PATS: Patch Area Transportation with Subdivision for Local Feature Matching

Junjie Ni[1,2*], Yijin Li[1,2*], Zhaoyang Huang[3], Hongsheng Li[3], Hujun Bao[1,2],
Zhaopeng Cui[1] and Guofeng Zhang[1,2†]
[1]State Key Lab of CAD&CG, Zhejiang University [2]ZJU-SenseTime Joint Lab of 3D Vision
[3]Multimedia Laboratory, The Chinese University of Hong Kong

Abstract

Local feature matching aims at establishing sparse correspondences between a pair of images. Recently, detector-free methods present generally better performance but are not satisfactory in image pairs with large scale differences. In this paper, we propose Patch Area Transportation with Subdivision (PATS) to tackle this issue. Instead of building an expensive image pyramid, we start by splitting the original image pair into equal-sized patches and gradually resizing and subdividing them into smaller patches with the same scale. However, estimating scale differences between these patches is non-trivial since the scale differences are determined by both relative camera poses and scene structures, and thus spatially varying over image pairs. Moreover, it is hard to obtain the ground truth for real scenes. To this end, we propose patch area transportation, which enables learning scale differences in a self-supervised manner. In contrast to bipartite graph matching, which only handles one-to-one matching, our patch area transportation can deal with many-to-many relationships. PATS improves both matching accuracy and coverage, and shows superior performance in downstream tasks, such as relative pose estimation, visual localization, and optical flow estimation. The source code is available at `https://zju3dv.github.io/pats/`.

1. Introduction

Local feature matching between images is essential in many computer vision tasks which aim to establish correspondences between a pair of images. In the past decades, local feature matching [3, 40] has been widely used in a large number of applications such as structure from motion (SfM) [44, 64], simultaneous localization and mapping (SLAM) [30,36,62], visual localization [19,41], object pose estimation [22, 61], etc. The viewpoint change from the source image to the target image may lead to scale variations, which is a long-standing challenge in local feature matching. Large variations in scale leads to two severe consequences: Firstly, the appearance is seriously distorted, which makes learning the feature similarity more challenging and impacts the correspondence accuracy. Secondly, there may be several pixels in the source image corresponding to pixels in a local window in the target image. However, existing methods [33, 40] only permit one potential target feature to be matched in the local window, and the following bipartite graph matching only allows one source pixel to win the matching. The coverage of correspondences derived from such feature matches is largely suppressed and will impact the downstream tasks.

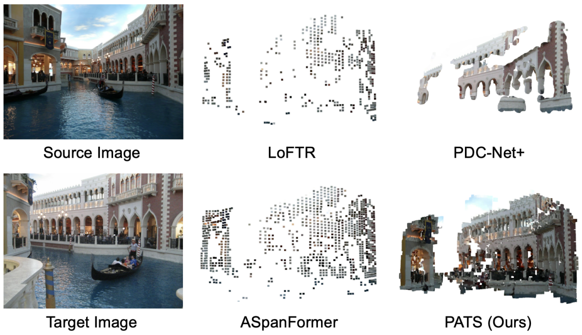
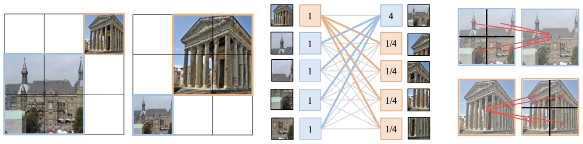


Figure 1. **Two-view reconstruction results of LoFTR [49], ASpanFormer [7], PDC-Net+ [58] and our approach on MegaDepth dataset [27].** PATS can extract high-quality matches under severe scale variations and in indistinctive regions with repetitive patterns, which allows semi-dense two-view reconstruction by simply triangulating the matches in a image pair. In contrast, other methods either obtain fewer matches or even obtain erroneous results.

Before the deep learning era, SIFT [33] is a milestone that tackles the scale problem by detecting local features on an image pyramid and then matching features crossing pyramid levels, called scale-space analysis. This technique is also adopted in the inference stage of learning-based local features [39]. Recently, LoFTR abandons feature detection stage and learns to directly draw feature matches via simultaneously encoding features from both images based

*Junjie Ni and Yijin Li contributed equally to this work.
†Guofeng Zhang is the corresponding author.



(a) Equal-sized patches (b) Patch area transportation (c) Crop & resize

Figure 2. **Scale Alignment with Patch Area Transportation.** Our approach learns to find the many-to-many relationship and scale differences through solving the patch area transportation. Then we crop the patches and resize the image content to align the scale, which remove the appearance distortion.

on the attention mechanism. By removing the information bottleneck caused by detectors, LoFTR [49] produces better feature matches. However, LoFTR does not handle the scale problem and the scale-space analysis is infeasible in this paradigm because conducting attention intra- and inter-different scales will bring unbearable increasing computational costs. As a result, the scale curse comes back again.

In this paper, we propose **P**atch **A**rea **T**ransportation with **S**ubdivision (PATS) to tackle the scale problem in a detector-free manner. The appearance distortion can be alleviated if the image contents are aligned according to their scale differences before feature extraction. As shown in Fig. 2, if the target image is simply obtained by magnifying the source image twice, a siamese feature encoder will produce features with large discrepancies at corresponding locations. The discrepancies are corrected if we estimate the scale difference and resize the target image to half before feature extraction. Considering that scale differences are spatially varying, we split the source image into equal-sized patches and then align the scale patch-wisely. Specifically, we identify a corresponding rectangular region in the target image for each source patch and then resize the image content in the region to align the scale. By also splitting the target image into patches, the rectangular regions can be represented with patches bounded by boxes. Based on this representation, one source patch corresponds to multiple target patches. Moreover, the bounding box may be overlapped, indicating that one target patch may also correspond to multiple source patches. Here comes the question: how can we find many-to-many patch matches instead of one-to-one [7,42,49]?

