Letter

MCNet: Multiscale Clustering Network for Two-View Geometry Learning and Feature Matching

Gang Wang and Yufei Chen

Dear Editor,

The main components of multi-view geometry and computer vision are robust pose estimation and feature matching. This letter discusses how to recover two-view geometry and match features between a pair of images, and presents MCNet (a multiscale clustering network) as an algorithm for extracting multiscale features. It can identify the true inliers from the established putative correspondences, where outliers may degenerate the geometry estimation. In particular, the proposed MCNet is based on graph clustering, in which the embedded correspondence features are mapped to a number of clusters by graph pooling. We designed a multiscale clustering layer into the two-view correspondence learning framework in order to improve correspondence representation efficiency. As a consequence of the multi-group feature fusion, we also constructed the network architectures termed MCNet-U and MCNet-M, respectively, utilizing the UNet and Pyramid techniques. Based on experimental results, the proposed model achieves state-of-the-art performance on feature matching with heavy outliers under weak supervision.

Related work: In the case of images with different views, a putative correspondence can be determined by local features like scale-invariant feature transform (SIFT) [1] and Superpoint [2], which always contain more outliers because of the ambiguity. As a result of the heavy outliers, matching-based tasks fail. A number of approaches have been developed to solve the geometry estimation and feature matching problems, from handcrafted matchers to deep neural networks. The RANSAC algorithm [3] and its variants are classical solutions in which the key idea is to obtain sampling consensus by hypothesizing and verifying. It is widely used for estimating multi-view geometry, but it is sensitive to outliers, especially heavy outliers in the scene. A prominent deep learning approach is the permutation-equivariant network, which operates on a single data point using a basic perceptron and extracts contextual information through global pooling. This can be achieved by simple normalization of the feature maps, a global procedure that is not influenced by the order. With weakly supervised learning, correspondence learning can be modeled to be a classification problem. PointCN [4] introduces context normalization (CN) and employs multilayer perceptrons (MLPs) to find reliable correspondences. CN can be viewed as an alternative to global feature pooling for PointNets [5], but with a different role: aggregating point feature maps and producing contextual information. DFE [6], N^{3}Net [7], OANet [8] and CAT-C [9] use PointCN as the baseline module to generate contextual information while retaining permutation equivariance. However, PointCN has two drawbacks: 1) It is hard to extract local context from correspondence owing to the MLPs being applied to each correspondence individually. 2) It encodes the global context through the mean and variance of features, which ignores potentially complex relationships between correspondences. This then leads to difficulty in adequately capturing the scene geometry encoded by inliers, especially in the case of heavy outliers. To this end, OANet [8] introduces an order-aware filtering module, named OAFilter, to integrate contextual and spatial correlations. ACNE [10] presents an attentive context normalization with training a perceptron to convert the intermediate feature maps to the corresponding attention weights. Besides, CAT-C [9] leverages self-attention in the PointCN to enhance the geometry from inliers and reject outliers using the extracted reliable context information.

Learning correspondence with classification and regression: Given a set of N putative correspondences $\mathbf{C}=[\boldsymbol{c}_1,\ldots,\boldsymbol{c}_N]\in\mathbb{R}^{N\times 4}$, which are constructed between two local key-points $\{(x_1,y_1)\}^N$,$\{(x_2,y_2)\}^N$, which are detected from a pair of images $(\boldsymbol{I}1,\boldsymbol{I}2)$, where $\mathbf{c}_i=(x_1^i,y_1^i,x_2^i,y_2^i)$ denotes a correspondence, and the key-points have been normalized by the known camera intrinsics. We aim to find the good correspondences (i.e., inliers $\mathbf{S}$) and estimate the geometry (i.e., an essential matrix $\mathbf{E}$). As formulated in [4], we construct a permutation-equivariant neural network $\mathbf{out}=f_\phi(\mathbf{C})$ with learnable parameters ϕ to learn the inlier probability $\mathbf{w}=[\mathbf{w}_1,\ldots,\mathbf{w}_N]=\tanh(\mathrm{ReLU}(\mathbf{out}))\in[0,1)$ under the weighted eight-point algorithm $g(\cdot,\cdot)$ [4] to identify inliers and get the regressed essential matrix $\hat{\mathbf{E}}=g(\mathbf{w},\mathbf{C})$, where $\mathbf{w}_i=0$ denotes an outlier, and the inlier weights can be easily obtained after operating tanh and ReLU activation functions. The overall training objective can be denoted as

$$\mathcal{L}=\alpha\mathcal{L}_{\mathrm{CLS}}(\mathbf{w},\mathbf{S})+\lambda\mathcal{L}_{\mathrm{REG}}(\hat{\mathbf{E}},\mathbf{E}) \tag{1}$$

where λ and α are weighting factors to balance the classification and regression. The classification term $\mathcal{L}_{\mathrm{CLS}}$ is a binary-cross entropy loss $\mathcal{L}_{\mathrm{CLS}}(\mathbf{w},\mathbf{S})=\frac{1}{N}\sum_{i=1}^{N}\gamma_i H(\mathbf{w}_i,\mathbf{S}_i)$ where H is the binary-cross entropy, and γ_i is the per-label weight to trade-off inlier and outlier examples. The regression term $\mathcal{L}_{\mathrm{REG}}$ represents a geometry loss on regressed $\hat{\mathbf{E}}$ and can be expressed as

