MS^{2}DG-Net: Progressive Correspondence Learning via Multiple Sparse Semantics Dynamic Graph

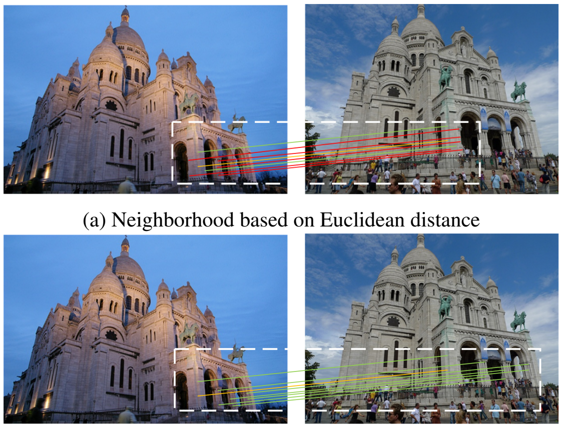
Luanyuan Dai[1] **Yizhang Liu**[2] **Jiayi Ma**[3]
Lifang Wei[1] **Taotao Lai**[4] **Changcai Yang**[1*] **Riqing Chen**[1]
[1] Fujian Agriculture and Forestry University
[2] Tongji University [3] Wuhan University [4]Minjiang University
{1191193002,riqing.chen}@fafu.edu.cn weilifang1981@163.com
{lyz8023lyp,jyma2010,laitaotao,changcaiyang}@gmail.com

Abstract

Establishing superior-quality correspondences in an image pair is pivotal to many subsequent computer vision tasks. Using Euclidean distance between correspondences to find neighbors and extract local information is a common strategy in previous works. However, most such works ignore similar sparse semantics information between two given images and cannot capture local topology among correspondences well. Therefore, to deal with the above problems, Multiple Sparse Semantics Dynamic Graph Network (MS^{2}DG-Net) is proposed, in this paper, to predict probabilities of correspondences as inliers and recover camera poses. MS^{2}DG-Net dynamically builds sparse semantics graphs based on sparse semantics similarity between two given images, to capture local topology among correspondences, while maintaining permutation-equivariant. Extensive experiments prove that MS^{2}DG-Net outperforms state-of-the-art methods in outlier removal and camera pose estimation tasks on the public datasets with heavy outliers. Source code:https://github.com/changcaiyang/MS2DG-Net

1. Introduction

Recently, finding high-quality correspondences has attracted broad attention in computer vision because of its wide applications, *e.g.*, visual localization [30], image fusion [17], virtual reality [34], Simultaneous Location and Mapping (SLAM) [21], Structure from Motion (SfM) [31], etc. However, the existing feature detection and description methods (SIFT [14], SuperPoint [6], ContextDesc [15], etc.) cannot provide significantly distinctive local features, which will lead to matching results ambiguity. Hence, the initial correspondence set, based on the nearest neighbor matching strategy, will inevitably have numerous incorrect correspondences (i.e., outliers).


(a) Neighborhood based on Euclidean distance

(b) Neighborhood based on sparse semantics similarity

Figure 1. The comparing illustration of (a) and (b), which are neighborhoods selected by Euclidean distance and sparse semantics similarity, respectively. The yellow, green and red lines represent the selected correspondences, inliers and outliers.

Outlier removal is an indispensable step to improve the correct correspondence (i.e., inlier) ratio of putative correspondences. Some traditional methods (RANSAC [7], VFC [19] and LPM [18]) are applicable to special scenes, but they may be unsuitable for the explosive growth of general datasets with an extremely low inlier ratio.

The permutation-equivariant deep learning-based outlier removal methods are data-driven, emerging as the times require, so they can capture the rich potential relationship among correspondences. Specifically, CNe [20], motivated by Point-Net [26], has adopted PointNet-like architectures to process each correspondence independently and predict

[*] Corresponding author

the probabilities of correspondences as inliers. Although it is a pioneering work and has achieved good performance, the network performance is adversely affected by ignoring the relationship among correspondences. Hence, local information has been introduced, in various ways, without losing permutation-equivariant, to capture the relationship among correspondences, in some networks, such as OA-Net [42], N^3-Net [25], NM-Net [43], ACNe [33] and so on.

Although the above networks perform well, the semantics information between two given images' sparse correspondences, which is called sparse semantics similarity in our paper, is not considered, and the local topology among correspondences can not capture well. Observing Figure 1, we can find that, for an image pair of the same scene, there are numerous outliers in the neighborhood based on Euclidean distance, while the neighborhood according to semantic similarity is more likely to be correct. In real life, humans match two images with naked eyes, without considering Euclidean distances of contents in images, but paying attention to similarity of them, that is, somewhere with similar semantics information. In Figure 1(a), neighbors are close to the selected correspondence in the Euclidean space, but without similar sparse semantics information, so they are more likely to be outliers. And in Figure 1(b), we can find that semantics similar structures, such as columnar structures of building archs are brought close together, although they are distant in the Euclidean space. By comparison, this phenomenon can be solved in Figure 1(b), but it will be ignored in Figure 1(a). Inspired by the human matching process, in this work, we use a graph neural network to construct dynamic graphs via sparse semantics similarity, to capture the local topology among correspondences. Specifically, we propose Multiple Sparse Semantics Dynamic Graph Network (MS^2DG-Net) to remove outliers while preserving permutation-equivariant, which greatly improves the matching effectiveness.

In this paper, MS^2DG-Net can dynamically adjust adjacency relationships between the selected correspondence feature map and residual ones in each graph. Meanwhile, it can also obtain multi-scale sparse semantics information, by concatenating sparse semantics information with different dimensions through a shortcut, which helps to prevent overfitting and capture richer contextual information.

