ECO-TR: Efficient Correspondences Finding Via Coarse-to-Fine Refinement

Dongli Tan[1,3⋆], Jiang-Jiang Liu[2,3⋆], Xingyu Chen[3], Chao Chen[3], Ruixin Zhang[3], Yunhang Shen[3], Shouhong Ding[3], and Rongrong Ji[1,4] ✉

[1] Media Analytics and Computing Lab, School of Informatics, Xiamen University
[2] TMCC, CS, Nankai University
[3] Youtu Lab, Tencent Technology (Shanghai) Co.,Ltd
[4] Institute of Artificial Intelligence, Xiamen University
{dltan921,j04.liu}@gmail.com, harleychen@tencent.com, chenchao.tencent@gmail.com, ruixinzhang@tencent.com, shenyunhang01@gmail.com, ericshding@tencent.com, rrji@xmu.edu.cn

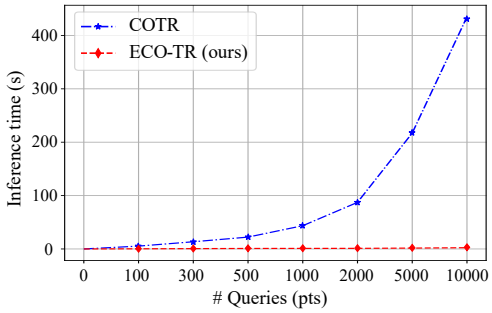
Abstract. Modeling sparse and dense image matching within a unified functional correspondence model has recently attracted increasing research interest. However, existing efforts mainly focus on improving matching accuracy while ignoring its efficiency, which is crucial for real-world applications. In this paper, we propose an efficient structure named Efficient Correspondence Transformer (**ECO-TR**) by finding correspondences in a coarse-to-fine manner, which significantly improves the efficiency of functional correspondence model. To achieve this, multiple transformer blocks are stage-wisely connected to gradually refine the predicted coordinates upon a shared multi-scale feature extraction network. Given a pair of images and for arbitrary query coordinates, all the correspondences are predicted within a single feed-forward pass. We further propose an adaptive query-clustering strategy and an uncertainty-based outlier detection module to cooperate with the proposed framework for faster and better predictions. Experiments on various sparse and dense matching tasks demonstrate the superiority of our method in both efficiency and effectiveness against existing state-of-the-arts. Project page: https://dltan7.github.io/ecotr/.

Keywords: Image Matching, Correspondence, Transformer, Functional Method, Coarse-to-Fine

1 Introduction

As a fundamental research direction in computer vision, finding the correspondences among pairs of images has been widely utilized in plenty of down-stream tasks, including optical flow estimation [19,8,53], visual localization [46,32,34], camera position calibration [15,41], 3D reconstruction [5,11], and visual tracking [30]. Given a pair of images, according to how the queries and correspondences

⋆ Authors contributed equally.



Time (s) / Method / # Queries	COTR	ECO-TR	Speed-up ratio
100	5.33	0.31	17.14
300	13.46	0.62	21.68
500	22.21	0.93	23.81
1,000	43.56	1.02	42.52
2,000	87.28	1.12	78.04
5,000	217.16	1.62	133.89
10,000	434.81	2.36	184.38

Fig. 1. Comparison of the inference time between the proposed ECO-TR and COTR [14]. The query numbers are set from 100 to 10,000. As we can see, the time-consuming of COTR increases linearly as the number of points increases, while our method basically does not change.

are determined, the applications mentioned above can be generally categorized into sparse matching and dense matching. The former focuses on two sets of keypoints being sparsely and respectively extracted from both images and matched to minimize a pre-defined alignment error [21,34,15]; the latter treats all pixels in the first image as queries which are densely mapped to the other image for correspondences [19,37,59,51].

The above two kinds of applications were studied independently for a long time, and various optimizations were designed separately. Recently, COTR [14] claims that these two applications can be naturally modeled within a unified framework since the only difference between the sparse and dense matching is the number of points to query. It proposes to recursively apply a transformer-based [4,52,7] model at multiple scales in a gradually zooming-in manner to obtain accurate correspondences. Though impressive performance has been achieved, its complex off-line pipeline and slow inference speed seriously limit its practicality in real-world applications.

We argue that there are three main reasons leading to the unsatisfactory COTR. The first is the recursive zoom-in refinement framework, which must re-extract the corresponding features in the next local patch matching. In the case of many queries, these features are likely to overlap, which means plenty of repeated and redundant calculations. The second is switching the role of the queries and correspondences to filter out the mismatched queries, which double the overall computation. The third is that the staged training strategy leads to unstable training convergence which needs to be carefully fine-tuned.

Instead of sacrificing speed for performance, in this work, we present an efficient correspondence transformer network (ECO-TR), showing that both efficiency and effectiveness can be achieved within a single feed-forward pass. Specifically, we propose to complete the coarse-to-fine refinement process of the found correspondences in a stage-by-stage manner. Our framework consists of a bottom-up convolutional neural network (CNN) for multi-scale feature extrac-

tion and several top-down transformer blocks corresponding to different matching accuracies. During the coarse-to-fine refinement process, rather than cropping image patches of different positions and sizes according to the coarsely predicted coordinates and recursively re-feeding them into CNN to obtain the corresponding feature maps, we obtain the multi-scale feature maps *w.r.t.* the input image at one time by taking advantages of the pyramid and translation invariance nature of modern CNNs, and directly crop on the collected feature maps. The proposed feature-level cropping method can effectively avoid repeated calculations. To a certain extent, the inference speed of the model does not increase linearly with the increase of query points.