$$\mathcal{L}_{\mathrm{REG}}=\frac{(\mathbf{p_2}\hat{\mathbf{E}}\mathbf{p_1})^2}{\|\mathbf{E}\mathbf{p_1}\|_{[1]}^2+\|\mathbf{E}\mathbf{p_1}\|_{[2]}^2+\|\mathbf{E}^T\mathbf{p_2}\|_{[1]}^2+\|\mathbf{E}^T\mathbf{p_2}\|_{[2]}^2} \tag{2}$$

where $\mathbf{p}_1$ and $\mathbf{p}_2$ denote a pair of components of one correspondence $\mathbf{c}=[\mathbf{p}_1,\mathbf{p}_2]$ between images $\boldsymbol{I}1$ and $\boldsymbol{I}2$ in homogenous coordinates. $\mathbf{t}_{[i]}$ denotes the i-th entry of vector $\mathbf{t}$.

Multiscale graph clustering network: In this letter, we propose a multiscale graph clustering network to capture more local-global context by using multi-cluster fusion instead of choosing a fixed cluster number. Following the order-aware paradigm, the proposed MCNet (Fig. 1) can be utilized to correspondence learning since it is a permutation-equivariant network in the guarantee. Our approach can deal with the unordered correspondences under permutation-equivariant. The multiscale cluster layer is plugged into the network after embedding with ResNet blocks, where each ResNet block consists of one CN layer, one Batch-Norm layer with ReLU, and one shared Perceptron layer as used in [4], [8] and [9]. Note that the first embedding layer is also one shared Perceptron layer to map the input from 4 (initialization) or 6 (information fusion) channels to 128 channels. Then, the following ResNet blocks and tanh(ReLU($\cdot$)) can map the $N\times C$ multiscale fusion output to $N\times 1$ weights.

Multiscale cluster layer: Given the embedded graph nodes, we can operate multiscale clustering with the Pooling-OAFilter-Unpooling pipeline [8]. The learned soft-assignments are used in differentiable pooling [11], which can be denoted as $\mathbf{A}_{\mathrm{down}}=\mathrm{softmax}(\mathrm{ResNetBlock}^{(N\rightarrow M)}(\mathbf{X}^{(l)}))\in\mathbb{R}^{N\times M}$. Then the feature at layer l is reduced to M clusters from N graph nodes: $\mathbf{X}^{(l+1)}=\mathbf{A}_{\mathrm{down}}^T\mathbf{X}^{(l)}$, where the M clusters can be defined as $[\frac{N}{2},\frac{N}{4},\frac{N}{8},\frac{N}{16}]$ in multiscale. For pre-

Corresponding author: Gang Wang.

Citation: G. Wang and Y. F. Chen, "MCNet: Multiscale clustering network for two-view geometry learning and feature matching," *IEEE/CAA J. Autom. Sinica*, vol. 10, no. 6, pp. 1507–1509, Jun. 2023.

G. Wang is with the Institute of Data Science and Statistics, the School of Statistics and Management, Shanghai University of Finance and Economics, Shanghai 200433, China (e-mail: gwang.cv@sufe.edu.cn).

Y. F. Chen is with the CAD Research Center, the College of Electronics and Information Engineering, Tongji University, Shanghai 201804, China (e-mail: yufeichen@tongji.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JAS.2023.123144

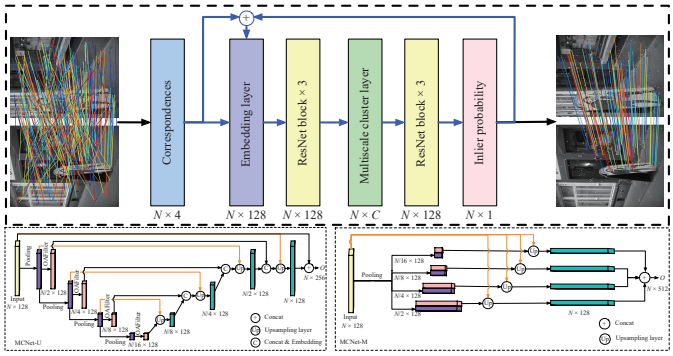

Fig. 1. The proposed multiscale clustering network (MCNet) architecture. Bottom left: MCNet-U. Bottom right: MCNet-M.