Our contribution is twofold. Firstly, we find sparse correspondences may have sparse semantics and introduce a novel fashion by graph neural network to dynamically construct graphs based on sparse semantics similarity in the image pair and capture the local topology among correspondences. Secondly, we design Multiple Sparse Semantics Dynamic Graph Network (MS^2DG-Net), while maintaining permutation-equivariant, to build different sparse semantics graphs in different layers and fuse multi-scale information to obtain richer contextual information.

2. Related Work

2.1. Outlier Removal

Conventional outlier removal methods are divided into three categories, i.e., resampling-based, non-parametric model-based and relaxed algorithms in the literature [16]. RANSAC [7] is representative of resampling-based methods, which adopts a generation and verification strategy to solve the problem, and has many variants (MLESAC [36], MAGSAC [1] and so on). VFC [19], providing a new framework to address non-rigid matching, is a representative work of non-parametric model-based algorithms. GMS [2] and AdaLAM [3] adopt less strict geometric constraints to adapt to more complex scenes, *e.g*., wide baselines, which are called relaxed algorithms.

The handcrafted ones have made remarkable achievements in specific scenes, however, their performance is not satisfactory in public datasets with a large number of outliers. Hence, deep learning-based outlier removal is put on the agenda, where some networks (CNe [20] and DFE [28]) have proved that taking only correspondence coordinates as input is a feasible way to remove outliers. After that, some networks (ACNe [33] and LAGA-Net [5]) have introduced the attention mechanism to improve network performance. OA-Net [42] has inserted a clustering layer to capture some useful information, *e.g*., the underlying epipolar geometry. LFLN-Net [38] and NM-Net [43] have redefined neighborhood to find reliable correspondences. LMC-Net [13] and CL-Net [44] have added motion coherence and local-to-global consensus, respectively, to improve the performance of networks. COTR [10] and LoFTR [32] introduce the idea of Transformer [37] to improve the performance of networks. The above networks perform well. However, all of them neglect similar sparse semantics information in the image pair and local topology among correspondences can not capture well.

2.2. Graph Neural Network

Lately, the graph neural network has been applied in computer vision, due to its expressive capabilities. In [22], the correctness of answer reasoning has been improved by a graph neural network to construct multiple facts at the same time. Yang et al. [41] have proposed a framework, in the generated scene graph, using a graph neural network to update representations of target objects and relationships, and correct the prediction of the scene graph. Wang et al. [39] have introduced a new graph convolutional network, which can dynamically update the local features of point clouds, and has a good effect in point cloud segmentation and classification tasks. Liang et al. [12] have built a graph neural network between superpixels, in the segmentation task, to make better use of remote correlation. Chen et al. [4] have proposed an iterative visual reasoning system based on the graph knowledge, which can integrate spatial and semantics information. In this paper, we construct dynamic graphs, via a graph neural network, based on sparse semantics similarity in the image pair, to capture local topology among correspondences well. After that, we concatenate multi-scale sparse semantics information to capture more abundant contextual information, which has made great process in outlier removal and camera pose estimation tasks.

3. Proposed Method

3.1. Problem Formulation

Given a pair of images (I, I'), our goal is to remove outliers and recover camera poses. First, local features (SIFT [14], SuperPoint [6], etc.) can be used to detect keypoints and obtain descriptors. Then, the initial correspondence set C can be built by a nearest neighbor matching strategy:

$$C = \{c_1; c_2; ...; c_N\} \in \mathbb{R}^{N\times 4}, c_i = (x_i, y_i, u_i, v_i) \tag{1}$$

where c_i denotes a putative correspondence between two keypoints (x_i, y_i) and (u_i, v_i) in the image pair, both of which are normalized with camera intrinsic parameters.

Following OA-Net [42], we iteratively use MS^{2}DG-Net to produce the probability set $P' = \{p'_1; p'_2; ...; p'_N\}$ with $p'_i \in [0, 1)$, which indicates the probability of each correspondence c_i as an inlier. As shown in Figure 2, after the i^{th} MS^{2}DG-Net, both of the residual value set R_i and the probability set P_i as guidance information and the initial correspondence set C are concatenated, and the results are put into the $(i+1)^{th}$ MS^{2}DG-Net. To obtain the residual value set R_i, the probability set P_i is successively operated by the weighted eight-point algorithm and epipolar error calculation [8]. Finally, we get the final probability set P'. Following CNe [20], the weighted eight-point algorithm is leveraged to recover an estimate for the essential matrix $\hat{E}_m$, which can eliminate the negative effects of incorrect correspondences, so that it can regress a more accurate essential matrix than the traditional one [8]. In addition, the algorithm is differentiable for P', so the essential matrix can be estimated in an end-to-end fashion. The above operations can be written as:

$$R_i = ep(C, (g(C, P_i))) \tag{2}$$

$$O_i = \begin{cases} z_\psi(C), & i = 1 \\ z_\psi\left([C||R_{i-1}||P_{i-1}]\right), & i \geq 2 \end{cases} \tag{3}$$

$$P_i = pre(O_i) \tag{4}$$

$$\hat{E}_m = g(C, P') \tag{5}$$

where ep is the epipolar error calculation operation; $g(\cdot)$ denotes the weighted eight-point algorithm; $z_\psi\left(\cdot\right)$ is described as the permutation-equivariant MS^{2}DG-Net with its parameters ψ; $[\cdot||\cdot]$ presents a concatenation operation; pre is a prediction layer; P_i equals P', representing the final probability set, in the last iteration.