To further improve the efficiency of our framework, an Adaptive Query-Clustering (AQC) module is proposed to gather similar queries into a cluster, which speeds up the inference. Moreover, we propose an uncertainty module to estimate the confidence of the predicted correspondences, which achieves good performance on outlier detection nearly for free. As illustrated in Table 1, our approach can process 1000 queries within one second on a single NVIDIA Telsa V100 GPU for a pair of images with size 800×800, which is around **40 times** faster than COTR under the same conditions.

To evaluate the performance of the proposed approach, we report the results on multiple challenging datasets covering both sparse and dense correspondence finding tasks. Experimental results demonstrate that our method surpasses COTR in performance and speed by a large margin. In addition, we conduct extensive ablation experiments to better understand the impact of each component in our framework. The contributions are summarized below:

- We propose a new coarse-to-fine framework for finding correspondence that can be applied to both sparse and dense matching tasks. Our method can be optimized end-to-end and evaluate an arbitrary number of queries within a single feed-forward.
- We design an adaptive query-clustering strategy and an uncertainty-based outlier filtering module to achieve a better balance between efficiency and effectiveness.
- Our method significantly outperforms the existing best-performing functional method in speed and still achieves comparable performance in sparse correspondence tasks and better in dense correspondence tasks.

2 Related Work

Sparse methods. The most common paradigm for sparse image matching pipelines consists of three stages: keypoint detection, keypoint description, and feature matching. In terms of the detection stage, a sparse set of repeatable and matchable keypoints are selected by the detection methods [31,35,2,50], which are robust against viewpoint changes and different lighting conditions. Then, the keypoints are described by patch-level input or image-level input. Patch-based description methods [44,24,45,10] take cropped patches as inputs and are usually trained by metric learning. Image-based description methods

such as [9,6,27,22,43] take a full image as input and apply fully-convolutional neural networks [20] to generate dense descriptors. This kind of method usually combines detector and descriptor, which share the same backbone in training and yield better performance on both tasks.

Traditional feature matching methods use Nearest Neighbor (NN) search to find potential matches. Recently, many approaches [3,54,55,56,40] filter outliers by heuristics or learned priors. SuperGlue [33] uses an attentional graph neural network and optimal transport method to obtain state-of-the-art performance on sparse matching tasks. Unlike the method mentioned above, given some keypoints as queries, COTR [14] refines the matches in the other image recursively by correspondence neural network. Following COTR, we design an end-to-end model to accelerate this scheme.

Dense methods. The main purpose of dense matching is to estimate the optical flow. NC-Net [29] represents all keypoints and possible correspondences as a 4D correspondence volume restricted to low-resolution images. Sparse NC-Net [28] applies sparse correlation layers instead of all possible correspondences to mitigate this restriction, whereby higher resolution images can be tackled. DRC-Net [17] reduces the computational cost and promotes performance by using coarse-resolution and fine-resolution feature maps of different layers. GLU-Net [48] finds pixel-wise correspondences by global and local features extracted from images with different resolutions. GOCor [47] disambiguates features in similar regions via an improved feature correlation layer. PDC-Net [49] excludes incorrect dense matches in occluded and homogeneous regions by estimating an uncertainty map and filtering the inaccurate correspondences. Patch2Pix [58] replaces pixel-level matches with patch-level match proposals and later refines them by regression layers. LoFTR [39] establishes accurate semi-dense matches with linear transformers in a coarse-to-fine manner. For COTR, the dense matching result is generated by interpolating sufficient sparse queries' results. Same with COTR, our method can give dense matching results by interpolation, too.

Functional methods. The functional method in image matching. COTR is the first one that obtains matches by a functional correspondence finding architecture. Given a pair of images and coordinates of one query, COTR regresses the possible match in the other image via a transformer-based correspondence finding network. Each query is processed independently, and dense correspondences are estimated by interpolating sparse correspondences using Delaunay triangulation of the queries. However, being a recursive method, it will be extremely time-consuming when many keypoints are queried. We mitigate this problem in an end-to-end manner, which runs dozens of times faster than COTR and achieves comparable or superior performance.

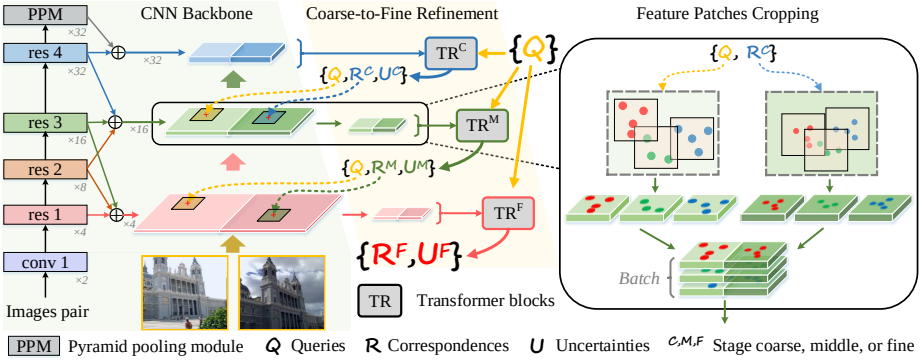


Fig. 2. The pipeline of our proposed framework. It takes a pair of images (bottom-left) and a set of queries ($\{Q\}$) of arbitrary numbers as input and outputs the correspondences ($\{R^F\}$) and uncertainty scores ($\{U^F\}$), respectively. The right part illustrates the feature patches cropping process during each prediction refinement stage.

3 Coarse-to-Fine Refinement Network

This section describes the proposed end-to-end framework that can find the correspondences for arbitrary queries given a pair of images within a single feed-forward pass in detail.

