2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

LGCNet: Feature Enhancement and Consistency Learning Based on Local and Global Coherence Network for Correspondence Selection

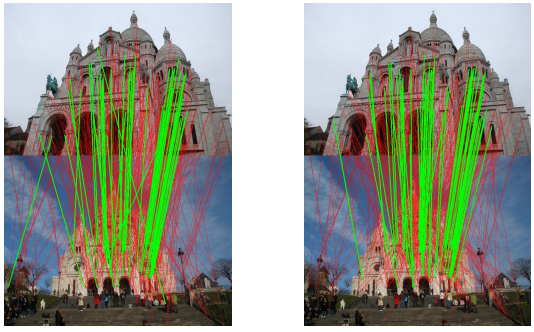
Tzu-Han Wu and Kuan-Wen Chen*

Abstract— **Correspondence selection, a crucial step in many computer vision tasks, aims to distinguish between inliers and outliers from putative correspondences. The coherence of correspondences is often used for predicting inlier probability, but it is difficult for neural networks to extract coherence contexts based only on quadruple coordinates. To overcome this difficulty, we propose enhancing the preliminary features using local and global handcrafted coherent characteristics before model learning, which strengthens the discrimination of each correspondence and guides the model to prune obvious outliers. Furthermore, to fully utilize local information, neighbors are searched in coordinate space as well as feature space. These two kinds of neighbors provide complementary and plentiful contexts for inlier probability prediction. Finally, a novel neighbor representation and a fusion architecture are proposed to retain detailed features. Experiments demonstrate that our method achieves state-of-the-art performance on relative camera pose estimation and correspondence selection metrics on the outdoor YFCC100M [1] and the indoor SUN3D [2] datasets.**

I. INTRODUCTION

Finding pixel-wise correspondences between a pair of images is a fundamental and crucial step in many computer vision tasks, such as relative camera pose estimation, Structure-from-Motion, and visual localization. Many feature-matching methods [3]–[8] focus on learning feature detectors and descriptors, but they are sensitive to extreme viewpoint and illumination changes because detectors and descriptors may differ considerably in these situations. Furthermore, there are usually numerous outliers among the correspondences established by these feature-matching methods, as a result of which downstream tasks cannot achieve good performance. Therefore, instead of learning task-specific feature detectors and descriptors, some approaches focus on learning matching criteria [9], [10] or correspondence selection mechanisms [11]–[14], which directly benefit downstream tasks.

The challenge of correspondence selection mechanisms is that models learn to distinguish between correct correspondences and false ones based only on the quadruple coordinates of correspondences. Furthermore, the order of correspondences is meaningless, so each correspondence is viewed as an independent datum. The lack of inherent neighbor information results in difficulties in establishing robust features. Consequently, many correspondence selection methods [12]–[14] define neighbors using their own criteria in order to capture the local coherence of correspondences, which is helpful information for inlier probability prediction and has been widely used. However, the ability of neural networks to learn coherence from quadruple coordinates is limited because coordinates have few attributes; furthermore, the putative data are usually extremely unbalanced. These limitations result in obvious misjudgements, such as false positive samples predicted by CLNet [13] in Fig. 1(a). Therefore, we propose Local and Global Coherence Network (LGCNet) to mitigate these concerns, as illustrated in Fig. 2.



(a) (b)

Fig. 1: Correspondence selection by (a) CLNet [13] and (b) our proposed LGCNet, where the green lines indicate the predicted inliers, and the red lines represent the predicted outliers. Some inliers in (a) are obvious false positive samples, while (b) shows more precise predictions.

LGCNet comprises three blocks. The first block, named Feature Enhancement Block, aims to relieve the burden of model learning when few attributes are available. It supplements additional attributes by using handcrafted characteristics at the beginning, guiding the model to prune obvious outliers from the perspectives of the local structures and the global tendencies. Some false positive samples identified by CLNet [13], shown in Fig. 1(a), are correctly predicted by LGCNet, as illustrated in Fig. 1(b). Feature Enhancement Block includes Local Structure Module (LSM) and Global Tendency Module (GTM) for better local and global coherence learning, respectively. The second block, named Local Consistency Block, aims to learn more informative contexts of local coherence. To capture plentiful neighbor contexts, a two-branch architecture, named Dual Criteria for Neighbors (DCN), is proposed. One branch searches k-nearest neighbors in coordinate space while the other does so in feature space. Complementary information can be acquired by these two kinds of neighbors, which can increase the robustness of features. Furthermore, Neighbor Auxiliary Module (NAM) is proposed to improve the neighbor representation and the aggregating mechanism adopted in [13], [15], so as to establish features in a more precise