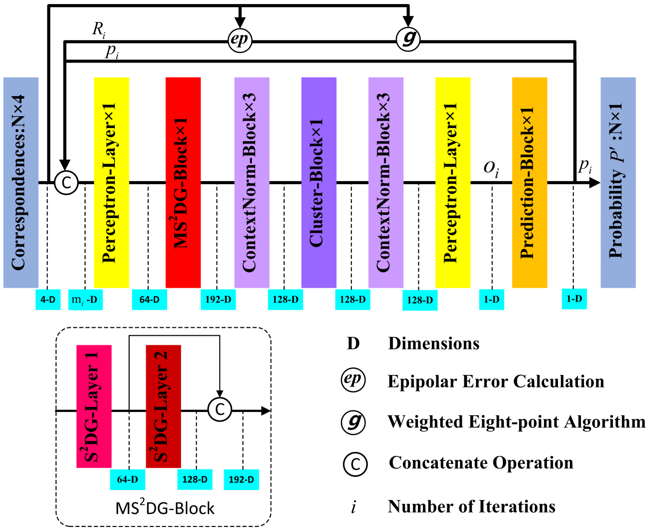


Figure 2. The Architecture of MS^{2}DG-Net. m_i represents the channel dimension of this position in the i-th iteration. When $i = 1$, $m_i = 4$, otherwise $m_i = 6$

3.2. Multiple Sparse Semantics Dynamic Graph

Build Sparse Semantics Dynamic Graph. Firstly, a correspondence c_i is encoded into a feature map $f_i = \{a_s\}, s = 1, 2, ...S$ by a multi-layer perceptron, where S is the dimension of the feature space. k-nearest neighbors of the selected feature map f_i are defined according to the sparse semantics similarity (inversely proportional to the sparse semantics distance d_{sem}) between f_i and each correspondence feature map $f_{ii'} = \{a'_s\}, i' = 1, 2, ...N$, in which the sparse semantics distance d_{sem} is denoted as:

$$d_{sem} = \left\| f_i - f_{ii'} \right\| \tag{6}$$

After selecting neighbors, we construct an edge set E_{ij} by concatenating the selected correspondence feature map and residual feature maps. And residual feature maps are obtained by subtracting the neighboring feature maps from the selected correspondence one, which can reduce the negative influence of absolute positions of neighboring feature maps. (See Table 3 for an ablation test.) So we choose this way to construct the edge set E_{ij}, and it can be defined as:

$$E_{ij} = [f_i || f_i - f_{ij}], j = 1, 2, ..., k \tag{7}$$

where $[\cdot||\cdot]$ presents concatenation; f_i, f_{ij} and $f_i - f_{ij}$ are the correspondence, neighborhood and residual feature maps, respectively.

Finally, a directed graph G is built for each feature map f_i with its k-nearest neighbors according to the sparse se-

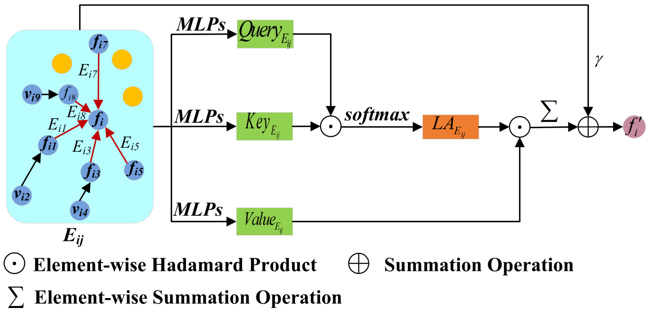


Figure 3. S^{2}DG-Layer 1.

mantics similarity, to capture the local topology, denoted as:

$$G_i = (V_{ij}, E_{ij}), i = 1, 2, ..., N, j = 1, 2, ..., k \qquad (8)$$

where G_i represents a directed graph of a feature map f_i; $V_{ij} = \{f_{i1}; f_{i2}; ...; f_{ik}\}$ indicates f_i's neighbors; E_{ij} denotes an oriented edge set.

Information Aggregation. We try three ways (maximum pooling, average pooling and Transformer [37]) to aggregate information. The first two indiscriminately treat each feature map, which ignore the different importance of each one. However, Transformer [37] can pay more attention to the potential inliers by the calculation of similarity. (See Table 5 for an ablation test.)

Therefore, we learn from Transformer [37] to aggregate dynamic graph information along the edge set E_{ij}. Figure 3 shows that we use $MLPs$ to map the edge set E_{ij} to $Query_{E_{ij}}$, $Key_{E_{ij}}$ and $Value_{E_{ij}}$, and then $Key_{E_{ij}}$ and $Query_{E_{ij}}$ are served to calculate the key-query similarity $LA_{E_{ij}}$. After that, we perform an element-wise Hadamard product between the key-query similarity $LA_{E_{ij}}$ and $Value_{E_{ij}}$, followed by an element-wise summation operation. To obtain a more robust feature map f_i', we integrate the feature map f_i, corresponding to the edge set E_{ij}, into the above results. These operations can be written as:

$$LA_{E_{ij}} = Softmax\left(Query_{E_{ij}} \odot Key_{E_{ij}}\right) \qquad (9)$$

$$f_i' = \sum_{j=1}^{k} \left(LA_{E_{ij}} \odot Value_{E_{ij}}\right) + \gamma f_i \qquad (10)$$

where $\odot$ and $Softmax(\cdot)$ are the element-wise Hadamard product and a softmax operation, respectively; $\sum$ represents the element-wise summation operation; γ is a learned hyper-parameter, which is initially set to 1, and then gradually learns an appropriate value.

Sparse Semantics Graph Rebuild. After each information aggregation, the sparse semantics similarity between feature maps has been updated. That is, while using the previous neighboring information, it may cause an irreparable error in subsequent calculations. Therefore, reconstructing graphs in each layer is necessary.

We build a new graph $^{(l+1)}G\left(^{(l+1)}V_{ij}, ^{(l+1)}E_{ij}\right)$ in the $(l+1)^{th}$ layer, where $^{(l+1)}V_{ij}$ is composed of the selected feature map f_i''s $k-$nearest neighbors according to the sparse semantics distance d_{sem} and $^{(l+1)}E_{ij}$ denotes the oriented edge set between $^{(l)}f_i'$ ($^{(l+1)}f_i$ and $^{(l)}f_i'$ represent the same spatial feature map in different layers) and elements in $^{(l+1)}V_{ij}$.