3.1 Overall Pipeline

We show a schematic diagram of the overall pipeline of the proposed framework in Fig. 2. It mainly consists of a bottom-up multi-scale feature extraction pathway based on the CNN and a top-down coarse-to-fine prediction refinement pathway based on the transformer. Given a pair of images I^A and I^B, we first resize them to the same spatial resolution ($B \times C \times H \times W$, B is the 'batch' dimension) and feed them into the CNN backbone to obtain multi-scale features. After that, the collected multi-scale features are used along with the input queries to predict the correspondences in a coarse-to-fine, gradually refining manner in the top-down pathway. We also predict an uncertainty score w.r.t. each correspondence representing how confident the network is of its prediction, which can be utilized to filter out the outliers nearly for free. Since it could be a bunch of queries to be processed in one feed-forward, we further introduce an adaptive query-clustering strategy to better balance efficiency and effectiveness. The following subsections describe the above-mentioned components in detail.

3.2 Efficient Feature Extraction

To obtain correspondence locations precisely, existing work usually crops image patches around potential matching regions and iteratively feeds them back into the network in a progressively enlarged manner. The main drawbacks of

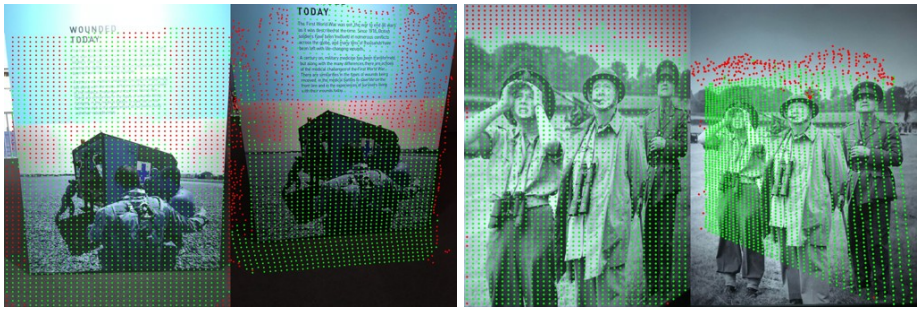


Fig. 3. Illustration of the uncertainty estimation branch. Green and red points indicate matches with low uncertainties and high uncertainties, respectively. ECO-TR gives ambiguous predictions in textureless regions and the border area with high uncertainties.

the aforementioned practice are: 1) the input image is cropped and resized into patches multiple times with different zoom-in factors around each query position. Each patch generated is then fed into the network, which involves many redundant computations. 2) Image patches for each query are cropped and processed by the network independently, which usually means serial processing and inefficient use of computational resources. We found that the main cause of these two shortcomings can be attributed to the setting of cropping patches at different spatial levels directly on the image.

Considering the pyramid and translation invariance nature of modern CNNs, we propose to alleviate the drawbacks mentioned above by deferring the cropping operation after the feature extraction step. Specifically, we first obtain the multi-scale feature maps w.r.t. each input image at one time and then directly crop on the collected feature maps to get feature patches at any position and scale. We take the ResNet-50[26] network as our backbone for multi-scale feature extraction without loss of generality. Following the previous success in generating more powerful and representative features, we attach a pyramid pooling module (PPM) [57] to capture more global information at the top of ResNet-50. The output of PPM and the side-outputs at `res1-4` stages of the ResNet-50 network are collected to build a hierarchical multi-scale feature integration structure. As shown in the left part of Fig. 2, to meet the needs of the subsequent top-down pathway which has three refinement stages (*i.e.*, coarse, middle, and fine), we choose to combine the intermediate outputs of {PPM, `res4`}, {`res2-4`} and {`res1-3`} stages, respectively. The integrated three sets of features (denoted as $\{\mathbb{F}^C, \mathbb{F}^M, \mathbb{F}^F\}$) are then resized to 1/32, 1/16, and 1/4 spatial resolutions w.r.t. the input stitched images pair, respectively.

3.3 Coarse-to-Fine Prediction Refinement

The schematic pipeline of the coarse-to-fine prediction refinement process is shown in the middle part (light orange parallelogram background) of Fig. 2. Generally speaking, it consists of three successively connected stages: coarse,

middle, and fine, respectively responsible for predicting correspondences with different precision. Each stage is a transformer building block of three encoders and three decoders. The coarse stage (TR^C) takes a set of queries $\mathbb{Q}$ of arbitrary numbers and the entire previously combined features $\mathbb{F}^C$ as input. It outputs the coarsely predicted correspondences set $\mathbb{R}^C$ along with their uncertainty scores. With the guidance of the coordinates in $\mathbb{Q}$ and $\mathbb{R}^C$, we crop square patches centered at them on the previously collected middle-level features $\mathbb{F}^M$ with a fixed window size of w^M, as illustrated by the dashed arrows in the middle left of Fig. 2. The cropped feature patches are then re-arranged into a new batch along with the input queries $\mathbb{Q}$ (normalized based on the cropping centers and window sizes) being forwarded to the next stage (*i.e.*, the middle stage (TR^M)). The fine stage shares similar procedures with the middle stage. After the fine stage, we obtain the final outputs of the proposed framework: the finest correspondences $\mathbb{R}^F$ and their uncertainty scores $\mathbb{U}^F$.

For each stage, concatenated backbone features are supplemented by 2D linear positional encoding in the sinusoidal format and flattened before being fed into the transformer encoder. During the decode stage, coordinates of queries with positional encoding attend to the output of the transformer encoder. Here, we disallow self-attention among the query points, for queries are independent of each other. COTR computes the cycle consistency errors and rejects matches whose errors are greater than a specified threshold to filter out uncertain matches, which doubles the computational cost. To further accelerate our framework, we introduce an uncertainty estimation branch. Two FFN branches follow the outputs of the last transformer decoder. One is employed to regress the corresponding relative coordinates of each query, and the other is to predict the uncertainties of these coordinates. Unreliable predictions with high uncertainties will be filtered during the inference stage.