We observe that finding target patches for a source patch can be regarded as transporting the source patch to the target bounding box, where each target patch inside the box occupies a portion of the content. In other words, the area proportion that the target patches occupying the source patch should be summed to 1. Motivated by this observation, we propose to predict the target patches' area and formulate patch matching as a patch area transportation problem that transports areas of source patches to target patches with visual similarity restrictions. Solving this problem with Sinkhorn [10], a differential optimal transport solver, also encourages our neural network to better capture complex visual priors. Once the patch matching is finished, the corresponding bounding boxes can be easily determined. According to the patch area transportation with patch subdivision from coarse to fine, PATS significantly alleviates appearance distortion, which largely eases the difficulty of feature learning to measure visual similarity. Moreover, source patches being allowed to match overlapped target patches naturally avoid the coverage reduction problem. After resizing the target regions according to estimated scale differences, we subdivide the corresponding source patch and target region to obtain finer correspondences, dubbed as scale-adaptive patch subdivision. Fig. 1 shows qualitative results of our approach.

Our contributions in this work can be summarized as three folds: 1) We propose patch area transportation to handle the many-to-many patch-matching challenge and grants the ability that learning scale differences in a self-supervised manner to the neural network. 2) We propose a scale-adaptive patch subdivision to effectively refine the correspondence quality from coarse to fine. 3) Our patch area transportation with subdivision (PATS) achieves state-of-the-art performance and presents strong robustness against scale variations.

2. Related works

Detector-Free Feature Matching. Given two images to be matched, classic methods [3,33,40] usually adopt three-phase pipeline: feature detection [12], feature description [14, 39], and feature matching [42, 63]. The phase of feature detection reduces the search space of matching but also introduces an information bottleneck. When the feature detector fails to extract feature points, we can not find good correspondence even with perfect descriptors and matching strategy. Consequently, an alternative approach that concentrates on producing correspondences directly from unprocessed images, known as a detector-free framework, has emerged. Earlier works [24] in detector-free matching usually rely on cost volume [48] to enumerate all the possible matches. Recently, Sun et al. [49] propose encoding features from both images based on the Transformer [18, 26, 46, 59], which better model long-range dependencies and achieve satisfying performance. Another concurrent work [20] show that this kind of framework bridge the task of local feature matching and optical flow estimation [17, 25, 45]. After that, many variants have been proposed [7, 52, 60]. However, these works rarely focus on the scale difference between the image pair. To fill the gap, we propose a new detector-free framework that efficiently and effectively tackles the scale problem. We show that by removing the one-to-one matching constraint and alleviating the appearance distortion, our methods obtain a significantly larger number of matches which is also more accurate.

Scale-Invariant Feature Matching. These methods attempt to address the challenges posed by the potential scale differences between the image pair. To this end, traditional methods [3, 33] usually detect local features on an image pyramid and then matching features crossing pyramid levels, which is called scale-space analysis [28]. The technique is also adopted in the inference stage of learning-based methods [21, 34, 39]. Recent methods tend to mitigate scale by directly predicting scale difference [2], overlap area [8], warping deformation [4,38], or iteratively estimating correspondence and computing co-visible area [20]. Compared to these methods, we propose patch area transportation which simultaneously infers the scale difference and correspondence. The structure of the area transportation enables the network to learn scale differences in a self-supervised manner and also encourages the network to better capture complex visual priors.

Optimal Transport in Vision-Related Tasks. Optimal transport has been widely used in various computer vision tasks such as object detection [5, 15], semantic segmentation [31], domain adaptation [9], shape matching [47], visual tracking [66], semantic correspondence [32] and so on. Most of these works seek to find the best one-to-one assignment with the lowest total cost, which is equal to bipartite matching. The problem is a special case of transportation problem which allows many-to-many relationships. In this paper, we show that introducing the transportation modeling to the feature matching can be of great benefit, and we hope it can further inspire research in other areas.

3. PATS

As presented in Fig. 3, given a pair of images, we divide them into equal-sized patches and seek corresponding patches in the target image for each patch in the source image. One source patch transported to the target image may be stretched and cover multiple target patches, so we regress the areas of target patches and find many-to-many patch matches via the proposed patch area transportation algorithm. Once the patches are matched, we prune unreliable matches, crop and resize the target patches to align the source patches' scale, which alleviates the appearance distortion derived from scale variations. Then, we conduct the patch subdivision to enter the finer level for correspondence refinement. Finally, PATS obtains accurate correspondences according to the matched fine-grained patches. We introduce how to extract features from patches in Sec. 3.1, the patch transportation in Sec. 3.2, the hierarchical patch subdivision in Sec. 3.3, and finally, explain the supervision in Sec. 3.4.

3.1. Patch Feature Extraction

Given the image pair $\mathbf{I}_S, \mathbf{I}_T \in \mathbb{R}^{H \times W \times 3}$, we divide them into square patches of equal size s. We use $\mathcal{S} := \{1, ..., i, ..., N\}$ and $\mathcal{T} := \{1, ..., j, ..., M\}$ to index the source patches and the target patches, respectively. For each patch, we associate it with a 2D position $\mathbf{p} \in \mathbb{R}^2$, a descriptor $\mathbf{f} \in \mathbb{R}^d$, and an area $a \in \mathbb{R}$. $\mathbf{p}$ is the image coordinate of each patch's center, and the descriptors $\mathbf{f}$ is obtained through the feature extraction module. a reflects the area of the patch when scaled to the source image. Certainly, each source patch has the unit area ($a_i = 1$). The areas of target patches a_j are regressed by our neural network. A straightforward method to supervise a_j is collecting the ground truth beforehand, but they are unavailable. Instead, we learn to predict a_j by the differential patch transportation, which will be introduced in section 3.2.

Inspired by LoFTR, we first encode an image pair with a siamese convolutional neural network [37], where the size of the feature map achieves $H/s \times W/s \times d$, and then encode both image features with stacked self-attention and cross-attention layers. This step produces the descriptor $\mathbf{f}_i \in \mathbb{R}^{N \times d}$, $\mathbf{f}_j \in \mathbb{R}^{M \times d}$ for the source patches and target patches. We have $N = M = (H/s \times W/s)$ at the beginning while N and M gradually increase during the following coarse-to-fine subdivision. For all target patches, we append a small CNN to the descriptors to regress their areas a_j. With such triplet features $(\mathbf{p}, a, \mathbf{f})$ for each patch, we propose a patch transportation method to match the patches.

