detector-based methods and 90% of the CNN-based methods. While on the more difficult outdoor cases in YFCC100M, the performance of our model is much better than detector-based methods and is comparable with detector-free methods. And still, our superiority on runtime is preserved.

4.3 Indoor Pose Estimation

We evaluate our method for indoor pose estimation with ScanNet-1500 [32] following [36, 20].

Experimental Setup. Just like outdoor cases, we report the pose accuracy in terms of the AUC metric at multiple thresholds (5°,10°,20°) and runtime for every approach. However, here we set all of the RANSAC thresholds as 0.5 pixels. All of the images are resized with longer dimensions equal to 640. This evaluation is conducted on RTX2080ti. We have done a warm-up here in measuring the efficiency.

Dataset. The ScanNet dataset represents a comprehensive indoor RGB-D collection encompassing 1,613 distinct sequences that cumulatively offer 2.5 million unique views. Each view within this dataset is meticulously annotated with a corresponding ground truth camera pose and depth map. We follow the same training and testing split used by [36].

Results. We compare our approach with SuperPoint [16], SuperGlue [36], LoFTR [20], ASpanFormer [19] and Quadtree [43]. The results are demonstrated in Table 3. We find that our results are comparable with LoFTR and are superior to SuperPoint+search and SuperPoint+SuperGlue, while much faster than any other methods.

Methods	Pose estimation AUC			average time
	@5°	@10°	@20°	ms
ASpanFormer [19]	24.5	45.0	62.8	160.0
Quadtree [43]	23.9	43.0	60.2	145.9
LoFTR [20]	21.4	40.3	57.2	94.2
SuperPoint [16]+SuperGlue [36]	13.7	29.8	47.2	63.1
Superpoint [16]+search	8.0	18.3	29.8	27.1
Ours	20.1	40.4	59.8	**24.2**

Table 3: Evaluation on Scannet [32] for indoor pose estimation.

Ablation	Pose estimation AUC			time
	@5°	@10°	@20°	ms
Base32 w/o Homography	9.5	21.3	36.3	8.5
Base8 w/o Homography (LoFTR coarse)	25.7	41.8	57.7	58.9
Base32 w/ Homography	28.5	46.2	61.4	8.5
Basic Refinement w/ segmentation	51.0	66.6	77.6	32.8
Uni-directional w/o segmentation	42.1	59.0	72.0	21.2
Full	51.7	66.6	77.4	22.0

Table 4: Ablation study based on Megadepth [30] for outdoor pose estimation.

4.4 Ablation Studies

To evaluate the impact of each design component on the overall structure, we perform an ablation study using the MegaDepth dataset. We systematically add each design element one at a time. The quantitative results are presented in Table 4.

Base32 w/o Homography We match the units on M_1 at H/32 * W/32 resolution and permit the target of unit centroid to be continuous, and we can compute it as Section. 3.2, while other parameters for the unit are still fixed. We output four virtual correspondences as matches. While it offers rapid processing, it does not achieve a high level of accuracy.

Base8 w/o Homography We set this ablation experiment as the coarse matching of LoFTR [20]. Here we match every possible 8*8 units. We output the center of corresponding units as matches. It is more accurate but too slow.

Base32 w/ Homography. Following Section. 3.2, we estimate the whole homography matrix and output four virtual correspondences as matches. It performs better than the coarse matching of LoFTR [20] while providing higher efficiency at the same time,

Basic Refinement w/ Segmentation. Following [20], we set a layer of transformer between 25 tokens on these two images and try to refine our results while the transformer is trained for 12 hours, which is the same as the training time of our uni-directional attention for the refinement stage. While full attention execution speed is considerably slower than that of uni-directional attention, its accuracy is merely comparable with the latter.

Uni-directional w/o Segmentation. Here we directly choose the homography hypotheses H_5 for each unit j which is in the center. Then, we conduct the refinement as the same. The results show that the segmentation stage significantly improves the accuracy.

5 Conclusion and Limitations.

In this paper, we propose Efficient Transformer-based Local Feature Matching by Organizing Multiple Homography hypotheses (ETO). ETO tries to approximate a continuous corresponding function with multiple homography hypotheses with fewer tokens fed to the transformer. Multiple datasets demonstrate that ETO delivers nearly comparable performance in relative pose estimation and homography estimation with other transformer-based methods, while its speed surpasses all of them by a large margin. However, there remains significant space for improvement in ETO's matching accuracy. Next, we could explore an end-to-end training mode, which would allow for further enhancement of the feature extractor at a fine-grained level. Moreover, we believe that intermediate-level features can provide not only segmentation information but also data conducive to more precise matching. Finally, the form of parametric scheme we present here may not be optimal and complete for homography transformation, so we will continue to explore better parametric schemes. These strategies are expected to enable our method to compete with approaches like PATS [18] and DKM [45] in terms of matching precision, without considerably compromising speed.