Having predicted matches $\mathbb{R}^i$ and their uncertainties $\mathbb{U}^i$ of level i, loss $\mathbb{L}^i$ is calculated by:

$$\mathbb{L}^i = \left\| \mathbb{R}^i - \mathbb{R}^i_{gt} \right\| \cdot (1 - \mathbb{U}^i) + \lambda^i \cdot \mathbb{U}^i, \tag{1}$$

where $\mathbb{R}^i_{gt}$ is ground truth matches coordinates of queries and λ^i is the threshold of level i, where $i \in \{C, M, F\}$ represents stages coarse, middle, and fine. We set $\lambda^C = 0.1$, $\lambda^M = 0.05$, $\lambda^F = 0.01$ during training.

All three stages are supervised during training at the same time. Specifically, the final loss $\mathbb{L}$ is defined as

$$\mathbb{L} = \mathbb{L}^C + \mathbb{L}^M + \mathbb{L}^F. \tag{2}$$

Experiments show that the mid- and fine-level supervision during training provides predictions for corresponding stages and gives distinctive back-propagation signals to the CNN backbone, which is beneficial to the prediction of coarse-level. More details are provided in Sec. 4.5.

3.4 Adaptive Query-Clustering

The transformer structure is capable of processing many queries in one forward propagation. To improve efficiency, each patch should contain as many queries as

Algorithm 1: Adaptive Query-Clustering Algorithm

Input: Coordinates of queries Q; Matches of Q predicted by previous stage R; Iteration number t; K-means class number K_{num}; Distance threshold Th

Output: All patch pairs and corresponding matches in these patches

```
for i = 1 to t do
    Divide Q to K_num clusters by K-means algorithm, and assign class labels
     to every pair in (Q, R) ;
    for each class j do
        Set (Q', R') = all pairs labeled j ;
        Set c_q = the center coordinates of Q' ;
        Set c_r = the center coordinates of R' ;
        for each pair (q, r) in (Q', R') do
            if ||q − c_q|| > Th or ||r − c_r|| > Th
                Set the class label of (q, r) = −1
        end
        Crop patches centered at c_q and c_r and assign pairs labeled j to these
         patches
    end
    Set (Q, R) = all pairs labeled -1
end
for each pair (q, r) labeled −1 in (Q, R) do
    Crop patches centered at q and r, and assign pair (q, r) to these patches
end
```

possible. A straightforward practice is to directly slice the input images pair into two sets of grids according to the pre-defined window sizes and strides (usually, the stride is set equal to the corresponding window size). By densely coupling the patches between these two sets, any query-correspondence pair can be assigned to one of the patch pairs. We denote the above way of point-to-patch assignment as GRID for simplicity. However, we observe that an inevitable drawback of the query-correspondence independent kind of assignment strategies is that some matches will always exist around the patches' borders, which usually got sub-optimal matching results. We attribute this unsatisfying phenomenon to the lack of sufficient contextual information around the border area.

To achieve a better trade-off between efficiency and effectiveness, we propose an Adaptive Query-Clustering(AQC) algorithm to automatically and dynamically assign images patches for all query-correspondence pairs, as illustrated in Alg. 1. To demonstrate the superiority of AQC, we compare it with GRID in Sec. 4.5. Experiments show that clustering by AQC gives better performance than GRID.

3.5 Implementation Details

We implemented our model in PyTorch [25]. The local feature CNN uses a modified version of ResNet-50 as a backbone without pretraining. For coarse-to-

Table 1. Quantitative results on HPatches. Average End Point Error (AEPE) and Percentage of Correct Keypoints (PCK) are reported here. For each method, different thresholds (1px, 3px and 5px) of PCK are used. For a fair comparison of PCK, we report the reproduced results of COTR under the same image size.

Method	AEPE ↓	PCK-1px ↑	PCK-3px ↑	PCK-5px ↑
LiteFlowNet [13]	118.85	13.91	-	31.64
PWC-Net [38]	96.14	13.14	-	37.14
GLU-Net [48]	25.05	39.55	71.52	78.54
GLU-Net+GOCor [47]	20.16	**41.55**	-	81.43
COTR+Interp (reproduce) [14]	3.83	36.64	76.65	87.42
ECO-TR+Interp	**2.67**	40.19	**79.89**	**90.24**
COTR(reproduce) [14]	3.62	**38.72**	**80.90**	**90.85**
ECO-TR	**2.52**	38.02	79.79	90.71

fine refinement modules, we set the crop window size $w^M = 17$, $w^F = 13$. For the AQC module, we set $t = 1$, $K_{num} = 128$. The distance threshold Th is set to 0.8 times of the corresponding side of patches during training and 0.6 times during inference. More details can be found in the supplementary material.

4 Experiments

We evaluate our method across several datasets. We do not retrain or fine-tune our model on any other dataset for a fair comparison. Experiments are arranged as follows:

1. Dense matching tasks are evaluated on HPatches [1], KITTI [12], and ETH [36] datasets. Following COTR's evaluation protocol, we evaluate the results of sampled matches and interpolated dense optical flow.
2. We evaluate the pose estimation task on the same scene as COTR from Megadepth [18] dataset for sparse matching.
3. For ablations studies, we evaluate the impact of each proposed contribution using the ETH3D dataset.

4.1 Results on HPatches Dataset

We evaluate ECO-TR on the HPatches dataset for dense matching tasks in the first experiment. HPatches dataset contains 116 scenes, with 57 scenes changing in viewpoint and 59 scenes changing in lighting conditions. Following COTR, we evaluate the dense matching results on viewpoint-changing splits. Same with GLU-Net, we resize the reference image during our evaluation, while COTR is evaluated under the original scale in its experiments, which is not comparable in PCK value. Therefore, we reproduce the number of COTR under fair settings. For each method, we find a maximum of 1,000 matches from each pair. Then, we

Table 2. Quantitative results on KITTI. Average End Point Error (AEPE) and flow outlier ratio (Fl) on KITTI-2012 and KITTI-2015 are reported below. COTR† means we evaluated it with DenseMatching tools provided by the authors of GLU-Net.