3.2. Patch Matching with Area Transportation

There are non-negligible scale differences between the image pairs to be matched, which makes one source patch may correspond to multiple target patches, and vice versa. Ignoring the scale and forcing one-to-one bipartite graph matching reduces the match number and impacts the matching accuracy. As shown in Fig. 3, when we transport a source patch to the target image, its area may be dispersed into multiple target patches, which are all corresponding patches. To this end, we propose to formulate the patch matching as an optimal transportation problem, i.e., transporting source patches' areas to target patches and minimizing the transportation cost defined as patch similarities.

Patch Area Transportation. Given all patches' areas a and descriptors $\mathbf{f}$, we define the cost between patches as the correlation of their descriptors:

$$C_{i,j} = -\left\langle \mathbf{f}_i, \mathbf{f}_j \right\rangle, \forall (i,j) \in \mathcal{S} \times \mathcal{T}, \tag{1}$$

where $\langle \, . \, , \, . \, \rangle$ is the inner product. A part of the source patch transported to the target image may be out of border or occluded, which is invisible. To handle these cases, we define the patch transportation as partial transportation. Formally, we seek to figure out the transportation matrix $\mathbf{P}$ to minimize the cost function such that transporting the source patches' area to the target patches:

$$\mathbf{P}\mathbf{1}_M \preceq \mathbf{a}_S, \; \mathbf{P}^T\mathbf{1}_N \preceq \mathbf{a}_T \tag{2}$$

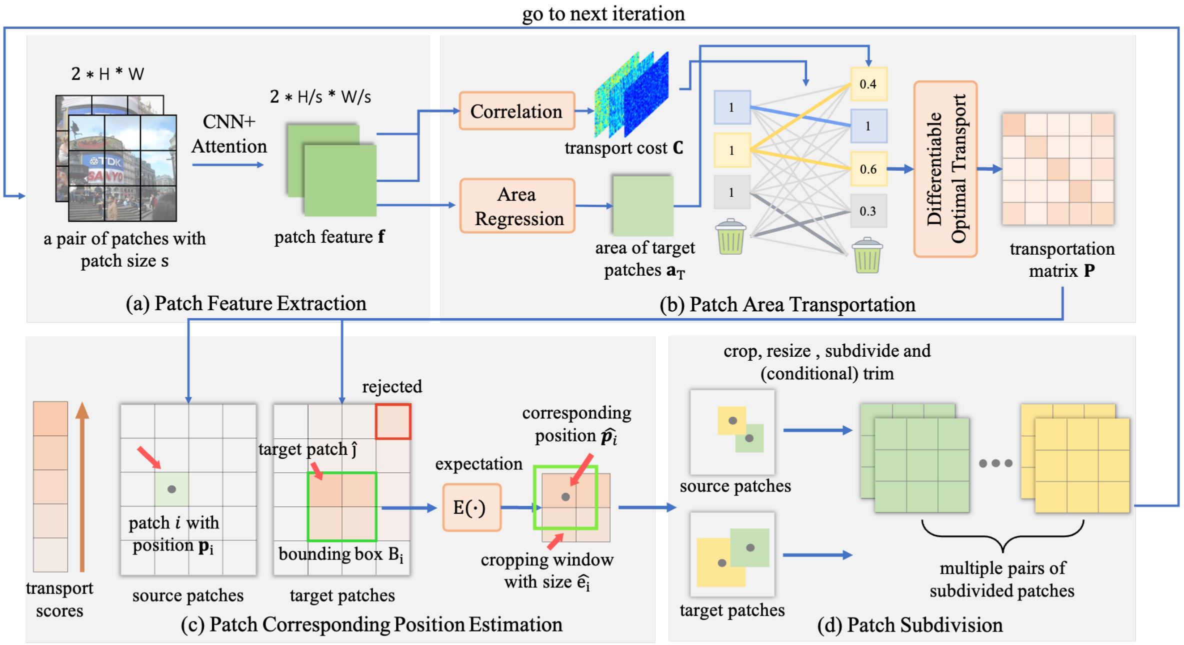

go to next iteration
2 * H * W
CNN+
Attention
2 * H/s * W/s
a pair of patches with patch size s
patch feature f
(a) Patch Feature Extraction
Correlation
transport cost C
Area Regression
area of target patches a_T
1
1
1
0.4
1
0.6
0.3
Differentiable Optimal Transport
transportation matrix P
(b) Patch Area Transportation
transport scores
patch i with position p_i
source patches
rejected
target patch ĵ
bounding box B_i
target patches
expectation
E(·)
corresponding position p̂_i
cropping window with size ê_i
(c) Patch Corresponding Position Estimation
crop, resize , subdivide and (conditional) trim
source patches
target patches
multiple pairs of subdivided patches
(d) Patch Subdivision

Figure 3. **Overview of PATS.** We a) extract features for patches. Then, we b) formulate the patch area transportation by setting source patches' area $\mathbf{a}_S$ as $\mathbf{1}_N$, regressing target patches' area $\mathbf{a}_T$, and bound the transportation via visual similarities $\mathbf{C}$. The feature descriptors $\mathbf{f}$ that produce $\mathbf{C}$ and the area regression $\mathbf{a}_T$ are learned by solving this problem differentially. The solution of this problem $\mathbf{P}$ also reveals many-to-many patch relationships. Based on $\mathbf{P}$, we c) find corresponding regions, represented by target patches inside a bounding box B_i, for each source patch. The exact patch corresponding position $\hat{\mathbf{p}}_i$ is the position expectation over B_i. After cropping and resizing image contents according to the obtained window sizes, which align the contents to the same scale, we d) subdivide the cropped contents to smaller patches and enter the next iteration.

where $\mathbf{a}_S \in \mathbb{R}^N, \mathbf{a}_T \in \mathbb{R}^M$ denote the area of the source patches and the target patches. As all source patches' area is 1, we have $\mathbf{a}_S = \mathbf{1}_N$. Following [42, 49], we add a dustbin to the patches collection in both the source image and the target image to handle the partial transportation. The transportation cost from a source patch i to a target patch j is defined as $P_{i,j} \times C_{i,j}$. The optimal transportation problem is computed efficiently with the Sinkhorn [10] algorithm, which is differentiable. The transportation solution that minimizes the total transportation cost maximizes the visual similarities weighted by the transport areas.