Acknowledgements. This work was partially supported by NSF of China (No. 61932003).

References

[1] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary R. Bradski. ORB: an efficient alternative to SIFT or SURF. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2564–2571. IEEE, 2011.

[2] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (SURF). *Comput. Vis. Image Underst.*, 110(3):346–359, 2008.

[3] Raul Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Trans. Robotics*, 31(5):1147–1163, 2015.

[4] Xingrui Yang, Hai Li, Hongjia Zhai, Yuhang Ming, Yuqian Liu, and Guofeng Zhang. Vox-fusion: Dense tracking and mapping with voxel-based neural implicit representation. In *IEEE International Symposium on Mixed and Augmented Reality*, pages 499–507. IEEE, 2022.

[5] Xinyang Liu, Yijin Li, Yanbin Teng, Hujun Bao, Guofeng Zhang, Yinda Zhang, and Zhaopeng Cui. Multi-modal neural radiance field for monocular dense slam with a light-weight tof sensor. In *Proceedings of the ieee/cvf international conference on computer vision*, pages 1–11, 2023.

[6] Jiarui Hu, Xianhao Chen, Boyin Feng, Guanglin Li, Liangjing Yang, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Cg-slam: Efficient dense rgb-d slam in a consistent uncertainty-aware 3d gaussian field. *arXiv preprint arXiv:2403.16095*, 2024.

[7] Jiarui Hu, Mao Mao, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Cp-slam: Collaborative neural point-based slam system. *Advances in Neural Information Processing Systems*, 36, 2024.

[8] Hongjia Zhai, Gan Huang, Qirui Hu, Guanglin Li, Hujun Bao, and Guofeng Zhang. Nis-slam: Neural implicit semantic rgb-d slam for 3d consistent scene understanding. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–11, 2024.

[9] Shuo Chen, Mao Peng, Yijin Li, Bing-Feng Ju, Hujun Bao, Yuan-Liu Chen, and Guofeng Zhang. Multi-view neural 3d reconstruction of micro-and nanostructures with atomic force microscopy. *Communications Engineering*, 3(1):131, 2024.

[10] Bangbang Yang, Yinda Zhang, Yijin Li, Zhaopeng Cui, Sean Fanello, Hujun Bao, and Guofeng Zhang. Neural rendering in a room: amodal 3d understanding and free-viewpoint rendering for the closed scene composed of pre-captured objects. *ACM Transactions on Graphics (TOG)*, 41 (4):1–10, 2022.

[11] 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.

[12] Zhaoyang Huang, Han Zhou, Yijin Li, Bangbang Yang, Yan Xu, Xiaowei Zhou, Hujun Bao, Guofeng Zhang, and Hongsheng Li. Vs-net: Voting with segmentation for visual localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6101–6111, 2021.

[13] Hongjia Zhai, Xiyu Zhang, Zhao Boming, Hai Li, Yijia He, Zhaopeng Cui, Hujun Bao, and Guofeng Zhang. Splatloc: 3d gaussian splatting-based visual localization for augmented reality. *arXiv preprint arXiv:2409.14067*, 2024.

[14] Yan Xu, Kwan-Yee Lin, Guofeng Zhang, Xiaogang Wang, and Hongsheng Li. Rnnpose: Recurrent 6-dof object pose refinement with robust correspondence field estimation and pose optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14880–14890, 2022.

[15] Guanglin Li, Yifeng Li, Zhichao Ye, Qihang Zhang, Tao Kong, Zhaopeng Cui, and Guofeng Zhang. Generative category-level shape and pose estimation with semantic primitives. In *Conference on Robot Learning*, pages 1390–1400. PMLR, 2022.

[16] 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*, pages 224–236, 2018.

[17] Mihai Dusmanu, Ignacio Rocco, Tomás 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.

[18] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17776–17786, 2023.

[19] 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 *European Conference on Computer Vision*, 2022.

[20] 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*, pages 8922–8931, 2021.

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

[22] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR, 2020.

[23] Adam W Harley, Zhaoyuan Fang, and Katerina Fragkiadaki. Particle video revisited: Tracking through occlusions using point trajectories. In *European Conference on Computer Vision*, pages 59–75. Springer, 2022.