Method	KITTI-2012		KITTI-2015	
	AEPE ↓	Fl.[%] ↓	AEPE ↓	Fl.[%] ↓
LiteFlowNet [13]	4.00	17.47	10.39	28.50
PWC-Net [38]	4.14	20.28	10.35	33.67
DGC-Net [23]	8.50	32.28	14.97	50.98
GLU-Net [48]	3.34	18.93	9.79	37.52
RAFT [42]	-	-	5.04	17.8
GLU-Net+GOCor [47]	2.68	15.43	6.68	27.57
PDC-Net [49]	2.08	7.98	5.22	15.13
COTR† + Interp. [14]	1.47	8.79	3.65	13.65
ECO-TR + Interp.	**1.46**	**6.64**	**3.16**	**12.10**
COTR† [14]	1.15	6.98	2.06	9.14
ECO-TR	**0.96**	**3.77**	**1.40**	**6.39**

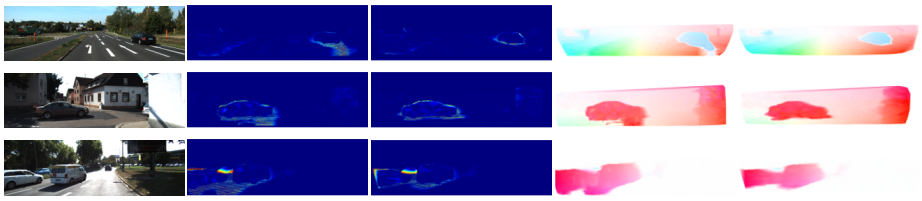
interpolate correspondences on the Delaunay triangulation map of the queries and get the dense correspondences. The results are reported in Table 1.

For the dense matching task, ECO-TR achieves better performance than COTR under all metrics. For the matching accuracy, COTR is a little better than ECO-TR evaluated by PCK. We attribute this gap to the difference in image resolution. COTR can utilize high-resolution images via four recursive zoom-ins, which is unmanageable for ECO-TR due to its end-to-end architecture. The average endpoint error(AEPE) for ECO-TR is lower than COTR.

4.2 Results on KITTI Dataset

We use the KITTI dataset to evaluate the performance of our method under real road scenes. KITTI2012 dataset contains static scenes only, while the KITTI2015 dataset has more challenging dynamic scenes. Following [42,47,14], we use the training split, which has ground truth of camera intrinsics, poses, and depth maps collected by LIDAR. All methods above-mentioned were trained on other datasets and evaluated on this training split. In line with previous works[DGC, GLU, GOC, COTR], We employ the Average End-point Error (AEPE) and percentage of optical flow outliers (Fl) as evaluation metrics. Here, inliers are defined as AEPE<3 pixels or $< 5\%$. Same with COTR, We sample $40,000$ points for a fair comparison.

As shown in Table 2, our method outperforms all others on these two datasets. For example, our method achieves AEPE= 1.09 and 1.70 on KITTI-2012 and KITTI-2015, respectively, which is 30% higher than COTR on average. The interpolated results are slightly worse than the sparse results, yet still better than the other dense methods by a large margin, including PDC-Net, which estimates dense correspondence and excludes unreliable matches, too. Qualitative examples on KITTI dataset are illustrated in Fig. 4.



(a) Input image (b) COTR (c) ECO-TR (d) COTR (e) ECO-TR

Fig. 4. Qualitative results on KITTI – We show the error map (Columns (b, c)) and optical flow (Columns (d, e)) for three pairs from KITTI-2015. ECO-TR provided clearer outlines of moving objects.

Table 3. Results on ETH3D. We evaluated our method over pairs of ETH3D images sampled from different frame intervals. Average End Point Error (AEPE) are reported here. Lower AEPE is better.

Method	AEPE ↓						
	rate=3	rate=5	rate=7	rate=9	rate=11	rate=13	rate=15
LiteFlowNet [13]	1.66	2.58	6.05	12.95	29.67	52.41	74.96
PWC-Net [38]	1.75	2.10	3.21	5.59	14.35	27.49	43.41
DGC-Net [23]	2.49	3.28	4.18	5.35	6.78	9.02	12.23
GLU-Net [48]	1.98	2.54	3.49	4.24	5.61	7.55	10.78
COTR+Interp. [14]	1.71	1.92	2.16	2.47	2.85	3.23	3.76
ECO-TR+Interp.	**1.52**	**1.70**	**1.87**	**2.06**	**2.21**	**2.44**	**2.69**
COTR [14]	1.66	1.82	1.97	2.13	2.27	2.41	2.61
ECO-TR	**1.48**	**1.61**	**1.72**	**1.81**	**1.89**	**1.97**	**2.06**

4.3 Results on ETH3D Dataset

ETH3D dataset contains ten image sequences of indoor and outdoor scenes and provides ground truth sparse correspondences under different frame intervals. Following COTR, we report the performance of our method under pairs with seven different intervals, from 3 to 15, respectively. The results in Table 3 show that our proposal outperforms other competitors under all rates, especially when matching pairs with large geometric transformations, *i.e.* pairs with a higher rate.