With the restrictions derived from differential patch area transportation, our neural network simultaneously learns to infer complex area relationships and patch descriptors. For example, transporting multiple source patches to the same target patch that is visually similar urges the area of target patches to be large such that it can accommodate these source patches. otherwise, some source patches leak to other target patches or the dustbin with a potentially higher transport cost. Our patch transportation formulation that minimizes the transport cost, the negative of patch similarity, weighted by the transport mass, the area, also encourages our network to properly predict the feature descriptors and areas to accomplish this goal. Compared with the bipartite graph matching that directly maximizes feature similarities, our patch transportation introduces a better inductive bias and provides guidance for the patch subdivision.

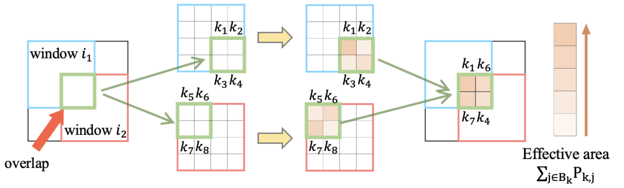
Patch Corresponding Position Estimation. The next step is to calculate the precise positions of corresponding patches in the target image for all source patches. After patch area transportation, we obtained the transportation matrix $\mathbf{P}$ that represents the relationship between each source patch and each target patch. For a source patch i, we first find the target patch $\hat{j}$ that occupies the largest area of i:

$$\hat{j} = \underset{1 \le j \le M}{\arg\max}\, \mathbf{P}_{i,j}. \tag{3}$$

A target patch j' is feasible if the transportation area is larger than a threshold $\mathbf{P}_{i,j'} \ge \epsilon$. We expand $\hat{j}$ to collect more feasible target patches j' with a 4-connected flood fill algorithm. The collection of j' presents irregular shapes, so we collect target patches inside its axis-aligned bounding box as the final corresponding target patches B_i. The corresponding position $\hat{\mathbf{p}}_i$ is the expectation over B_i:

$$\begin{aligned} w_{i,j} &= \sqrt{\frac{P_{i,j}}{a_j}}, w_i = \sum_{j \in B_i} w_{i,j} \\ \hat{\mathbf{p}}_i &= \sum_{j \in B_i} \frac{w_{i,j}}{w_i} \cdot \mathbf{p}_j, \quad \forall i \in \mathcal{S}. \end{aligned} \tag{4}$$

Considering the area quadratically increases with the length of side, we use the root of areas as weights. $(\mathbf{p}_i, \hat{\mathbf{p}}_i)$ constitutes a match and will be refined during patch subdivisions.



(a) Crop and subdivide (b) Area transportation (c) Sub-patch trimming

Figure 4. **Sub-patches Trimming.** a) The windows of neighboring source patches are partially overlapped due to the expansion. b) After subdivision, the sub-patches at the overlapped locations are redundant. c) We reserve the sub-patches that send the largest effective area $\sum_{j\in B_k} P_{k,j}$ as patches for the next level.

3.3. Patch Subdivision

Generally, the error of the target position is proportional to the size of the patch, but the smaller size of patches indicates more patches, which leads to unacceptable computations and memory in transformers. To this end, we start with a large patch size and adopt the coarse-to-fine mechanism. With $\{(\mathbf{p}_i, \hat{\mathbf{p}}_i, B_i)|i \in \mathcal{S}\}$ obtained by the patch transportation at level l, we crop a local window around each source and target position, resize the target window to the size of the source window, and subdivide the windows into smaller patches for the next level. To ensure the cropped source-target window pairs share the same image content, we need to carefully compute their window size. The window size for a source patch is naturally defined by its patch size, but how to determine the window size for its corresponding target position is not easy. Fortunately, we can compute the scaling factor γ_i from the source position to the target position via the obtained areas. With the scaling factor, we can determine the target window size from the source window size. For convenience, we omit the superscript l of the variables at the l level.

Patch Cropping and Subdivision. Given the matched patch $(\mathbf{p}_i, \hat{\mathbf{p}}_i)$, we estimate their scaling factor γ_i and crop different-sized local windows around them with respect to the scale differences. Specifically, the source window size e is fixed, determined by the patch size s at this level. To determine the target window size of a source patch, we first compute the area expectation of the target position:

$$\hat{a}_i = \sum_{j\in B_i} \frac{P_{i,j}}{\sum_{j\in B_i} P_{i,j}} a_j. \tag{5}$$

Note that $j \in B_i$ are the corresponding target patches we find for the source patch i, so $\sum_{j\in B_i} P_{i,j}$ indicates the area that we successfully transport to the target image. $\hat{a}_i$ indicates the area of the target window received from the source patch. We thus are able to compute the scaling factor by: $\gamma_i = \sqrt{a_i}/\sqrt{\hat{a}_i}$ and then compute the target window size $\hat{e}_i = \gamma_i e$ with high confidence that the target window covers what the source window sees. A choice for the source window size e is to let it equal the patch size s, which means directly cropping the source patch. However, potential errors during patch matching and the regular shape of patches may lead to missing the matching near the window boundary. To this end, we choose to crop a larger window with the size $e = ns$ with an expansion factor n. The expansion operation increases the matching recall with the cost of bringing patch overlapping, which needs to be trimmed in the following step. After that, we resize the cropped target window to the source window size e, which narrows the scale differences and alleviates the appearance distortion. As shown in the fourth part in the Fig. 3, the source windows and resized target windows are finally subdivided into sub-patches with a smaller patch size s^{l+1}, so that each window produce $K \times K$ sub-patches ($K = e/s^{l+1}$).