Multiple Sparse Semantics Dynamic Graph Fusion. Feature maps with different dimensions S contain different sparse semantics information, which can provide complementary information for each other. Hence, we concatenate different feature maps along their dimensions, so that feature maps can learn multi-scale sparse semantics information and obtain stronger representation abilities. However, Transformer [37] acquires that feature maps have the same dimension. But, if we do that, we will lose multiple sparse semantics information and Table 5 can support this point. We also try a trick (through $MLPs$) to transform feature maps in the l^{th} layer into the same dimension S as the $(l+1)^{th}$ layer. Comparison results can be seen from the 6th and 8th rows in Table 5, and the operation reduces the performance of MS^{2}DG-Net a lot. Hence, we choose the suboptimal way (maxpooling) to aggregate information in the next layer. Specifically, before a maxpooling operation, we use $MLPs$ for feature projection and dimension transformation. The above operations can be written as:

$$^{(l+1)}E_{ij} = [^{(l+1)}f_i||^{(l+1)}f_i - ^{(l+1)}f_{ij}], j = 1, 2, ..., k \qquad (11)$$

$$^{(l+1)}f_i' = Maxpooling(MLPs(^{(l+1)}E_{ij})) \qquad (12)$$

$$^{(l+1)}f_{iout} = [^{(l+1)}f_i'||^{(l)}f_i'], \ ^{(l+1)}S = 2 \times ^{(l)}S \qquad (13)$$

where $^{(l+1)}E_{ij}$ is the edge set in the $(l+1)^{th}$ layer; $^{(l+1)}f_i$ and $^{(l+1)}f_i - ^{(l+1)}f_{ij}$ are the selected and residual feature maps in the $(l+1)^{th}$ layer, respectively; $^{(l)}f_i'$ and $^{(l+1)}f_i'$ are the updated feature maps in the S^{2}DG Layer 1 and S^{2}DG Layer 2, respectively; $^{(l+1)}f_{iout}$ is the output of the MS^{2}DG Block; $^{(l)}S$ and $^{(l+1)}S$ are the feature map dimensions in the l^{th} and $(l+1)^{th}$ layer, respectively.

3.3. Property

Why Can We Directly Connect Edges. According to the Markov assumption [29], in an oriented graph, a random variable X is independent of its non-descendants given its parents. That is, in Fig. 3, f_{i1} is the parent feature map (node) of v_{i2}, and f_i is not the child feature map of v_{i2}, so that v_{i2} has no influence in feature map f_i as an inlier, due to the information of v_{i2} embedded in f_{i1}. Therefore, we only need to aggregate the edge features directly connecting to the feature map f_i and that is enough to get the local topology information of the whole graph.

3.4. Loss Function

Following CNe [20], a hybrid loss function is leveraged to optimize the proposed network:

$$L = L_c(P', G) + \beta L_e(E_m, \hat{E}_m) \tag{14}$$

where $L_c(\cdot, \cdot)$ presents a binary cross entropy loss function for the classification operation; P' and G are the predicted probability set and weakly supervised ground truth (labels), respectively; The later is chosen under the epipolar error [8] threshold of 10^{-4}; $L_e(\cdot, \cdot)$ is used to regress the essential matrix, where E_m is the ground truth essential matrix and $\hat{E}_m$ is the predicted essential matrix; β is a weight parameter to balance these two losses.

3.5. Implementation Details

As shown in Figure 2, MS^{2}DG-Net is composed of a Perceptron Layer, a MS^{2}DG Block, three ContextNorm Blocks, a Cluster Block, another three ContextNorm Blocks, another Perceptron Layer and a Prediction Block in order. Notably, the ContextNorm Block, including Perceptron, Context Normalization, Batch Normalization and ReLU activation function, is proposed in CNe [20]. The Cluster Block is introduced in OA-Net [42], and the Prediction Block consists of a tanh and a ReLU activation functions. MS^{2}DG-Net performs two iterations in total. In the first iteration, the first Perceptron Layer transforms the correspondence set $C \in \mathbb{R}^{N\times4}$ into a feature map set $F \in \mathbb{R}^{N\times64}$. After the MS^{2}DG Block, the dimension of the feature map set is changed from 64 to 192. After that, the feature map set F passes through the remaining network, and we can obtain the first residual value set $R_1 \in \mathbb{R}^{N\times1}$ and the first probability set $P_1 \in \mathbb{R}^{N\times1}$. Following OA-Net [42], we concatenate both of them as prior information to guide network learning. Then, the prior information and the initial correspondence set are put into the iteration network and we can obtain the final probability set P'.

The proposed network is trained on the PyTorch [24]. We follow the parameter setting of OA-Net [42]. Adam [23] optimizer with a learning rate of 10^{-3} is adopted. In addition, the batchsize of the input is 32. And the parameter β is set as 0 at the begining, and after $20k$ iterations, it is changed to 0.1. Experiments are implemented on Ubuntu 18.04 by NVIDIA GTX 3090 GPUs.

4. Experiments

In the section, we firstly introduce evaluation protocols. After that, to prove the effectiveness of MS^{2}DG-Net, we perform outlier removal and camera pose estimation tasks under outdoor and indoor scenes. Finally, we do ablation studies about the proposed operations in outdoor scenes.

4.1. Evaluation Protocols

In this section, datasets (outdoor and indoor scenes) and evaluation metrics will be orderly introduced.

Outdoor Scenes. Yahoo's YFCC100M dataset [35] as outdoor scenes, made up of 100 million images from the Internet, is splitted into 72 sequences in [9]. Following OA-Net [42], we choose 68 sequences as training sequences, and the remaining 4 sequences as unknown scenes to test all the methods. Moreover, [9] is used to generate the ground truth (labels) as well as recover camera poses.

Indoor Scenes. SUN3D [40] as indoor scenes, composed of a sequence of indoor RGB-D videos, is sampled in every ten frames to form a video picture dataset. We select 239 sequences as training sequences, and the rest 15 sequences as unknown scenes to test.