Authors are with the Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu 300, Taiwan. (E-mail of corresponding author* Kuan-Wen Chen: kuanwen@cs.nycu.edu.tw)

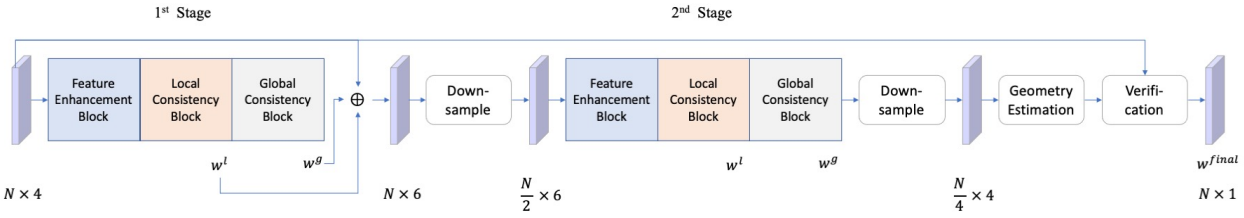


Fig. 2: Overview of LGCNet. Two-stage learning is applied with progressive pruning. Each stage predicts local and global inlier probabilities, i.e., w^l and w^g, and half of the correspondences having the higher global scores are selected for the second stage. In addition, the local and global scores are also regarded as enhanced features for the second stage.

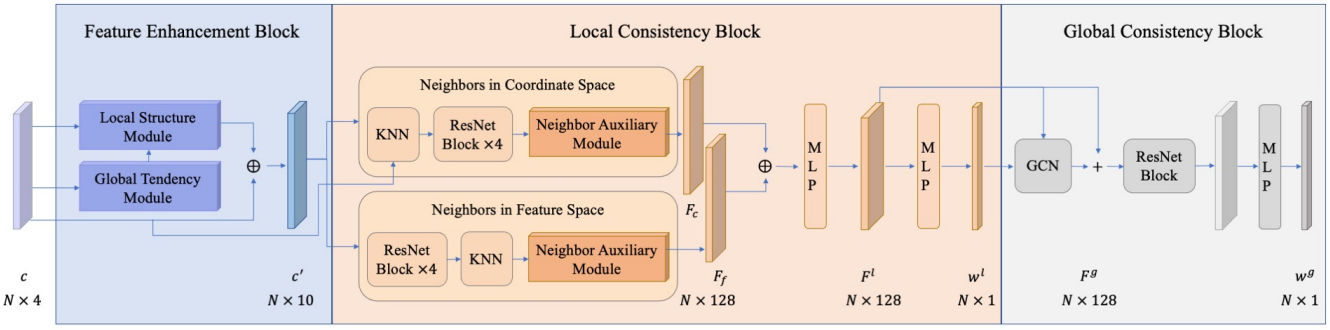


Fig. 3: Detailed architecture of one stage in LGCNet, which includes Feature Enhancement Block, Local Consistency Block, and Global Consistency Block. Inputs are the quadruple coordinates of correspondences; outputs are the predicted local and global inlier probabilities, i.e., w^l and w^g.

and informative manner. The third block is named Global Consistency Block. Following [13], each correspondence is regarded as a node on a graph, and its affinity with other nodes is represented by the predicted local probability of the correspondence. Then, Graph Convolutional Network (GCN) [16] is applied to learn global coherence.

Overall, our contributions are summarized as follows.

- We propose a novel correspondence selection network, LGCNet, which relies on LSM and GTM for feature enhancement to guide the model in pruning obvious outliers.
- DCN is proposed to leverage the neighbors in coordinate space as well as feature space, which can provide complementary information to benefit inlier probability prediction. In addition, NAM is proposed to represent features in a more precise and informative manner.
- LGCNet achieves state-of-the-art performance on the camera pose estimation and the correspondence selection task on YFCC100M [1] and SUN3D [2] datasets.

II. RELATED WORKS

Some handcrafted approaches utilize local coherence to distinguish between inliers and outliers. GMS [17] indicates that the neighbors of correct correspondences are likely to be coherent; conversely, those of false correspondences tend to be distributed randomly. Coherence is measured by the number of coherent neighbors in a small region. Recently, many learning-based correspondence selection approaches have flourished. They are based on point-cloud architectures because correspondences and point clouds have similar properties. PointNet [18] is a noted learning-based approach for point cloud classification and segmentation, but it processes each point individually, which results in limited generalizability to complex scenes. Therefore, many correspondence selection methods [12]–[15] follow a PointNet-like architecture but improve on capturing global information or defining local neighbors by using various criteria to construct discriminative features.