Sub-patch Trimming with Area Transportation We perform sub-patch trimming as shown in Fig. 4. Note that the total number of sub-patches ($N \times K \times K$) in the source image is larger than directly dividing the image by the patch size s^{l+1} ($H/s^{l+1} \times W/s^{l+1}$) due to overlapping brought by the expansion operation. As we find correspondence for each source patch, the overlapped source sub-patches cause ambiguous correspondence at the next level. We thus need to trim the sub-patches $k \in \{1, ..., N \times K \times K\}$ to obtain non-repeated patches for the next level $i^{l+1} \in \{1, ..., H/s^{l+1} \times W/s^{l+1}\}$. Here comes the question: suppose $r_{i^{l+1}}$ contains the sub-patches that are repeated on the patch for the next level i^{l+1}, which sub-patch $k \in r_{i^{l+1}}$ worth keeping? Recall that $\sum_{j\in B_i} P_{i,j}$ indicates the area that we successfully transport from the source patch i to the target image. Intuitively, the area of a patch transported to the target image is small indicates the patch loses its target with a high probability, so we can regard the total transport area as confidence and keep the sub-patch owning the largest transported area. Specifically, inside each window pair, we extract the features for sub-patches and apply the patch area transportation to compute the transportation matrix as introduced in Sec. 3.1 and Sec. 3.2, so that we obtained corresponding $\mathbf{f}_k^{l+1}, B_k^{l+1}$ for source sub-patches and $\mathbf{P}^{l+1}$. Here, we slightly abuse $j \in 1, ..., N \times K \times K$ to denote the index of sub-patches in target windows. Suppose the sub-patches repeated in i^{l+1} are collected in $r_{i^{l+1}}$, we keep the sub-patch that has the maximal area:

$$\hat{k} = \underset{k\in r_{i^{l+1}}}{\arg\max} \sum\nolimits_{j\in B_k} P_{k,j}. \tag{6}$$

After sub-patch trimming, we get smaller patches, and each source patch only focuses on a small group of target patches. We can now enter the next level to refine the correspondence.

3.4. Supervision

We collect the ground truth correspondences for supervision following LoFTR [49]. Given a pair of images with their depth map and camera poses, we warp a pixel at $\mathbf{p}_i$ to the target image via the camera pose and its depth as the ground truth correspondence $\tilde{\mathbf{p}}_i$. We supervise our neural network with three losses: an outlier patch loss L_o, an inlier patch loss L_i, and a patch concentration loss L_c. The estimated position $\hat{\mathbf{p}}_i$ is the expectation of patches inside the bounding box. However, the ground truth position $\tilde{\mathbf{p}}_i$ maybe outside the bounding box at the beginning. Directly supervising $\hat{\mathbf{p}}_i$ with $\tilde{\mathbf{p}}_i$ provides meaningless gradients, and the neural network can not achieve convergence. We thus divide our source patches into outlier patches $\mathcal{M}_o$ and inlier patches $\mathcal{M}_i$ according to the distance error:

$$\begin{aligned} \mathcal{M}_o &= \{(i,j)|\ ||\hat{\mathbf{p}}_i - \tilde{\mathbf{p}}_i||^2 > \theta\}, \\ \mathcal{M}_i &= \{(i,j)|\ ||\hat{\mathbf{p}}_i - \tilde{\mathbf{p}}_i||^2 \leq \theta\}. \end{aligned} \tag{7}$$

Here, the index j indicates the target patch where the ground truth correspondence $\tilde{\mathbf{p}}_i$ locate inside. We apply the outlier patch loss for $i \in \mathcal{M}_o$ and the inlier patch loss with the patch concentration loss for $i \in \mathcal{M}_i$.

Outlier Patch Loss. For the patches belonging to the outlier set, we directly encourage the source patch to transport its area to the target patch by minimizing the negative log of the corresponding transportation mass:

$$L_o = -\frac{1}{|\mathcal{M}_o|} \sum_{(i,j)\in\mathcal{M}_o} log P_{i,j}. \tag{8}$$

This outlier patch loss effectively pulls the estimated target location into the corresponding target patch.

Inlier Patch Loss. The outlier patch loss can only achieve patch-wise accuracy. For inlier patches, we directly minimize the distance for fine-grained correspondence accuracy:

$$L_i = \frac{1}{|\mathcal{M}_i|} \sum_{(i,j)\in\mathcal{M}_i} ||\hat{\mathbf{p}}_j - \tilde{\mathbf{p}}_j||^2 \tag{9}$$

Patch Concentration Loss. One source patch transported to the target image is expected to be clustered together. In our PATS, the cluster is the target patches we collected in the bounding box, so the area transported from the source patch to the target image is expected to concentrate in the bounding box. Therefore, for the target patches outside the bounding box $j' \notin B_i$, we suppress the area they received:

$$L_c = \frac{1}{|\mathcal{M}_i|} \sum_{(i,j)\in\mathcal{M}_i, j'\notin B_i} P_{i,j'} \tag{10}$$

We use these three losses to supervise our model: $L = L_o + L_i + L_c$.

3.5. Implementation Details

We conduct the patch transportation and the subdivision alternatively for $L = 3$ times, with patch sizes of 32, 8, and 2 at each hierarchy, respectively. We train the network progressively by adding and training hierarchy one by one while freezing the weights of all previously trained hierarchies. The network is trained using AdamW [43] with an initial learning rate of 1e-4 and batch sizes of 128, 48, and 12 were set for each hierarchy. Each hierarchy converges after 30 hours of training with 3-4 NVIDIA RTX 3090 GPUs. We set the expansion factor n to 3 and 2 for the cropping in the first and second hierarchy. The threshold ϵ is set to 1e-5. We set θ to be equal to the patch size at each hierarchy.

Category	Method	Pose estimation AUC		
		@5°	@10°	@20°
Detector-based	RootSIFT [1]+SGMNet [6]	35.5	55.2	71.9
	SP [12]+SuperGlue [42]	38.7	59.1	75.8
Detector-free	DRC-Net [24]	29.5	50.1	66.8
	PDC-Net+ [58]	39.1	60.1	76.5
	LoFTR [49]	42.4	62.5	77.3
	ASpanFormer [7]	44.5	63.8	78.4
	Ours	**47.0**	**65.3**	**79.2**

Table 1. Evaluation on YFCC100M [54] for outdoor pose estimation.

Category	Method	Pose estimation AUC		
		@5°	@10°	@20°
Detector-based	SP [12]+OANet [29]	11.8	26.9	43.9
	SP [12]+SGMNet [6]	15.4	32.1	48.3
	SP [12]+SuperGlue [42]	16.2	33.8	51.8
Detector-free	DRC-Net [24]	7.7	17.9	30.5
	LoFTR [49]	22.1	40.8	57.6
	MatchFormer [60]	24.3	43.9	61.4
	QuadTree [52]	24.9	44.7	61.8
	ASpanFormer [7]	25.6	46.0	63.3
	Ours	**26.0**	**46.9**	**64.3**

Table 2. Evaluation on ScanNet [11] for indoor pose estimation.