In this paper, all the networks are trained under the same setting. The training sequences are divided into three sets, including training (60%), validation(20%) and testing (20%), and the testing is picked up as known scenes. The images of known scenes are more like to images in training than unknown scenes.

Evaluation Metrics. Following OA-Net [42], we choose mAP5° as the default metric in the camera pose estimation task, and $Precision(P)$, $Recall(R)$ and $F\text{-}score(F)$ are chosen as metrics on the outlier removal task.

4.2. Baselines

We choose a traditional method (RANSAC [7]) and six learning-based networks (Point-Net++ [27], DFE [28], ACNe [33], CNe [20], OA-Net++ [42] and NM-Net [43]) as baselines. Following [42], 3D Euclidean space is instead of 4D to select the neighborhood in PointNet++ [27]. OA-Net++ [42] employs an iterative network, while OA-Net [42] not.

4.3. Outlier Removal

Outlier removal, a basic and important step, is significant for higher-level computer vision tasks. So we evaluate the proposed MS^{2}DG-Net and baselines on the outlier removal task. Table 1 presents the comparison results of outlier removal on outdoor and indoor datasets under known and unknown scenes. The quantitative results show that MS^{2}DG-Net surpasses the baselines in the outlier removal task on all evaluation metrics. Furthermore, compared with classical RANSAC, learning-based networks have a significant improvement of more than 10% on $F\text{-}score$. That is, the learning-based networks can effectively deal with outliers in the general datasets with an extremely low inlier ratio.

In addition, Figure 5 can support this point. Meanwhile we can find that visualization results of RANSAC, OA-Net++ and MS^{2}DG-Net are shown from top to bottom. Our MS^{2}DG-Net is obviously superior to other algorithms in different challenging scenarios.

Table 1. Quantitative comparative results of outlier removal on outdoor and indoor datasets with SIFT under known and unknown scenes. The bold result of each column represents the best result.

Datasets	Outdoor(%)						Indoor(%)					
Matcher	Known Scene			Unknown Scene			Known Scene			Unknown Scene		
	P	R	F	P	R	F	P	R	F	P	R	F
RANSAC [7]	47.35	52.39	49.74	43.55	50.65	46.83	51.87	56.27	53.98	44.87	48.82	46.76
Point-Net++ [27]	49.62	86.19	62.98	46.39	84.17	59.81	52.89	86.25	65.57	46.30	82.72	59.37
DFE [28]	56.72	87.16	68.72	54.00	85.56	66.21	53.96	87.23	66.68	46.18	84.01	59.60
ACNe [33]	60.02	88.99	71.69	55.62	85.47	67.39	54.11	88.46	67.15	46.16	84.01	59.58
CNe [20]	54.43	86.88	66.93	52.84	85.68	65.37	53.70	87.03	66.42	46.11	83.92	59.37
OA-Net++ [42]	60.03	89.31	71.80	55.78	85.93	67.65	54.30	88.54	67.32	46.15	84.36	59.66
NM-Net [43]	-	-	-	55.30	85.80	64.71	-	-	-	46.68	83.98	56.34
MS^{2}DG-Net	**63.17**	**90.98**	**74.57**	**59.11**	**88.4**	**70.85**	**54.50**	**88.63**	**67.50**	**46.95**	**84.55**	**60.37**

4.4. Camera Pose Estimation

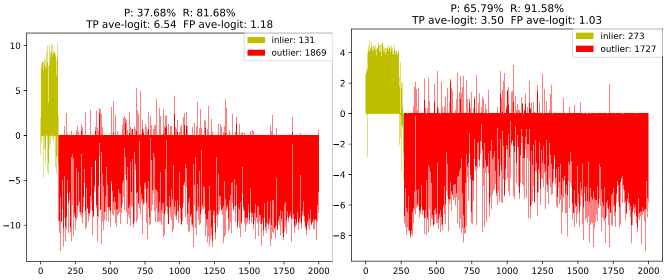
Recovering camera poses needs ample inliers, which puts forward a higher challenge to networks, and is also a basic task for follow-up computer vision tasks. This task requires not only a robust matcher, but also appropriate local features. Hence, we evaluate our MS^{2}DG-Net and baselines with different local features (SIFT [14] and SuperPoint [6]) under outdoor and indoor scenes.

Table 2 shows that our MS^{2}DG-Net significantly outstrips all the other comparison methods on all scenes with SIFT [14] or SuperPoint [6]. Specifically, compared to the second best network (OA-Net++ [42]), the parameters of OA-Net++ and our MS^{2}DG-Net are $2.47M$ and $2.61M$. Although ours has a litte more parameters, our MS^{2}DG-Net (with SIFT [14]) gains performance increasements of 5.79% and 10.18% for mAP5° without RANSAC under known and unknown scenes, respectively. Besides, with RANSAC post-processing, MS^{2}DG-Net also has a significant performance improvement on baselines in all scenarios. From Table 2, it is surprising to discover that the performance of all methods (except RANSAC) combined with SuperPoint [6] are lower than those combined with SIFT [14]. To explain this, we show partial typical visualization results of logit values by MS^{2}DG-Net on YFCC100M in Figure 4. Comparing with Figure 4 (a) and Figure 4 (b), we can find that although SuperPoint [6] has more inliers than SIFT [14], its average logit value is much lower than SIFT [14], which can explain why our network can be combined with SIFT [14] better.

Additionally, in Figure 5, partial typical visualization results of RANSAC, OA-Net++ and MS^{2}DG-Net are shown from top to bottom. Our MS^{2}DG-Net has achieved the best performance under different challenging scenes.

4.5. Ablation Studies

In the section, ablation studies about the number of neighbors and proposed operations on the outdoor dataset with SIFT are provided.