In terms of local coherence, PointNet++ [19] defines neighbors as all points within a radius. OANet [12] groups all correspondences into clusters, and the correspondences in each cluster are regarded as neighbors of each other. DGCNN [15] and CLNet [13] select k-nearest neighbors in feature space to capture semantic information. LMCNet [14] finds k-nearest neighbors in coordinate space to exploit local coherence. These approaches aim to learn local coherence by networks, but it is difficult to learn well when depending only on the quadruple coordinates. Therefore, we add handcrafted coherent contexts at the beginning, which guide the model toward learning discriminative features. In terms of global coherence, OANet [12] applies MLPs in the spatial dimension to capture global contexts. DGCNN [15] stacks their proposed EdgeConv blocks to learn global properties progressively. CLNet [13] regards correspondences as a connected graph and applies a GCN [16] to capture global information. LMCNet [14] finds global smooth motions

by fitting a smooth function, and it obtains a closed-form solution from graph Laplacian. In LGCNet, we add some handcrafted global contexts to filter out the correspondences whose tendencies substantially deviate from those of the others as a global indication.

III. METHOD

A. Problem Formulation and Overview

Given a pair of images (I, I'), keypoints and descriptors can be extracted by handcrafted [20]–[22] or learning [3], [4], [23] methods, and putative correspondences are established by nearest-neighbor search. There are still numerous outliers among these putative correspondences, so our method, LGC-Net, is designed to further distinguish between the correct correspondences and the false ones. We denote N putative correspondences as

$$C = [c_1, c_2, \dots, c_N], \quad c_i = (x_i, y_i, u_i, v_i), \tag{1}$$

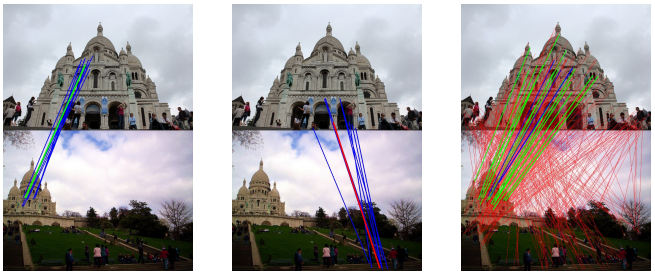
where c_i is the quadruple coordinate of a correspondence, and (x_i, y_i) and (u_i, v_i) are its coordinates in two images. Then, we propose a learning-based framework to accurately identify the inliers among the putative correspondence set C.

In this paper, four novel modules are proposed, and they are categorized into two blocks, as illustrated in Fig. 3. In Feature Enhancement Block, the dispersion score σ_i is measured by LSM, and the tendency score τ_i is estimated by GTM. The enhanced features are the quadruple coordinates concatenated with these two scores, denoted as $c_i' = [c_i, \sigma_i, \tau_i]$, such that they are more informative and discriminative for better model learning. In Local Consistency Block, features are first extracted by ResNet blocks [24], after which k-nearest neighbor search is applied in coordinate space and feature space in order to exploit neighbors of these two criteria for consistency learning. Neighbor information is aggregated through NAM. Finally, the features of DCN are fused, and the local inlier probability w^l is predicted by an MLP layer. Subsequently, in Global Consistency Block, an adjacent matrix is constructed from the predicted local inlier probabilities. A GCN [16], a ResNet block, and an MLP layer are applied to predict global inlier probabilities w^g as in [13]. The global inlier probabilities are regarded as the final scores because they are calculated from the local inlier probabilities by GCN.

Following [13], progressive pruning in two stages is adopted, i.e., some candidates having higher scores are selected in the first stage, and they are further pruned in the second stage. The scores of the surviving correspondences after two pruning stages are denoted as w^{final}, and they are applied in a weighted eight-point algorithm for recovering the essential matrix. Through this strategy, the model is able to alleviate the misjudgement of outliers and learn more detailed information.