[24] Yijin Li, Yichen Shen, Zhaoyang Huang, Shuo Chen, Weikang Bian, Xiaoyu Shi, Fu-Yun Wang, Keqiang Sun, Hujun Bao, Zhaopeng Cui, Guofeng Zhang, and Hongsheng Li. Blinkvision: A benchmark for optical flow, scene flow and point tracking estimation using rgb frames and events. In *European conference on computer vision*. Springer, 2024.

[25] Weikang Bian, Zhaoyang Huang, Xiaoyu Shi, Yitong Dong, Yijin Li, and Hongsheng Li. Context-pips: Persistent independent particles demands context features. *Advances in Neural Information Processing Systems*, 36, 2024.

[26] 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.

[27] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3): 24, 2009.

[28] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[29] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local Feature Matching at Light Speed. In *ICCV*, 2023.

[30] 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*, 2018.

[31] 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. *Commun. ACM*, 59(2):64–73, 2016.

[32] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas A. 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.

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

[34] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.*, 60(2):91–110, 2004.

[35] Jérôme Revaud, César Roberto de Souza, Martin Humenberger, and Philippe Weinzaepfel. R2D2: reliable and repeatable detector and descriptor. *Advances in Neural Information Processing Systems*, 32, 2019.

[36] 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.

[37] 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.

[38] 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 171–180. Springer, 2021.

[39] 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. In *European Conference on Computer Vision*, pages 317–334. Springer, 2022.

[40] Yijin Li, Xinyang Liu, Wenqi Dong, Han Zhou, Hujun Bao, Guofeng Zhang, Yinda Zhang, and Zhaopeng Cui. DELTAR: depth estimation from a light-weight tof sensor and RGB image. In *European Conference on Computer Vision*, pages 619–636. Springer, 2022.

[41] Zhaoyang Huang, Xiaoyu Shi, Chao Zhang, Qiang Wang, Yijin Li, Hongwei Qin, Jifeng Dai, Xiaogang Wang, and Hongsheng Li. Flowformer: A transformer architecture and its masked cost volume autoencoding for optical flow. *arXiv preprint arXiv:2306.05442*, 2023.

[42] Yijin Li, Zhaoyang Huang, Shuo Chen, Xiaoyu Shi, Hongsheng Li, Hujun Bao, Zhaopeng Cui, and Guofeng Zhang. Blinkflow: A dataset to push the limits of event-based optical flow estimation. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3881–3888. IEEE, 2023.

[43] Shitao Tang, Jiahui Zhang, Siyu Zhu, and Ping Tan. Quadtree attention for vision transformers. In *The International Conference on Learning Representations*. OpenReview.net, 2021.

[44] Prune Truong, Martin Danelljan, Radu Timofte, and Luc Van Gool. Pdc-net+: Enhanced probabilistic dense correspondence network. *arXiv preprint arXiv:2109.13912*, 2021.

[45] Johan Edstedt, Ioannis Athanasiadis, Mårten Wadenbäck, and Michael Felsberg. Dkm: Dense kernelized feature matching for geometry estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17765–17775, 2023.

[46] Songyan Zhang, Xinyu Sun, Hao Chen, Bo Li, and Chunhua Shen. Rgm: A robust generalist matching model. *arXiv preprint arXiv:2310.11755*, 2023.

[47] Zachary Teed and Jia Deng. RAFT: recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision*, pages 402–419. Springer, 2020.

[48] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6589–6598, 2020.

[49] Tony Lindeberg. Feature detection with automatic scale selection. *Int. J. Comput. Vis.*, 30(2): 79–116, 1998.

[50] Axel Barroso-Laguna, Yurun Tian, and Krystian Mikolajczyk. Scalenet: A shallow architecture for scale estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12808–12818, 2022.

[51] Miao Fan, Mingrui Chen, Chen Hu, and Shuchang Zhou. Occˆ 2net: Robust image matching based on 3d occupancy estimation for occluded regions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9652–9662, 2023.

[52] Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers. *arXiv preprint arXiv:1807.03819*, 2018.

[53] Maha Elbayad, Jiatao Gu, Edouard Grave, and Michael Auli. Depth-adaptive transformer. *arXiv preprint arXiv:1910.10073*, 2019.

[54] Yifan Wang, Xingyi He, Sida Peng, Dongli Tan, and Xiaowei Zhou. Efficient loftr: Semi-dense local feature matching with sparse-like speed. *arXiv preprint arXiv:2403.04765*, 2024.

[55] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[56] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[57] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in Neural Information Processing Systems*, 26, 2013.

[58] Xiaoyang Guo, Kai Yang, Wukui Yang, Xiaogang Wang, and Hongsheng Li. Group-wise correlation stereo network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3273–3282, 2019.