4.4 Results on Megadepth Dataset

MegaDepth [18] images show extreme viewpoint and appearance variations. The poses of images are generated via structure-from-motion and multi-view stereo (MVS) methods, which can be used as ground truth during evaluation. We choose St. Paul's Cathedral as our test scene. We sample 900 pairs of images that have commonly visible regions. Mean average accuracy(mAA) at a 5° and 10° error threshold are reported here, where the error is defined as the maximum of angular error in rotation and translation. For COTR, we follow the strategy used in its paper and evaluate the performance under different numbers of matches. For

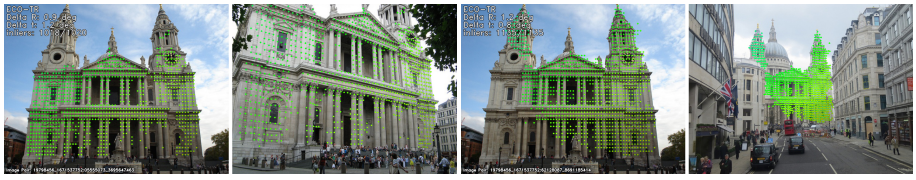


Fig. 5. Qualitative results on MegaDepth dataset. We set queries on left images and obtain matches in right images. We estimate the relative pose between image pairs and the angular errors in rotation and translation are reported in the upper-left corner. The number of inliers evaluated by epipolar distance is shown as well.

Table 4. Quantitative results on MegaDepth. We evaluated our method against COTR with different numbers of predicted matches. Mean average accuracy(mAA) at a 5° and 10° error threshold are reported here.

#Matches / Method	N=2048		N=1024		N=512		N=300		N=100	
	@5	@10	@5	@10	@5	@10	@5	@10	@5	@10
COTR	0.443	0.660	0.448	**0.665**	0.434	0.650	**0.434**	**0.654**	0.410	0.626
ECO-TR	**0.453**	**0.661**	**0.452**	0.664	**0.447**	**0.656**	0.430	0.652	**0.418**	**0.636**

ECO-TR, we estimate the scale of buildings in pairs first. We sample sparse points in one image as queries and predict their correspondences by coarse-stage ECO-TR. Then, we crop original images and obtain patches that share regions of two images. We resize cropped patches and feed them to the model again, and take random points in one image as queries and find reliable matches with low uncertainty in the other image. To further improve performance, a cycle consistency check is applied here. To compare the performance under the same number of matches, we drop some matches randomly. For a fair comparison, other settings except the matching method are fixed for two methods. The results in Table 4 show that ECO-TR gives a comparable performance, while our pipeline is significantly faster than COTR. Qualitative examples of MegaDepth are illustrated in Fig. 5.

4.5 Ablation Studies

In this section, we will conduct several ablation experiments on ETH3D dataset to discuss the efficiency and effectiveness of our method. More ablations on KITTI dataset are provided in the supplementary material.

Analysis of inference time. Table 5 reports the time cost of each component of ECO-TR. Table 6 further compares the runtimes of the corresponding components between ECO-TR and COTR with similar GPU memory costs (about 8192MB). As can be seen, all components in ECO-TR are more efficient than

Table 5. Detailed inference time (sec.) of each component.

#points	pre- and post-process	backbone	TR^C	TR^M	TR^F
0.1k	0.036	0.064	0.012	0.120	0.081
10k	0.037	0.062	0.026	0.480	1.740

Table 6. Detailed comparison of inference time (sec.) with COTR.

Method	#points	backbone	transformer	pre- and post-process	sum
COTR	0.1k	0.67	3.74	1.03	5.44
ECO-TR	0.1k	0.06	0.21	0.04	0.31
COTR	10k	92.55	60.71	280.27	433.53
ECO-TR	10k	0.06	2.24	0.05	2.35

COTR's, where the end-to-end framework (pre- and post-process in an end-to-end manner) contributes most to the efficiency.

Analysis of multistage zoom-ins. First, we analyze the effect of multistage zoom-ins architecture. As shown in Table 7, we evaluate the result of ECO-TR without middle- and fine-stage inference (E_C). It leads to substantially worse results. Adding middle-stage inference benefits the results(E_{CM}) but is still less effective than three stages version(E_{CMF}). We can see that the design of three-stage refinement is essential for good performance. Furthermore, instead of training with the supervision of all three branches, we detach the middle-stage and fine-stage branches during training($E_{C'}$). The result shows that it leads to worse results, which indicates that deeply supervised models give more distinctive features which yield better performance.

Analysis of clustering method. We test the performance of our pipeline with different clustering methods mentioned in Sec. 3.4. GRID and AQC are evaluated under the same distance threshold Th for a fair comparison. The results of AQC and GRID clustering are provided in E_{AQC} and E_{GRID} in Table 7, respectively. The result shows that our Adaptive Query-Clustering yields better performance than GRID clustering. The gap between the two strategies gradually increases as the difficulty of test pairs increases.

Analysis of transformer type. We replace the full attention transformer block in our middle- and fine-stage model with the linear substitution [16] used in LoFTR, and the corresponding results are shown in E_{linear}. Compared with full attention result in E_{fully}, the AEPE of pairs with rate=3 increases by 0.02 and pairs with rate=3,5 increase by 0.01, while still better than other methods in Table 3 by a large margin. Furthermore, the average inference time of ECO-TR is reduced by 20 percent when the linear transformer is applied, but this generally leads to a slight degradation in performance. It shows our pipeline has the potential to be further accelerated at a small cost.

Table 7. Ablations on ETH3D. We evaluate the impact of each component of our method over image pairs from the ETH3D dataset. Pairs are sampled from 3 different frame intervals, which indicate varying difficulty levels. Average End Point Error (AEPE) is reported here. Lower AEPE is better.