B. Feature Enhancement Block

There are usually many coherent neighbors of correct correspondences in coordinate space, so the dispersion of the inliers and their neighbors is narrow, as illustrated in Fig. 4(a). The green line denotes a correct correspondence, and the blue lines indicate its neighbors. In contrast, false correspondences tend to be distributed randomly, so there are seldom coherent neighbors. This means that the trends of neighbors are likely to be different from the target correspondences such that the dispersion of the outliers and their neighbors is wide. In Fig. 4(b), the red line denotes a false correspondence, and the blue lines are its neighbors. Thus, the dispersion of neighbors is seen as a crucial clue for estimating inlier probability, so we introduce LSM to utilize this indication. K-nearest neighbor search is applied on the quadruple coordinates of correspondences, and the neighbors of a correspondence c_i are denoted as $[c_i^1, c_i^2, \dots, c_i^k]$. The dispersion score is formulated as



(a) (b) (c)

Fig. 4: Visualization of LSM and GTM. The green line in (a) denotes an inlier; the red line in (b) is an outlier; the blue lines are their neighbors. In (c), the green lines are inliers; the red lines are outliers; the blue lines denote the support bearings.

$$\sigma_i = \sqrt{\frac{\sum_{j=1}^{k} (c_i^j - \mu)^2}{k-1}}, \tag{2}$$

where

$$\mu = \frac{c_i + c_i^1 + \dots + c_i^k}{k+1}. \tag{3}$$

The dispersion score is calculated by the variance of distances from neighbors to the mean of the entire local structure.

Among the putative correspondences, the tendencies of some outliers are obviously different from those of others, so we hope to prune them with more certainty. Therefore, we propose GTM to predict the tendency score for each correspondence, which is a helpful attribute for the model to distinguish between inliers and outliers. The tendency of a correspondence is defined as $t_i = (u_i - x_i,\ v_i - y_i)$. The tendencies of correct correspondences across two images usually follow some specific bearings, and those of false correspondences are more likely to be distributed randomly, as the green and red lines respectively illustrate in Fig. 4(c). Blue lines denote the main tendencies among all correspondences. We exploit this characteristic for neural networks to easily remove outliers whose tendencies considerably deviate from the primary bearings. $B = [b_1, b_2, \dots, b_M]$ is a set of candidate bearings, where $b_i = (\cos(\theta * i), \sin(\theta * i))$ and $\theta = 2\pi/M$. M is the number of candidate bearings.

Each putative correspondence is classified into its closest bearing, and m bearings with most members are regarded as primary bearings, which represent the main tendencies of all correspondences. The correspondences belonging to these primary bearings are likely to be inliers. In order to obtain the tendencies of all correspondences more precisely, we compute the mean of all members belonging to each primary bearing, and the resulting vectors are named as support bearings, denoted as $[s_1, s_2, \ldots, s_m]$.

The tendency score is estimated by an MLP layer as follows:

$$\tau_i = \max_j MLP(t_i - s_j). \tag{4}$$

where t_i is the tendency of a correspondence, and s_j is a support bearing. We choose the maximum among the scores calculated by all the support bearings because the closest bearing is the most valuable for judging the outliers. If the tendency score of a correspondence is high, it is likely to be an outlier because its tendency is not coherent with any of the support bearings. Thus, the tendency score is helpful for removing outliers whose tendencies are substantially different from support bearings.

C. Local Consistency Block

Given that it is difficult to predict inlier probability using only a single correspondence, neighbors are usually regarded as important auxiliary information. z_i is the feature of correspondence c_i, and z_i^j is the feature of the j-th neighbor of c_i. The neighbor representation is commonly described as $e_i^j = [z_i, \Delta z_i^j] \in \mathbb{R}^{256}$, as in [13], [15], and the residual is $\Delta z_i^j = z_i - z_i^j$. To account for the target correspondence, c_i itself is also regarded as its own neighbor, but this leads to $\Delta z_i^j = 0$ when $j = 0$, because the residual of the target correspondence and itself is zero. Furthermore, the features in the first half dimensions, i.e., z_i, of all neighbors are the same, which results in redundant feature representation. In light of the two above-mentioned drawbacks, we propose NAM to form a compact neighbor representation $n_i^j \in \mathbb{R}^{128}$ as follows,

$$n_i^j = \begin{cases} z_i, & j = 0 \\ \Delta z_i^j, & j \neq 0 \end{cases} \tag{5}$$

NAM retains the features of the target correspondences but represents neighbors in a residual manner. In addition, position embedding $p_i^j \in \mathbb{R}^4$ is applied on the quadruple coordinates c_i^j. The representation of the position is the same as above.

$$p_i^j = \begin{cases} c_i, & j = 0 \\ \Delta c_i^j, & j \neq 0 \end{cases} \tag{6}$$

In order to combine e_i^j and p_i^j, they are processed by separate 2-layer convolutions to establish 128-dimensional features and are concatenated to form the integrated feature $f_i^j \in \mathbb{R}^{256}$.

$$f_i^j = [MLP(n_i^j),\ MLP(p_i^j)]\ , \tag{7}$$

(a) (b)

Fig. 5: Architectures of feature fusion by neighbors: (a) NAM in LGCNet (b) is Annular Convolution in CLNet [13].