(a) YFCC100M(SIFT)
(b) YFCC100M(SuperPoint)

Figure 4. Partial typical visualization results of logit values of putative correspondence sets by our MS^{2}DG-Net on YFCC100M dataset. The red and yellow lines represent inliers and outliers, respectively. The length of the line is the size of the logit value passing our network. And if the logit value is greater than 0, it is used to calculate the probability of a correspondence as an inlier, otherwise it is considered as an outlier.

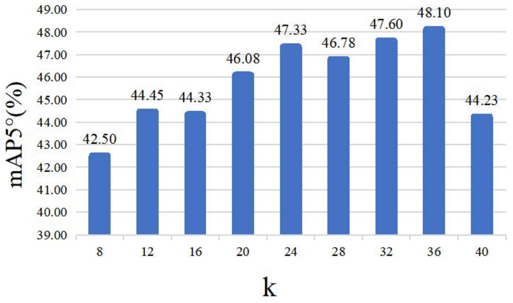


Figure 5. Parametric analysis of k on YFCC100M with SIFT.

How to select edges? We adopt two ways to construct an edge set E_{ij}, that is, the selected correspondence feature map is concatenated with neighboring feature maps or residual ones. For the former, the absolute positions of the neighboring feature maps will bring adverse effects to the

Table 2. Quantitative comparison results of camera pose estimation on the outdoor and indoor datasets. The mAP5°(%) without/with RANSAC as a post-processing step is reported.

Local Features	Matcher	Outdoor(%)		Indoor(%)	
		Known Scene	Unknown Scene	Known Scene	Unknown Scene
SIFT [14]	RANSAC [7]	-/5.81	-/9.07	-/4.52	-/2.84
	Point-Net++ [27]	10.49/33.78	16.48/46.25	10.58/19.17	8.10/15.29
	DFE [28]	19.13/36.46	30.27/51.16	14.05/21.32	12.06/16.26
	ACNe-Net [33]	29.17/40.32	33.06/50.89	18.86/22.12	14.12/16.99
	CNe [20]	13.81/34.55	23.95/48.03	11.55/20.60	9.30/16.40
	OA-Net++ [42]	32.57/41.53	38.95/52.59	20.86/22.31	16.18/17.18
	NM-Net [43]	-/-	32.93/51.90	-/-	14.13/16.86
	MS^{2}DG-Net	**38.36/45.34**	**49.13/57.68**	**22.20/23.00**	**17.84/17.79**
SuperPoint [6]	RANSAC [7]	-/12.85	-/17.47	-/14.93	-/12.15
	Point-Net++ [27]	11.87/28.46	17.95/38.83	11.40/21.19	9.38/17.08
	DFE [28]	18.79/31.72	29.13/43.00	13.35/22.57	12.04/17.41
	ACNe-Net [33]	26.72/31.16	32.98/45.34	18.35/21.12	13.82/18.05
	CNe [20]	12.18/30.25	24.25/42.57	12.63/21.81	10.68/17.36
	OA-Net++ [42]	29.52 /35.72	35.27/45.45	20.01/24.43	15.62/18.56
	MS^{2}DG-Net	**30.4/36.02**	**37.38/46.48**	**20.28/24.86**	**16.08/18.67**

Table 3. Evaluation of MS^{2}DG-Net(NF) and MS^{2}DG-Net(RF) on outdoor dataset with SIFT. mAP5° (%), P(%), R(%) and F(%) (under unknown scenes) are reported.

Method	Outdoor(%)				
	Known	Unknown	P	R	F
MS^{2}DG-Net(NF)	34.27/43.52	40.90/53.45	55.60	87.45	67.98
MS^{2}DG-Net(RF)	**37.96/45.12**	**46.08/56.60**	**57.28**	**87.69**	**69.30**

selected correspondence feature map. And from Table 3, we can find that the later performs better than the former in camera pose estimation and outlier removal tasks, so we choose the later.

How to choose k? Neighbor number k, deciding the amount of information in each dynamic graph, is pretty important. As shown in Figure 6, with the increase of k, the effect continues to improve, but after k = 20, the effect improves slowly and unstably. After comprehensive consideration, we choose k = 20 for subsequent operations.

Fixed Graph $vs.$ **Dynamic Graph.** Fixed graph: Establish a directed graph according to the Euclidean distance to capture the local geometric structure among correspondences. Dynamic graph: Build a dynamic graph based on sparse semantics similarity, to capture the local topology structure among feature maps (the dimension S = 64). Table 4 presents that MS^{2}DG-Net(FG) achieves a marginal improvement and even decreases comparing to OA-Net++, while MS^{2}DG-Net(DG) has a significant improvement. Due to initial correspondences with abundant outliers in the public dataset, the local geometry is not obvious or easy to capture. However, similar sparse semantics information among inliers can be used to efficiently capture the local topology information.

Table 4. Comparison results of MS^{2}DG-Net(FG) and MS^{2}DG-Net(DG) on the outdoor dataset with SIFT. mAP5° (%), P(%), R(%) and F(%) (under unknown scenes) are reported.

Method	Outdoor(%)				
	Known	Unknown	P	R	F
OA-Net++ [42]	34.04/42.06	38.95/51.65	55.27	87.00	67.60
MS^{2}DG-Net(FG)	32.51 /43.06	39.84 /54.40	56.56	86.99	68.55
MS^{2}DG-Net(DG)	**37.96 /45.12**	**46.08/56.60**	**57.28**	**87.69**	**69.30**

Table 5. Ablation studies on the outdoor dataset with SIFT under known and unknown scenes. mAP5°(%) without/with RANSAC are shown. 64, 128 and 256 are dimensions of feature maps.

max pooling	ave pooling	Transformer	Outdoor(%)	
			Known	Unknown
64			37.96 /45.12	46.08/56.60
	64		37.14/44.36	45.20/ 55.60
		64	38.58/45.16	47.43/ 57.50
		128	37.72/45.22	44.43/54.90
		64 & 64	35.97/44.88	45.05/56.18
		64 & 128	35.34/44.09	43.63/55.33
128 & 256		64	38.15/44.64	47.82/56.95
128		64	**38.36/45.34**	**49.13/57.68**

How to aggregate information? Three methods (maximum pooling, average pooling and Transformer [37]) are used to aggregate information along the edges. And Table 5 presents that Transformer [37] aggregation method performs best in all evaluation metrics and the reason has been explained above.