Method	320	480	640	1024	1600
Pose estimation (AUC @5°)					
SP [12]+SuperGlue [42]	25.5	33.5	38.7	41.4	43.0
LoFTR [49]	35.9	51.6	54.2	52.9	22.2
ASpanFormer [7]	43.4	51.0	54.0	55.8	51.8
Ours	**48.5**	**58.5**	**61.1**	**61.1**	**57.2**
Matching coverage (%)					
SP [12]+SuperGlue [42]	13.1	20.2	23.7	22.0	16.1
LoFTR [49]	21.4	47.2	65.8	70.9	35.8
ASpanFormer [7]	24.7	50.0	68.4	83.7	62.9
Ours	**77.3**	**89.5**	**90.8**	**88.4**	**81.1**
Matching precision (%)					
SP [12]+SuperGlue [42]	57.9	67.9	67.4	63.0	59.3
LoFTR [49]	74.5	77.5	75.7	65.6	55.5
ASpanFormer [7]	77.7	78.1	76.6	69.3	63.1
Ours	**85.9**	**81.8**	**78.4**	**70.0**	**64.3**

Table 3. Evaluation on extreme-scale dataset. Compared with other methods, PATS presents strong robustness against scale variations.

4. Experiments

4.1. Relative Pose Estimation

We use ScanNet [11] and YFCC100M [54] to evaluate the effectiveness of our method for relative pose estimation

Method	DUC1	DUC2
	(0.25m, 10°) / (0.5m, 10°) / (1m, 10°)	
SP [12]+SuperGlue [42]	49.0/68.7/80.8	53.4/77.1/82.4
LoFTR [49]	47.5/72.2/84.0	54.2/74.8/**85.5**
ASpanFormer [7]	51.5/**73.7**/**86.0**	55.0/74.0/81.7
Ours	**55.6**/71.2/81.0	**58.8**/**80.9**/**85.5**

Table 4. Visual Localization on the InLoc benchmark [50].

Method	Day	Night
	(0.25m, 2°) / (0.5m, 5°) / (1m, 10°)	
Visual Localization of Aachen v1.0		
SP [12]+SuperGlue [42]	—	79.6/90.8/**100**
PDC-Net+ [58]	—	79.6/90.8/**100**
PoSFeat [23]	—	81.6/90.8/**100**
Ours	—	**85.7**/**94.9**/**100**
Visual Localization of Aachen v1.1		
SP [12]+SuperGlue [42]	**89.8**/**96.1**/**99.4**	77/90.6/**100.0**
LoFTR [49]	88.7/95.6/99.0	**78.5**/90.6/99.0
ASpanFormer [7]	89.4/95.6/99.0	77.5/91.6/99.5
Ours	89.6/95.8/99.3	73.8/**92.1**/99.5

Table 5. Visual Localization on the Aachen day-night benchmark [65].

Training Data	Method	KITTI-2012		KITTI-2015	
		APAE	Fl-all	APAE	Fl-all
C + T	PWC-Net [48]	4.14	20.28	10.35	33.67
	GLU-Net [57]	3.34	18.93	9.79	37.52
	RAFT [53]	—	—	5.04	17.8
	GLU+GOCor [56]	2.68	15.43	6.68	27.52
	FlowFormer [18]	—	—	4.09	14.72
M	PDC-Net+ [58]	1.76	6.6	4.53	12.62
	COTR + Intp. [20]	1.47	8.79	3.65	13.65
	ECO-TR + Intp. [51]	1.46	6.64	**3.16**	12.10
	Ours + Intp.	**1.17**	**4.04**	3.39	**9.68**

Table 6. Optical flow estimation on the KITTI [16] benchmark. C + T indicates FlyingChairs and FlyingThings, and M indicates Megadepth.

Ablation study	AUC@5°	Coverage	Precision
L=2 w/o area regression	50.7	49.8	68.4
L=2 w/o transportation	40.1	52.8	66.7
L=2 w/o concentration loss	41.6	32.1	75.9
L=2 w/o outlier loss	36.3	**98.4**	62.8
L=2 w/o spliting M	50.9	92.4	72.1
L=1	0.7	5.8	55.5
L=2	51.9	92.5	72
L=3 (full)	**61.1**	92.5	**78.4**

Table 7. Abaltion studies on the Megadepth [27] dataset.

in both indoor and outdoor scenes, respectively. Then to fully evaluate the performance under scale changes, we create an extreme-scale dataset by artificially scaling the images from MegaDepth [27] dataset.

Experimental setup. Following [42, 49], we train the indoor model of PATS on the ScanNet dataset, while training the outdoor model on MegaDepth. We report the pose accuracy in terms of AUC metric at multiple thresholds (5°,10°,20°). For the evaluation on the extreme-scale dataset, we show two additional metrics to analyze factors that contribute to accurate pose estimation, including matching precision [42] and matching coverage [44]. While the former metric measures the precision, the latter one describes how well the matched feature is distributed uniformly in the image.

Dataset. ScanNet is composed of 1613 indoor sequences. We follow the same training and testing split used by [42] , where 1.5K image pairs are used to evaluate. YFCC100M [54] contains a total of 100 million media objects. We evaluate on a subset of YFCC100M, which consists of 4 selected image collections of popular landmarks following [42, 49]. MegaDepth consists of one million Internet images of 196 outdoor scenes. We sample 1000 image pairs and manually scale the second image to five different resolutions from 320 to 1600 along the longer side, which make up our extreme-scale dataset.

Results. We compare PATS with both detector-based and detector-free methods. The detector-based methods have SuperPoint(SP) [12] as feature extractor. According to the results shown in Table 1 and Table 2, our method achieves state-of-the-art performance in both indoor and outdoor scenarios. Next, we evaluate our approach against three representative methods using an extreme-scale dataset. As presented in Table 3, LoFTR exhibits a significant decrease in performance under extreme scale change, while our method presents strong robustness (e.g., LoFTR 22.2 vs. our 57.2 under 1600 resolution in terms of AUC) ASpanFormer is an improved version of LoFTR, which still falls far short of our PATS at all resolutions. The evaluation in terms of matching accuracy and matching coverage demonstrates that our PATS obtains matches that are more accurate and uniformly distributed in the image. All of them contribute to the more accurate pose estimation. We show the qualitative result in Fig. 5.