The dimension of our final feature is the same as that of the common description in [13], [15], but our representation is more precise and informative because it involves all the original features and additionally includes position embedding.

It should be noted that the features of the target correspondence, i.e., f_i, and its neighbors have different meanings, so they should not be fused by a unified convolution layer, which treats them equally, such as Annular Convolution in [13]. Annular Convolution is illustrated in Fig. 5(b). Therefore, we propose the target-separated hierarchical convolution architecture, NAM, to properly fuse the neighbor information into the target feature, as illustrated in Fig. 5(a). In terms of neighbor fusion, small kernels are applied to fuse them hierarchically, so more details can be preserved. In terms of the target feature, another convolution layer is applied to obtain robust features. Finally, target features and neighbor features are combined using concatenation and a convolution layer. Through the target-separated hierarchical convolution architecture, primary features can be reserved by the target part, and auxiliary information can be captured by the neighbor part.

In addition, some works [14], [19] find k-nearest neighbors in coordinate space and focus on the variation in local structures. Some works [13], [15] find k-nearest neighbors in feature space and aim to capture coherent semantic features. We find that these two kinds of neighbors can provide complementary information, and both are crucial clues for predicting local inlier probability; therefore, the two-branch architecture, DCN, is proposed to fully utilize them. The first branch finds k-nearest neighbors in coordinate space, followed by four ResNet blocks [24] and our proposed NAM, to form the feature map F_c. Similarly, the second branch applies four ResNet blocks at the beginning and finds k-nearest neighbors on the current features. Then, NAM is adopted to form the feature map F_f. Finally, these two feature maps are combined by concatenating $F^l = [F_c, F_f]$ and are fused by an MLP layer. In a nutshell, more informative features are acquired by extracting neighbors using dual criteria, and they are beneficial to predicting local inlier probability $w^l \in \mathbb{R}^{N\times 1}$.

D. Global Consistency Block

Following [13], each correspondence is regarded as a node on a graph, and the weight of an edge between two nodes (c_i, c_j) is the product of their local inlier probabilities:

$$e_{i,j} = w_i^l * w_j^l\ . \tag{8}$$

The weights of the edges measure the affinities between every pair of correspondences, and they are aggre-

gated to establish an adjacent matrix $A \in \mathbb{R}^{N\times N}$ by $A_{ij} = e_{ij}$. $\tilde{A}_{ij} = A_{ij} + I$ retains robust transfers using self-connections. In GCN [16], a degree matrix is defined as $D_{ii} = \sum_i \tilde{A}_{ij}$, which refers to the number of neighbors for each node, and a graph Laplacian matrix is defined as $L = D^{-0.5}\tilde{A}D^{-0.5}$. The global embedding features are then obtained by $F^g = L\,F^l\,W^g$, where F^l is the local embedding features established by Local Consistency Block, and W^g is a trainable parameter matrix. Finally, the embedding feature F^g is encoded by a ResNet block, and then global inlier probability $w^g \in \mathbb{R}^{N\times 1}$ is predicted by an MLP layer.

E. Loss Function

To optimize the neural network, the overall training objective is denoted as

$$L = L_{cls} + \lambda L_{ess}\ , \tag{9}$$

where L_{cls} is a classification loss and L_{ess} is an essential matrix loss. λ is a weighting factor. The classification loss is the binary cross-entropy of the predicted and ground-truth labels for each correspondence. To alleviate the label ambiguity of correspondences whose epipolar distances are smaller than but close to the inlier threshold, their adaptive temperatures τ are given lower values. The classification loss is formulated as follows:

$$L_{cls} = \sum_{j=1}^{K} \ell_{bce}(\sigma(\tau^j * w^j), y^j)\ , \tag{10}$$

where τ is the adaptive temperature; w is the predicted inlier probability; y is the ground-truth label; and σ is a sigmoid function. As numerous inlier probabilities are predicted sequentially, we integrate and express them as K layers, and w_j represents local probabilities w^l, global probabilities w^g, and final probabilities w^{final}. [13] can be referred to for detailed formulations.