Is a wider dimension useful? From Table 5, comparing with the third and fourth lines, we can find that the model with the dimension S = 64 performs better. The model, whose dimension S = 128, has more parameters and may be overfitting.

Does multi-scale information help? In Table 5, compar-

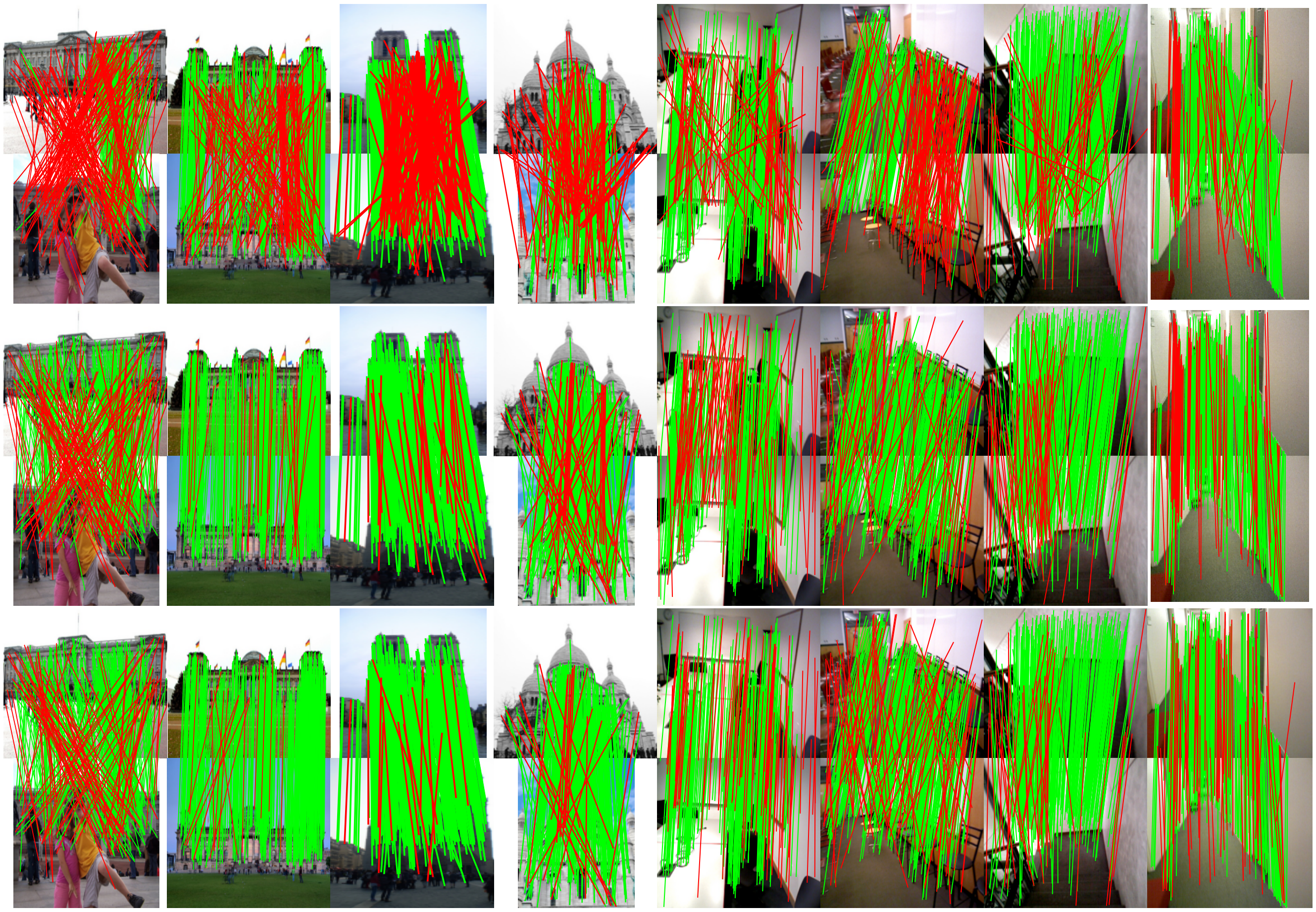


Figure 6. Partial typical visualization results on YFCC100M and SUN3D datasets with SIFT. From left to right: images of Buckingham-palace, Reichstag, Notre-dame-front-facade, Sacre-coeur, Te-mit1, Te-harvard1, Te-hotel1 and Te-brown1. From top to bottom: the results of RANSAC, OA-Net++ and our MS^{2}DG-Net. The green lines describe correct correspondences, the red lines otherwise.

ing with the third, fifth and last lines, we see that the model including two different dimensions performs best, and the model including two dynamic graphs with the same dimension performs even worse than the model with one dynamic graph, which can prove that multi-scale information is useful.

Is a larger model helpful? [11] has suggested that, in general, a network with two graph convolutional layers is enough. Meanwhile, comparing with the last and penultimate lines in Table 5 can support the above view. So a larger model may not provide help, but increase the number of parameters, so we choose a two-layer dynamic graph in our network.

5. Conclusion

In this work, motivated by the human matching process in the real world, we design MS^{2}DG-Net to find high-quality correspondences. Compared with operations based on the Euclidean space, our MS^{2}DG-Net employ a multi-scale dynamic graph to capture the local topology among correspondences based on the sparse semantics similarity in the image pair. Therefore, our MS^{2}DG-Net can learn to find correct correspondences according to sparse semantics information and local topology, which improves the performance of our network, while maintaining permutation-equivariant. Our experiments demonstrate that our MS^{2}DG-Net has clear superiority than several state-of-the-art methods in outlier removal and camera pose estimation tasks in publicly available datasets.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62171130, 62172197, 61802064, and 61501120, in part by the Natural Science Fund of Fujian Province under Grant 2019J01402 and 2020J01825.