[59] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[60] Irving John Good. Rational decisions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 14(1):107–114, 1952.

[61] Qunjie Zhou, Torsten Sattler, and Laura Leal-Taixé. Patch2pix: Epipolar-guided pixel-level correspondences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4669–4678, 2021.

[62] Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. Roma: Robust dense feature matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19790–19800, 2024.

Checklist

1. Claims: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? Claims in the paper should match theoretical and experimental results in terms of how much the results can be expected to generalize. The paper's contributions should be clearly stated in the abstract and introduction, along with any important assumptions and limitations. It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper. Enter yes, no, or an explanation if appropriate. Answers are visible to reviewers. Claims in the paper should match theoretical and experimental results in terms of how much the results can be expected to generalize. [Yes] The main claims are clearly stated in the abstract and introduction.

2. Limitations: The authors are encouraged to create a separate "Limitations" section in their paper. The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). [Yes] Limitations are discussed in Sec. 5.

3. Theory, Assumptions and Proofs: If you are including theoretical results, did you state the full set of assumptions of all theoretical results, and did you include complete proofs of all theoretical results? All assumptions should be clearly stated or referenced in the statement of any theorems. The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, authors are encouraged to provide a short proof sketch to provide intuition. Enter yes, no, n/a, or an explanation if appropriate. Answers are visible to reviewers. [Yes] We provide a simple proof for using homography hypotheses in supplementary materials.

4. Experimental Result Reproducibility: If the contribution is a dataset or model, what steps did you take to make your results reproducible or verifiable? Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), release of a model checkpoint, or other means that are appropriate to your research. Enter yes, no, n/a, or an explanation if appropriate. Answers are visible to reviewers. [Yes]

5. Open Access to Data and Code: If you ran experiments, did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? Please see the NeurIPS code and data submission guidelines for more details. While we encourage release of code and data, we understand that this might not be possible, so no is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark). At submission time, to preserve anonymity, remember to release anonymized versions. Enter yes, no, n/a, or an explanation if appropriate. Answers are visible to reviewers. [No]

6. Experimental Setting/ Details: If you ran experiments, did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? The full details can be provided with the code, but the important details should be in the main paper, and information about how hyperparameters were selected should appear either in the paper or supplementary materials. Enter yes, no, n/a, or an explanation if appropriate. Answers are visible to reviewers. [Yes]

7. Experiment Statistical Significance: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments? [No]

8. Experiments Compute Resource: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments? [Yes]

9. Code Of Ethics: Have you read the NeurIPS Code of Ethics and ensured that your research conforms to it? Enter yes, no, or an explanation if appropriate. Answers are visible to reviewers. [Yes]

10. Broader Impacts: If appropriate for the scope and focus of your paper, did you discuss potential negative societal impacts of your work? Please see the Paper Checklist Guidelines for detailed instructions and examples of points that you may choose to discuss. Enter yes, no, n/a, or an explanation if appropriate. Answers are visible to reviewers. [No]

11. Safeguards: Do you have safeguards in place for responsible release of models with a high risk for misuse (e.g., pretrained language models)? Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model. Enter yes, no, n/a, or an explanation if appropriate. Answers are visible to reviewers. [Yes]

12. Licenses: If you are using existing assets (e.g., code, data, models), did you cite the creators and respect the license and terms of use? Cite the original paper that produced the code package or dataset. If possible, include a URL. Be sure to check the original license and respect its conditions. Enter yes, no, n/a, or an explanation if appropriate. Answers are visible to reviewers. [Yes]

13. Assets: If you are releasing new assets, did you document them and provide these details alongside the assets? Researchers should communicate the details of the dataset or the model as part of their submissions via structured templates. This includes details about training, license, limitations, etc. Enter yes, no, n/a, or an explanation if appropriate. Answers are visible to reviewers. [NA]

14. Crowdsourcing and Research with Human Subjects: If you used crowdsourcing or conducted research with human subjects, did you include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)? Including this information in the supplemental material is fine, but if the main contribution of your paper involves human subjects, then we strongly encourage you to include as much detail as possible in the main paper. According to the NeurIPS Code of Ethics, you must pay workers involved in data collection, curation, or other labor at least the minimum wage in your country. Enter yes, no, n/a, or an explanation if appropriate. Answers are visible to reviewers. [NA]

15. IRB Approvals: Did you describe any potential participant risks and obtain Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your institution), if applicable? Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper. For initial submissions, do not include any information that would break anonymity, such as the institution conducting the review. Enter yes, no, n/a, or an explanation if appropriate. Answers are visible to reviewers. [NA]