The essential matrix loss [12]–[14] computes the distances from ground-truth correspondences to the epipolar line, which is established by the predicted essential matrix. It is formulated as follows:

$$L_{ess} = \frac{(x_2^T \hat{E} x_1)^2}{\|\hat{E}x_1\|^2_{[1]} + \|\hat{E}x_1\|^2_{[2]} + \|\hat{E}^T x_2\|^2_{[1]} + \|\hat{E}^T x_2\|^2_{[2]}}\ , \tag{11}$$

where x_1 and x_2 are homogeneous coordinates of a ground-truth correspondence, and $\hat{E}$ is the predicted essential matrix.

IV. EXPERIMENTAL RESULTS

A. Evaluation Protocols

Implementation details. The outdoor YFCC100M [1] and the indoor SUN3D [2] datasets are used for training, and the train-test split follows [12], [13] for fair comparison. We apply SIFT [20] to extract keypoints and descriptors, and we establish putative correspondences using nearest-neighbor search. The inputs to the network are $N \times 4$, which represent the quadruple coordinate for each correspondence. N is the number of putative correspondences; typically, $N = 2000$.

TABLE I: Pose Estimation on Outdoor YFCC100M Dataset

Method	**AUC@5°**	**AUC@10°**	**AUC@20°**
RANSAC [25]	14.33	27.08	42.27
MAGSAC [26]	17.01	29.49	44.03
LPM [27]	10.22	20.65	33.96
GMS [17]	19.05	32.35	46.79
CODE [28]	16.99	30.23	43.85
PointCN [11]	26.53	43.93	61.01
ACNe [29]	28.81	48.02	65.39
NM-Net [30]	28.56	46.53	63.55
OANet [12]	28.76	48.42	66.18
SuperGlue [9]	30.49	51.29	69.72
CLNet [13]	32.79	52.70	69.76
LMCNet [14]	33.91	53.57	70.24
LGCNet (ours)	**34.98**	**54.84**	**71.54**

TABLE II: Pose Estimation on Indoor SUN3D Dataset

Method	AUC@5°	AUC@10°	AUC@20°
RANSAC [25]	3.93	10.28	21.04
MAGSAC [26]	3.94	10.33	21.25
LPM [27]	3.31	8.56	17.73
GMS [17]	4.36	11.08	21.68
CODE [28]	3.52	8.91	18.32
PointCN [11]	5.86	14.40	27.12
ACNe [29]	7.10	17.92	33.56
NM-Net [30]	6.45	16.44	31.16
OANet [12]	6.83	17.10	32.28
CLNet [13]	7.78	19.07	35.25
LMCNet [14]	6.77	17.14	32.55
LGCNet (ours)	**8.01**	**19.48**	**35.73**

TABLE III: Generalization Ability of LGCNet

Method	SUN3D [2]	YFCC100M [1]	
	SIFT [20]	ORB [22]	SuperPoint [3]
LMCNet [14]	5.38	5.84	17.66
CLNet [13]	5.64	7.95	19.12
LGCNet (ours)	**5.83**	**8.15**	**19.82**

Following [13], we adopt the two-stage pruning strategy, and half the correspondences are preserved at each stage. In the first stage, we set the number of neighbors to k=9, and to k=6 in the second stage. We use $M = 24$ bearings in GTM, i.e., the angle θ between each bearing is 15°. The five bearings having the most members are selected as the support bearings. A ResNet block [24] contains two MLP layers, Context Normalization [11], Batch Normalization [31], and ReLU.

Evaluation metrics. We evaluate the rotation and translation angular differences between the ground truth and our predicted camera pose. The pose error is defined as the larger of rotation and translation angular differences. The predicted camera pose is considered correct if the pose error is below a specific threshold. We report the AUCs at thresholds of 5°, 10°, and 20°.

B. Camera Pose Estimation

We compare LGCNet with handcrafted as well as learning-based methods. Handcrafted selection algorithms include RANSAC [25], MAGSAC [26], LPM [27], GMS [17], and CODE [28]. Learning-based selection networks include PointCN [11], ACNe [29], NM-Net [30], OANet [12], Su-

TABLE IV: Ablation Study on Each Module

CLNet [13]	LSM	GTM	NAM	DCN	AUC@5°
✓					32.79
✓	✓				33.82
✓		✓			33.26
✓			✓		33.58
✓				✓	34.07
✓	✓			✓	34.17
✓	✓		✓	✓	34.63
✓	✓	✓	✓	✓	34.98

perGlue [9], CLNet [13], and LMCNet [14].