4.2. Visual Localization

Visual localization aims to recover 6-DoF poses of query images concerning a 3D model. We use the InLoc [50] and Aachen Day-Night [65] datasets to validate the visual localization in indoor and outdoor scenes, respectively.

Experimental setup. Following [7, 42, 49], we use the localization toolbox HLoc [41] with the matches computed by PATS and evaluate on the LTVL benchmark [55]. The performance is measured by the percentage of queries localized within multiple error thresholds.

Dataset. The Aachen day-night v1.0 provides 4328 database images and 922 query images, and provides additional 2359 database images and 93 query images in v1.1. A great challenge is posed in matching texture-less or repetitive patterns with a huge difference in perspective.. Inloc contains 9972 RGB-D images geometrically registered to the floor maps. Great challenges are posed in identifying correspondences from images, in particular nighttime scenes, which occur in extremely large illumination changes.

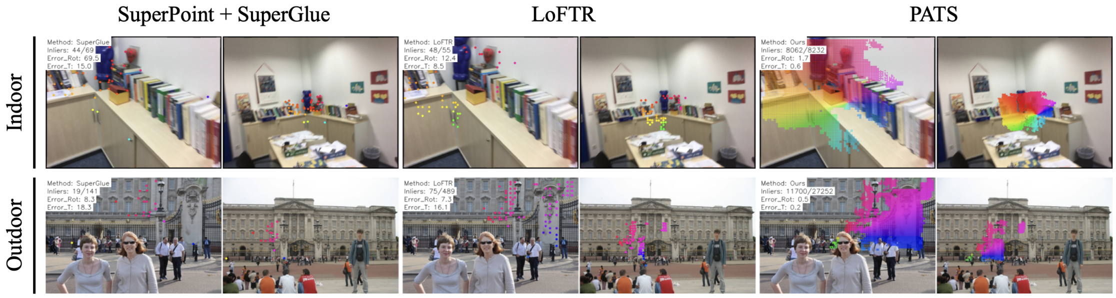


Figure 5. **Qualitative Comparison of Feature Matching.** The matched features are visualized as the same color. We have filtered incorrect matches. PATS shows superior performance on both accuracy and coverage.

Results. In Table 4, our method achieves the overall best results on InLoc dataset and outperforms other methods by a large margin in terms of percentage within the minimum error threshold (51.5 vs. our 55.6 in DUC1 and 55.0 vs. our 58.8 in DUC2). We report the results on Aachen day-nigh in Table 5 where our method achieves state-of-the-art performance in the challenging night track, especially on the v1.0 (81.6 vs. our 85.7 and 90.8 vs. our 94.9 in the first two metrics). On the day track of v1.1, our method outperforms all other methods but is slightly inferior to SuperGlue.

4.3. Optical Flow Estimation

We also evaluate our model on optical flow, which estimates per-pixel correspondences between an image pair.

Experimental setup. Our evaluation metrics include the Average End-point Error (AEPE) and Fl-all. Fl-all denotes the proportion of pixels for which the flow error exceeds either 3 pixels or 5% of the length of the ground truth flows. Following [20,51], we interpolate our model's output to obtain per-pixel correspondences ("+intp.") in Table 6.

Dataset. KITTI datasets are collected in urban traffic scenes. KITTI-2012 dataset contains simpler scenarios, while KITTI-2015 dataset has challenging and dynamic scenarios. Following [20, 57], we evaluate on the KITTI training sets without finetuning.

Results. We divide the methods into two categories for comparison. One is trained on the FlyingChairs [13] and FlyingThings [35], specially designed for optical flow, and the other is trained on MegaDepth. As shown in Table 6, our method obtains overall state-of-the-art performance. In terms of Fl-all metric, our method outperforms second-best method (ECO-TR) by a large margin in both KITTI-2012(+39%) and KITTI-2015(+20%).

4.4. Ablation Studies

To understand the effect of each design on the overall structure, we conduct the ablation study on MegaDepth dataset by disabling each design individually. In Table 7, we show the quantitative results. For the sake of the training time, we conduct most of the ablation studies only in the second hierarchy (denoted as $L = 2$), fixing the first hierarchy and removing the third one.

Removing the Area Regression. We re-train a model by removing the area regression (denoted as "w/o area regression"). The performance drop indicates that modeling the area relationship positively contribute to the model.

Removing Transportation. We also try removing the entire module of patch area transportation (denoted as "w/o transportation"). In this case when computing the corresponding position of each source patch, we compute the expectation based on the feature similarity instead of the transported area (Eq. 4), which is the similar to LoFTR. The performance degradation shows the importance of the proposed area transportation.

Impact of Each Loss Term. We study the impact of concentration loss and outlier patch loss by disabling them respectively and we also study the impact of dividing inlier and outlier patch (denoted as "w/o spliting M"). Note that the regression supervision in the inlier patch loss is the basic of our loss function so we do not remove them in all the ablation experiments. Overall, removing any design result in a decrease in the accuracy of the pose estimation.

Number of Hierarchies. Besides, we show the impact of increasing the number of hierarchies (denoted as "L=1,2,3", respectively). It can be seen that the performance keeps improving as the number of hierarchies increases.

5. Conclusion

In this paper, we propose Patch Area Transportation with Subidivion (PATS) for local feature matching. PATS learns to find the many-to-many relationship and scale differences through the proposed patch area transportation. Accompanied by the proposed patch subdivision algorithm, PATS enables extracting high-quality semi-dense matches even under severe scale variations. Multiple datasets demonstrate that PATS delivers superior performance in relative pose estimation, visual localization, and optical flow estimation, surpassing state-of-the-art results. One drawback of PATS is its inability to deliver real-time performance. However, it is promising to enhance our model and expanding the system to accommodate real-time applications such as SLAM.

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

References

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

[2] 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.

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

[4] Gabriele Moreno Berton, Carlo Masone, Valerio Paolicelli, and Barbara Caputo. Viewpoint invariant dense matching for visual geolocalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12169–12178, 2021.

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020.