AEPE ↓	E_C	$E_{C'}$	E_{CM}	E_{CMF}	E_{AQC}	E_{GRID}	E_{fully}	E_{linear}	E_{cyc}	E_{unc}	$E_{cyc+unc}$
rate=3	5.21	5.63	2.47	1.53	1.53	1.64	1.53	1.55	1.53	1.48	1.48
rate=9	7.17	7.50	3.09	2.11	2.11	2.32	2.11	2.12	2.00	1.82	1.81
rate=15	9.19	9.53	3.83	2.72	2.72	3.10	2.72	2.74	2.45	2.08	2.06

Analysis of outlier filtering method. We compare the effectiveness of the uncertainty-based outlier filtering algorithm in Table 7. We run ECO-TR with different filtering strategies. E_{cyc} employs cycle consistency check as a filter, and E_{unc} employs uncertainty estimation as a filter. The result shows that filtering by uncertainty estimation gives better performance than filtering by cycle consistency check method. Additionally, $E_{cyc+unc}$ employs uncertainty estimation and cycle consistency checks together. Results show that by further using these two strategies together, ECO-TR achieves better performance.

5 Conclusions

This paper introduces an efficient coarse-to-fine transformer-based network for local feature matching. The main improvement is from three sides: 1) We propose an efficient network structure in a coarse-to-fine manner, fully utilizing the information from different layers and can be trained integrally. 2) We design an adaptive query-clustering (AQC) module that gathers similar query points in the same patch and achieves a better balance between efficiency and effectiveness. 3) An uncertainty-based outlier detection module is proposed to filter out the queries without correspondence. Our method significantly improves the speed of functional matching and achieves comparable or better performance both on sparse and dense matching tasks.

Limitations The main limitation is that the training of ECO-TR requires a large amount of GPU computing resources. In addition, simple interpolation and refinement techniques limit the performance of dense estimates. We leave these for the future work.

Acknowledgments This work was supported by the National Science Fund for Distinguished Young Scholars (No.62025603), the National Natural Science Foundation of China (No. U21B2037, No. 62176222, No. 62176223, No. 62176226, No. 62072386, No. 62072387, No. 62072389, and No. 62002305), Guangdong Basic and Applied Basic Research Foundation(No.2019B1515120049), and the Natural Science Foundation of Fujian Province of China (No.2021J01002).

References

1. Balntas, V., Lenc, K., Vedaldi, A., Mikolajczyk, K.: Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5173–5182 (2017)
2. Barroso-Laguna, A., Riba, E., Ponsa, D., Mikolajczyk, K.: Key. net: Keypoint detection by handcrafted and learned cnn filters. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5836–5844 (2019)
3. Bian, J., Lin, W.Y., Matsushita, Y., Yeung, S.K., Nguyen, T.D., Cheng, M.M.: Gms: Grid-based motion statistics for fast, ultra-robust feature correspondence. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4181–4190 (2017)
4. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision. pp. 213–229. Springer (2020)
5. Cheng, J., Leng, C., Wu, J., Cui, H., Lu, H.: Fast and accurate image matching with cascade hashing for 3d reconstruction. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1–8 (2014)
6. DeTone, D., Malisiewicz, T., Rabinovich, A.: Superpoint: Self-supervised interest point detection and description. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 224–236 (2018)
7. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
8. Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2758–2766 (2015)
9. Dusmanu, M., Rocco, I., Pajdla, T., Pollefeys, M., Sivic, J., Torii, A., Sattler, T.: 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. pp. 8092–8101 (2019)
10. Ebel, P., Mishchuk, A., Yi, K.M., Fua, P., Trulls, E.: Beyond cartesian representations for local descriptors. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 253–262 (2019)
11. Fan, B., Kong, Q., Wang, X., Wang, Z., Xiang, S., Pan, C., Fua, P.: A performance evaluation of local features for image-based 3d reconstruction. IEEE Transactions on Image Processing **28**(10), 4774–4789 (2019)
12. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. The International Journal of Robotics Research **32**(11), 1231–1237 (2013)
13. Hui, T.W., Tang, X., Loy, C.C.: Liteflownet: A lightweight convolutional neural network for optical flow estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8981–8989 (2018)
14. Jiang, W., Trulls, E., Hosang, J., Tagliasacchi, A., Yi, K.M.: Cotr: Correspondence transformer for matching across images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6207–6217 (2021)
15. Jin, Y., Mishkin, D., Mishchuk, A., Matas, J., Fua, P., Yi, K.M., Trulls, E.: Image matching across wide baselines: From paper to practice. International Journal of Computer Vision **129**(2), 517–547 (2021)

16. Katharopoulos, A., Vyas, A., Pappas, N., Fleuret, F.: Transformers are rnns: Fast autoregressive transformers with linear attention. In: International Conference on Machine Learning. pp. 5156–5165. PMLR (2020)
17. Li, X., Han, K., Li, S., Prisacariu, V.: Dual-resolution correspondence networks. Advances in Neural Information Processing Systems **33**, 17346–17357 (2020)
18. Li, Z., Snavely, N.: Megadepth: Learning single-view depth prediction from internet photos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2041–2050 (2018)
19. Liu, C., Yuen, J., Torralba, A.: Sift flow: Dense correspondence across scenes and its applications. IEEE transactions on pattern analysis and machine intelligence **33**(5), 978–994 (2010)
20. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3431–3440 (2015)
21. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60**(2), 91–110 (2004)
22. Luo, Z., Zhou, L., Bai, X., Chen, H., Zhang, J., Yao, Y., Li, S., Fang, T., Quan, L.: Aslfeat: Learning local features of accurate shape and localization. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6589–6598 (2020)
23. Melekhov, I., Tiulpin, A., Sattler, T., Pollefeys, M., Rahtu, E., Kannala, J.: Dgc-net: Dense geometric correspondence network. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 1034–1042. IEEE (2019)
24. Mishchuk, A., Mishkin, D., Radenovic, F., Matas, J.: Working hard to know your neighbor's margins: Local descriptor learning loss. Advances in neural information processing systems **30** (2017)
25. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32** (2019)
26. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28** (2015)
27. Revaud, J., Weinzaepfel, P., De Souza, C., Pion, N., Csurka, G., Cabon, Y., Humenberger, M.: R2d2: repeatable and reliable detector and descriptor. arXiv preprint arXiv:1906.06195 (2019)
28. Rocco, I., Arandjelović, R., Sivic, J.: Efficient neighbourhood consensus networks via submanifold sparse convolutions. In: European conference on computer vision. pp. 605–621. Springer (2020)
29. Rocco, I., Cimpoi, M., Arandjelović, R., Torii, A., Pajdla, T., Sivic, J.: Neighbourhood consensus networks. Advances in neural information processing systems **31** (2018)
30. Rosten, E., Drummond, T.: Fusing points and lines for high performance tracking. In: Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1. vol. 2, pp. 1508–1515. Ieee (2005)
31. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In: European conference on computer vision. pp. 430–443. Springer (2006)
32. Sarlin, P.E., Cadena, C., Siegwart, R., Dymczyk, M.: From coarse to fine: Robust hierarchical localization at large scale. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12716–12725 (2019)

33. Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: Superglue: Learning feature matching with graph neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4938–4947 (2020)
34. Sattler, T., Leibe, B., Kobbelt, L.: Improving image-based localization by active correspondence search. In: European conference on computer vision. pp. 752–765. Springer (2012)
35. Savinov, N., Seki, A., Ladicky, L., Sattler, T., Pollefeys, M.: Quad-networks: unsupervised learning to rank for interest point detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1822–1830 (2017)
36. Schops, T., Schonberger, J.L., Galliani, S., Sattler, T., Schindler, K., Pollefeys, M., Geiger, A.: A multi-view stereo benchmark with high-resolution images and multi-camera videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3260–3269 (2017)
37. Sun, D., Roth, S., Black, M.J.: A quantitative analysis of current practices in optical flow estimation and the principles behind them. International Journal of Computer Vision **106**(2), 115–137 (2014)
38. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8934–8943 (2018)
39. Sun, J., Shen, Z., Wang, Y., Bao, H., Zhou, X.: Loftr: Detector-free local feature matching with transformers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8922–8931 (2021)
40. Sun, W., Jiang, W., Trulls, E., Tagliasacchi, A., Yi, K.M.: Acne: Attentive context normalization for robust permutation-equivariant learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11286–11295 (2020)
41. Svärm, L., Enqvist, O., Kahl, F., Oskarsson, M.: City-scale localization for cameras with known vertical direction. IEEE transactions on pattern analysis and machine intelligence **39**(7), 1455–1461 (2016)
42. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: European conference on computer vision. pp. 402–419. Springer (2020)
43. Tian, Y., Balntas, V., Ng, T., Barroso-Laguna, A., Demiris, Y., Mikolajczyk, K.: D2d: Keypoint extraction with describe to detect approach. In: Proceedings of the Asian Conference on Computer Vision (2020)
44. Tian, Y., Fan, B., Wu, F.: L2-net: Deep learning of discriminative patch descriptor in euclidean space. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 661–669 (2017)
45. Tian, Y., Yu, X., Fan, B., Wu, F., Heijnen, H., Balntas, V.: Sosnet: Second order similarity regularization for local descriptor learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11016–11025 (2019)
46. Toft, C., Stenborg, E., Hammarstrand, L., Brynte, L., Pollefeys, M., Sattler, T., Kahl, F.: Semantic match consistency for long-term visual localization. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 383–399 (2018)
47. Truong, P., Danelljan, M., Gool, L.V., Timofte, R.: Gocor: Bringing globally optimized correspondence volumes into your neural network. Advances in Neural Information Processing Systems **33**, 14278–14290 (2020)
48. Truong, P., Danelljan, M., Timofte, R.: Glu-net: Global-local universal network for dense flow and correspondences. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6258–6268 (2020)

49. Truong, P., Danelljan, M., Van Gool, L., Timofte, R.: Learning accurate dense correspondences and when to trust them. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5714–5724 (2021)
50. Tyszkiewicz, M., Fua, P., Trulls, E.: Disk: Learning local features with policy gradient. Advances in Neural Information Processing Systems **33**, 14254–14265 (2020)
51. Ummenhofer, B., Zhou, H., Uhrig, J., Mayer, N., Ilg, E., Dosovitskiy, A., Brox, T.: Demon: Depth and motion network for learning monocular stereo. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5038–5047 (2017)
52. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
53. Weinzaepfel, P., Revaud, J., Harchaoui, Z., Schmid, C.: Deepflow: Large displacement optical flow with deep matching. In: Proceedings of the IEEE international conference on computer vision. pp. 1385–1392 (2013)
54. Yi, K.M., Trulls, E., Ono, Y., Lepetit, V., Salzmann, M., Fua, P.: Learning to find good correspondences. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2666–2674 (2018)
55. Zhang, J., Sun, D., Luo, Z., Yao, A., Zhou, L., Shen, T., Chen, Y., Quan, L., Liao, H.: Learning two-view correspondences and geometry using order-aware network. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5845–5854 (2019)
56. Zhao, C., Cao, Z., Li, C., Li, X., Yang, J.: Nm-net: Mining reliable neighbors for robust feature correspondences. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 215–224 (2019)
57. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2881–2890 (2017)
58. Zhou, Q., Sattler, T., Leal-Taixe, L.: Patch2pix: Epipolar-guided pixel-level correspondences. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4669–4678 (2021)
59. Zhou, T., Brown, M., Snavely, N., Lowe, D.G.: Unsupervised learning of depth and ego-motion from video. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1851–1858 (2017)