These methods are evaluated on the YFCC100M [1] and the SUN3D [2] dataset, and the results are listed in Table I and Table II, respectively. The results indicate that LGCNet outperforms all the other methods, including handcrafted and learning-based approaches, on both datasets for all thresholds. On the YFCC100M dataset, LGCNet performs better than the state-of-the-art method, LMCNet [14], by approximately 1.21% on average. Compared with our baseline method CLNet [13], LGCNet gets about 2% improvement, proving that our proposed modules contribute to optimal performance by integrating informative contexts in Feature Enhancement Block and forming discriminative and representative features in Local Consistency Block. Section IV-C presents a more detailed analysis.

In addition, to evaluate the generalization ability of LGCNet, we train the model using SIFT on YFCC100M dataset and then test it on different datasets or different detector-descriptor approaches, i.e., SIFT [20] on SUN3D, ORB [22] on YFCC100M, and SuperPoint [3] on YFCC100M. The performances under an AUC threshold of 5° are reported in Table III. LGCNet achieves better performance than CLNet [13], indicating that LGCNet has good generalization and can be feasibly applied to different datasets and different detector-descriptor approaches.

C. Ablation Study and Module Generalizability

We evaluate our proposed modules individually, i.e., LSM, GTM, NAM, and DCN. Table IV reports the AUC at a 5° threshold for each module. The results indicate that each module improves the overall performance, and superposing some components results in increased improvement. LGCNet applies all four modules and achieves the best performance. Among these modules, DCN is the most crucial component and achieves the most improvement, indicating that neighbors in coordinate space and feature space can provide complementary and informative contexts.

To evaluate the generalizability of our proposed modules, we apply them on another model, LMCNet [14]. Based on original architecture of LMCNet, we supplement LSM, GTM, and DCN. LMCNet proposes Laplacian Motion Fitting to learn motion coherence, which is its main contribution, so we do not replace NAM for feature fusion. The data in Table V indicate that each of our modules is applicable and beneficial to LMCNet, and likewise, DCN results in the most improvement.

TABLE V: Module Generalizability

LMCNet [14]	LSM	GTM	DCN	AUC@5°
✓				33.91
✓	✓			34.32
✓		✓		34.09
✓			✓	34.94

TABLE VI: Precision, recall, F1-score, and the average number of predicted inliers on Correspondence Selection

Method	**Precision**	**Recall**	**F1-Score**	**# of Inliers**
LMCNet [14]	**87.29**	73.27	78.21	180.48
CLNet [13]	77.77	78.62	77.99	346.85
LGCNet (ours)	79.55	**80.45**	**79.84**	346.96

D. Correspondence Selection

To evaluate the accuracy of correspondence selection, we compare LGCNet with some baselines, i.e., LMCNet [14] and CLNet [13], on the YFCC100M [1] dataset. LMCNet outputs the predicted inlier probability for each correspondence. Following official setting of LMCNet, the correspondences whose inlier probabilities are higher than 0.95 are seen as inliers. However, CLNet [13] and our LGCNet adopt a progressive pruning strategy, so the outputs are the inlier probabilities for only the surviving correspondences. Therefore, following [13], CLNet and LGCNet apply a generation-verification pipeline to recover the inlier probabilities for all the input correspondences. Inliers are selected as those correspondences whose epipolar distances are less than 1e-3 as calculated by the predicted essential matrix. The precision, recall, F1-score, and the average number of inliers are reported in Table VI. LMCNet [14] directly predicts the inlier probability for each correspondence; consequently, it is easy to select a small subset that is full of inliers, which contributes to high precision. However, LGCNet and CLNet [13] recover inliers from the predicted essential matrix, which is relatively imprecise, so in general, loose thresholds are set to ensure that adequate inliers are chosen. Therefore, the precision for LGCNet and CLNet is inferior, but LGCNet achieves the best results on recall and the comprehensive metric F1-score.

V. CONCLUSION

LGCNet, having four novel modules, aims to enhance features and learn the local and global coherence. Among the four modules, LSM and GTM enhance preliminary attributes for better model learning, while NAM and DCN utilize complementary neighbors and precise representations to form informative features. LGCNet achieves state-of-the-art performance on camera pose estimation and the correspondence selection metrics on the outdoor YFCC100M [1] and the indoor SUN3D [2] datasets.