References

[1] Daniel Barath, Jiri Matas, and Jana Noskova. Magsac: marginalizing sample consensus. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10197–10205, 2019. 2

[2] JiaWang Bian, Wen-Yan Lin, Yasuyuki Matsushita, Sai-Kit Yeung, Tan-Dat Nguyen, and Ming-Ming Cheng. Gms: Grid-based motion statistics for fast, ultra-robust feature correspondence. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4181–4190, 2017. 2

[3] Luca Cavalli, Viktor Larsson, Martin Ralf Oswald, Torsten Sattler, and Marc Pollefeys. Adalam: Revisiting handcrafted outlier detection. *arXiv preprint arXiv:2006.04250*, 2020. 2

[4] X. Chen, L. J. Li, L. Fei-Fei, and A. Gupta. Iterative visual reasoning beyond convolutions. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 2

[5] Luanyuan Dai, Xin Liu, Yizhang Liu, Changcai Yang, Lifang Wei, Yaohai Lin, and Riqing Chen. Enhancing two-view correspondence learning by local-global self-atention. *Neurocomputing*, 2021. 2

[6] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018. 1, 3, 6, 7

[7] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 1, 2, 5, 6, 7

[8] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition, 2004. 3, 5

[9] Jared Heinly, Johannes L Schonberger, Enrique Dunn, and Jan-Michael Frahm. Reconstructing the world* in six days*(as captured by the yahoo 100 million image dataset). In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3287–3295, 2015. 5

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

[11] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 7

[12] Xiaodan Liang, Xiaohui Shen, Jiashi Feng, Liang Lin, and Shuicheng Yan. Semantic object parsing with graph lstm. In *European Conference on Computer Vision*, pages 125–143. Springer, 2016. 2

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

[14] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 1, 3, 6, 7

[15] Zixin Luo, Tianwei Shen, Lei Zhou, Jiahui Zhang, Yao Yao, Shiwei Li, Tian Fang, and Long Quan. Contextdesc: Local descriptor augmentation with cross-modality context. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2527–2536, 2019. 1

[16] Jiayi Ma, Xingyu Jiang, Aoxiang Fan, Junjun Jiang, and Junchi Yan. Image matching from handcrafted to deep features: A survey. *International Journal of Computer Vision*, 129(1):23–79, 2021. 2

[17] Jiayi Ma, Yong Ma, and Chang Li. Infrared and visible image fusion methods and applications: A survey. *Information Fusion*, 45:153–178, 2019. 1

[18] Jiayi Ma, Ji Zhao, Junjun Jiang, Huabing Zhou, and Xiaojie Guo. Locality preserving matching. *International Journal of Computer Vision*, 127(5):512–531, 2019. 1

[19] Jiayi Ma, Ji Zhao, Jinwen Tian, Alan L Yuille, and Zhuowen Tu. Robust point matching via vector field consensus. *IEEE Transactions on Image Processing*, 23(4):1706–1721, 2014. 1, 2

[20] Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to find good correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2666–2674, 2018. 1, 2, 3, 5, 6, 7

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

[22] Medhini Narasimhan, Svetlana Lazebnik, and Alexander G Schwing. Out of the box: Reasoning with graph convolution nets for factual visual question answering. *arXiv preprint arXiv:1811.00538*, 2018. 2

[23] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 5

[24] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019. 5

[25] Tobias Plötz and Stefan Roth. Neural nearest neighbors networks. *Advances in Neural Information Processing Systems*, 31:1087–1098, 2018. 2

[26] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 1

[27] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30:5099–5108, 2017. 5, 6, 7

[28] René Ranftl and Vladlen Koltun. Deep fundamental matrix estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 284–299, 2018. 2, 5, 6, 7

[29] L Chris G Rogers and David Williams. *Diffusions, Markov processes and martingales: Volume 2, Itô calculus*, volume 2. Cambridge university press, 2000. 4

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

[31] 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. 1

[32] 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. 2

[33] Weiwei Sun, Wei Jiang, Eduard Trulls, Andrea Tagliasacchi, and Kwang Moo Yi. Acne: Attentive context normalization for robust permutation-equivariant learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11286–11295, 2020. 2, 5, 6, 7

[34] Richard Szeliski. Image mosaicing for tele-reality applications. In *Proceedings of 1994 IEEE Workshop on Applications of Computer Vision*, pages 44–53. IEEE, 1994. 1

[35] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016. 5

[36] Phil Torr and Andrew Zisserman. Robust computation and parametrization of multiple view relations. In *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*, pages 727–732. IEEE, 1998. 2

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 2, 4, 7

[38] Yang Wang, Xiaoguang Mei, Yong Ma, Jun Huang, Fan Fan, and Jiayi Ma. Learning to find reliable correspondences with local neighborhood consensus. *Neurocomputing*, 406:150–158, 2020. 2

[39] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019. 2

[40] Jianxiong Xiao, Andrew Owens, and Antonio Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *Proceedings of the IEEE international conference on computer vision*, pages 1625–1632, 2013. 5

[41] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graph generation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 670–685, 2018. 2

[42] Jiahui Zhang, Dawei Sun, Zixin Luo, Anbang Yao, Lei Zhou, Tianwei Shen, Yurong Chen, Long Quan, and Hongen Liao. Learning two-view correspondences and geometry using order-aware network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5845–5854, 2019. 2, 3, 5, 6, 7

[43] Chen Zhao, Zhiguo Cao, Chi Li, Xin Li, and Jiaqi Yang. Nm-net: Mining reliable neighbors for robust feature correspondences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 215–224, 2019. 2, 5, 6, 7

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