[6] Hongkai Chen, Zixin Luo, Jiahui Zhang, Lei Zhou, Xuyang Bai, Zeyu Hu, Chiew-Lan Tai, and Long Quan. Learning to match features with seeded graph matching network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6301–6310, 2021.

[7] 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.

[8] Ying Chen, Dihe Huang, Shang Xu, Jianlin Liu, and Yong Liu. Guide local feature matching by overlap estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 365–373, 2022.

[9] Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1, 2016.

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

[11] 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.

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

[13] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Häusser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2758–2766, 2015.

[14] 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.

[15] Zheng Ge, Songtao Liu, Zeming Li, Osamu Yoshie, and Jian Sun. OTA: optimal transport assignment for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 303–312, 2021.

[16] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *Int. J. Robotics Res.*, 32(11):1231–1237, 2013.

[17] Zhaoyang Huang, Xiaokun Pan, Weihong Pan, Weikang Bian, Yan Xu, Ka Chun Cheung, Guofeng Zhang, and Hongsheng Li. Neuralmarker: A framework for learning general marker correspondence. *ACM Transactions on Graphics*, 41(6):1–10, 2022.

[18] Zhaoyang Huang, Xiaoyu Shi, Chao Zhang, Qiang Wang, Ka Chun Cheung, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Flowformer: A transformer architecture for optical flow. In *European Conference on Computer Vision*, 2022.

[19] 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.

[20] 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.

[21] Axel Barroso Laguna, Yannick Verdie, Benjamin Busam, and Krystian Mikolajczyk. Hdd-net: Hybrid detector descriptor with mutual interactive learning. In *Asian Conference on Computer*. Springer, 2020.

[22] 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.

[23] Kunhong Li, Longguang Wang, Li Liu, Qing Ran, Kai Xu, and Yulan Guo. Decoupling makes weakly supervised local feature better. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15838–15848, 2022.

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

[25] 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. *arXiv preprint arXiv:2303.07716*, 2023.

[26] 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.

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

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

[29] Huanyu Liu, Chao Peng, Changqian Yu, Jingbo Wang, Xu Liu, Gang Yu, and Wei Jiang. An end-to-end network for panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6172–6181, 2019.

[30] Haomin Liu, Guofeng Zhang, and Hujun Bao. Robust keyframe-based monocular slam for augmented reality. In *IEEE International Symposium on Mixed and Augmented Reality*, pages 1–10. IEEE, 2016.

[31] Weide Liu, Chi Zhang, Henghui Ding, Tzu-Yi Hung, and Guosheng Lin. Few-shot segmentation with optimal transport matching and message flow. *arXiv preprint arXiv:2108.08518*, 2021.

[32] Yanbin Liu, Linchao Zhu, Makoto Yamada, and Yi Yang. Semantic correspondence as an optimal transport problem. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4463–4472, 2020.

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

[34] 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.

[35] Nikolaus Mayer, Eddy Ilg, Philip Häusser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4040–4048, 2016.

[36] 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.

[37] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of International Conference on Machine Learning*, 2010.

[38] Udit Singh Parihar, Aniket Gujarathi, Kinal Mehta, Satyajit Tourani, Sourav Garg, Michael Milford, and K. Madhava Krishna. Rord: Rotation-robust descriptors and orthographic views for local feature matching. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1593–1600, 2021.

[39] 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.

[40] 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.

[41] 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.

[42] 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.

[43] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[44] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016.

[45] Xiaoyu Shi, Zhaoyang Huang, Weikang Bian, Dasong Li, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Videoflow: Exploiting temporal cues for multi-frame optical flow estimation. *arXiv preprint arXiv:2303.08340*, 2023.

[46] Xiaoyu Shi, Zhaoyang Huang, Dasong Li, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Flowformer++: Masked cost volume autoencoding for pretraining optical flow estimation. *arXiv preprint arXiv:2303.01237*, 2023.

[47] Zhengyu Su, Yalin Wang, Rui Shi, Wei Zeng, Jian Sun, Feng Luo, and Xianfeng Gu. Optimal mass transport for shape matching and comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(11):2246–2259, 2015.

[48] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8934–8943, 2018.

[49] 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.

[50] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomás Pajdla, and Akihiko Torii. Inloc: Indoor visual localization with dense matching and view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7199–7209, 2018.

[51] 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*. Springer, 2022.

[52] Shitao Tang, Jiahui Zhang, Siyu Zhu, and Ping Tan. Quadtree attention for vision transformers. In *The International*

Conference on Learning Representations. OpenReview.net, 2021.

[53] 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.

[54] 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.

[55] Carl Toft, Will Maddern, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Tomás Pajdla, Fredrik Kahl, and Torsten Sattler. Long-term visual localization revisited. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[56] Prune Truong, Martin Danelljan, Luc Van Gool, and Radu Timofte. Gocor: Bringing globally optimized correspondence volumes into your neural network. *Advances in Neural Information Processing Systems*, pages 14278–14290, 2020.

[57] Prune Truong, Martin Danelljan, and Radu Timofte. Glunet: Global-local universal network for dense flow and correspondences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6258–6268, 2020.

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

[59] 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.

[60] Qing Wang, Jiaming Zhang, Kailun Yang, Kunyu Peng, and Rainer Stiefelhagen. Matchformer: Interleaving attention in transformers for feature matching. In Lei Wang, Juergen Gall, Tat-Jun Chin, Imari Sato, and Rama Chellappa, editors, *Proceedings of the Asian Conference on Computer Vision*, pages 256–273, 2022.

[61] 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.

[62] 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.

[63] 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.

[64] Guofeng Zhang, Haomin Liu, Zilong Dong, Jiaya Jia, Tien-Tsin Wong, and Hujun Bao. Efficient non-consecutive feature tracking for robust structure-from-motion. *IEEE Transactions on Image Processing*, 25(12):5957–5970, 2016.

[65] Zichao Zhang, Torsten Sattler, and Davide Scaramuzza. Reference pose generation for long-term visual localization via learned features and view synthesis. *Int. J. Comput. Vis.*, 129(4):821–844, 2021.

[66] Qi Zhao, Zhi Yang, and Hai Tao. Differential earth mover's distance with its applications to visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(2):274–287, 2008.