ACKNOWLEDGMENT

This work was supported in part by the National Science and Technology Council, Taiwan (111-2628-E-A49 -003 -MY2 and 111-2634-F-A49 -010-).

REFERENCES

[1] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li, "Yfcc100m: The new data in multimedia research," *Communications of the ACM*, vol. 59, no. 2, pp. 64–73, 2016.

[2] J. Xiao, A. Owens, and A. Torralba, "Sun3d: A database of big spaces reconstructed using sfm and object labels," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 1625–1632.

[3] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 224–236.

[4] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. 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*, 2019, pp. 8092–8101.

[5] J. Revaud, P. Weinzaepfel, C. De Souza, N. Pion, G. Csurka, Y. Cabon, and M. Humenberger, "R2d2: repeatable and reliable detector and descriptor," *arXiv preprint arXiv:1906.06195*, 2019.

[6] Z. Luo, L. Zhou, X. Bai, H. Chen, J. Zhang, Y. Yao, S. Li, T. Fang, and L. Quan, "Aslfeat: Learning local features of accurate shape and localization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 6589–6598.

[7] Y. Zhang, J. Wang, S. Xu, X. Liu, and X. Zhang, "Mlifeat: Multi-level information fusion based deep local features," in *Proceedings of the Asian Conference on Computer Vision*, 2020.

[8] M. Tyszkiewicz, P. Fua, and E. Trulls, "Disk: Learning local features with policy gradient," *Advances in Neural Information Processing Systems*, vol. 33, pp. 14 254–14 265, 2020.

[9] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 4938–4947.

[10] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, "Loftr: Detector-free local feature matching with transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 8922–8931.

[11] K. M. Yi, E. Trulls, Y. Ono, V. Lepetit, M. Salzmann, and P. Fua, "Learning to find good correspondences," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2666–2674.

[12] J. Zhang, D. Sun, Z. Luo, A. Yao, L. Zhou, T. Shen, Y. Chen, L. Quan, and H. Liao, "Learning two-view correspondences and geometry using order-aware network," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 5845–5854.

[13] C. Zhao, Y. Ge, F. Zhu, R. Zhao, H. Li, and M. Salzmann, "Progressive correspondence pruning by consensus learning," in *ICCV*, 2021.

[14] Y. Liu, L. Liu, C. Lin, Z. Dong, and W. Wang, "Learnable motion coherence for correspondence pruning," in *CVPR*, 2021.

[15] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *Acm Transactions On Graphics (tog)*, vol. 38, no. 5, pp. 1–12, 2019.

[16] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[17] J. Bian, W.-Y. Lin, Y. Matsushita, S.-K. Yeung, T.-D. Nguyen, and M.-M. Cheng, "Gms: Grid-based motion statistics for fast, ultra-robust feature correspondence," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4181–4190.

[18] C. R. Qi, H. Su, K. Mo, and L. 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*, 2017, pp. 652–660.

[19] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[20] P. C. Ng and S. Henikoff, "Sift: Predicting amino acid changes that affect protein function," *Nucleic acids research*, vol. 31, no. 13, pp. 3812–3814, 2003.

[21] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," in *European conference on computer vision*. Springer, 2006, pp. 404–417.

[22] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*. Ieee, 2011, pp. 2564–2571.

[23] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "Lift: Learned invariant feature transform," in *European conference on computer vision*. Springer, 2016, pp. 467–483.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[25] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[26] D. Barath, J. Matas, and J. Noskova, "Magsac: marginalizing sample consensus," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 197–10 205.

[27] J. Ma, J. Zhao, J. Jiang, H. Zhou, and X. Guo, "Locality preserving matching," *International Journal of Computer Vision*, vol. 127, no. 5, pp. 512–531, 2019.

[28] W.-Y. Lin, F. Wang, M.-M. Cheng, S.-K. Yeung, P. H. Torr, M. N. Do, and J. Lu, "Code: Coherence based decision boundaries for feature correspondence," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 1, pp. 34–47, 2017.

[29] W. Sun, W. Jiang, E. Trulls, A. Tagliasacchi, and K. M. Yi, "Acne: Attentive context normalization for robust permutation-equivariant learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 286–11 295.

[30] C. Zhao, Z. Cao, C. Li, X. Li, and J. Yang, "Nm-net: Mining reliable neighbors for robust feature correspondences," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 215–224.

[31] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.