dicting the whole correspondences, the clusters should be upsampled to the entire graph nodes with a permutation-equivariant differential unpooling layer: $\hat{\mathbf{X}}^{(l)} = \mathbf{A}_{up}\hat{\mathbf{X}}^{(l+1)}$, where feature $\hat{\mathbf{X}}^{(l+1)} \in \mathbb{R}^{M\times D}$ is in a canonical order which denotes the new feature computed by $\mathbf{X}^{(l+1)}$ at the same layer $l+1$, and the feature $\hat{\mathbf{X}}^{(l+1)}$ can be transformed back to $\hat{\mathbf{X}}^{(l)} \in \mathbb{R}^{N\times D}$ at layer l with the learned upsampling soft-assignments $\mathbf{A}_{up} = \text{softmax}(\text{ResNetBlock}^{(M\to N)}(\mathbf{X}^{(l)})) \in \mathbb{R}^{M\times N}$, where each cluster in $\mathbf{X}^{(l)}$ corresponds to one row in $\mathbf{A}_{up}$, and the encoded the order information in $\mathbf{X}^{(l)}$ can be decoded to the previous step in the same way. In order to obtain the global context information, a spatial correlation layer is applied to a ResNetBlock to relate the cluster nodes [8] in OAFilter, where weight-sharing perceptrons are used on the spatial dimension to build relations between nodes. This is due to the fact that the cluster features are in a canonical order.

The exploited two styles of multiscale cluster layer are shown in Fig. 1, where MCNet-U follows [8] to obtain a large model with deep levels with UNet [12]. As discussed in [8], fusing two levels with two cluster scales $M = [500, 125]$ cannot improve the performance. Here, we use four levels with $M = [\frac{N}{2}, \frac{N}{4}, \frac{N}{8}, \frac{N}{16}]$ to redesign the model (see Fig. 1), and the output can be written as follows:

$$\mathbf{O}_{\text{MCNet-U}} = [\mathbf{X}, \hat{\mathbf{X}}]_M \in \mathbb{R}^{N\times 256} \tag{3}$$

where $[\cdot]_M$ denotes the multiscale concatation.

Besides, considering the multiscale features after the graph clustering, we construct a multi-layer pyramid model, called MCNet-M (see Fig. 1), instead of extracting features by progressive pipeline as used in MCNet-U. Precisely, the input $\mathbf{X} \in \mathbb{R}^{N\times 128}$ is mapped into four scales of clusters in MCNet-M with operating graph pooling, and the filtered features are transformed back to each new feature at the same level with six OAFilter blocks and one upsampling layer. Then, the new features $\{\hat{\mathbf{X}}_m\}_{m=1}^4 \in \mathbb{R}^{N\times 128}$ are concatenated to create an output

$$\mathbf{O}_{\text{MCNet-M}} = [\hat{\mathbf{X}}_1, \hat{\mathbf{X}}_2, \hat{\mathbf{X}}_3, \hat{\mathbf{X}}_4]_M \in \mathbf{R}^{N\times 512}. \tag{4}$$

MCNet-U requires the pooling output from the previous step. Experiments have demonstrated that MCNet-M is more effective than MCNet-U because of the ability to apply multi-head techniques to the feature representation. A fixed number of clusters cannot be adapted to practical applications because the inlier ratios of correspondence vary. Multiscale clustering MCNet-M captures the complex local and global contextual information to adapt to the inlier and outlier distribution of nodes in a graph.

Experimental setup: Yahoo's YFCC100M [13] and [14] outdoor scenes and the SUN3D [15] indoor scenes are used to evaluate the models. Following [8], the entire YFCC100M scenes are segmented into disjoint subsets for training (60%, 68 sequences), validation (20%) and testing (20%). Here, we retrain all models on outdoor scenes and test them on the indoor testing scenes (15 sequences) to evaluate the generalization ability. The ground-truth can be obtained by the generated camera pose via VisualSFM [16]. The putative correspondences are obtained by the SIFT and Superpoint local features and ratio test with setting $N = 2000$.

Results on geometry estimation: We use mean average precision (mAP) under 5°, 10° and 20° as the evaluation protocol for the camera pose estimation. The baselines contain PointCN, OANet++ and CAT-C, where OANet++ is trained with 6 OAFilter blocks while vanilla OANet uses one OAFilter block. There are two approaches to generating the essential matrix with the weighted eight-point algorithm and RANSAC. We also evaluate the performance using SIFT and Superpoint correspondences. Table 1 reports the quantitative results of the YFCC100M training (known scene) and test sets (unknown scene). All models can get more accuracy by post-processing with RANSAC. The proposed MCNet-M provides better performance than PointCN and OANet++ because it operates multiscale feature fusion. CAT-C applies self-attention to capture context information but needs more time and space consumption. MCNet-M outperforms CAT-C on the known scenes and is comparable to CAT-C in the unknown scenes. Table 2 reports the results on unknown indoor scenes, where MCNet gets the better generalization ability.

Table 1. Camera Pose Estimation Comparison Results on YFCC100M With or Without RANSAC (R)

		Model	mAP@5°		mAP@10°		mAP@20°	
			w/o R	w/ R	w/o R	w/ R	w/o R	w/ R
SIFT	Known	PointCN [4]	18.93	36.42	29.05	46.70	42.27	58.47
		OANet++ [8]	44.06	46.97	55.67	57.83	67.26	69.27
		CAT-C [9]	35.09	42.87	46.72	53.55	59.29	65.11
		MCNet-U	**52.09**	50.58	62.98	61.39	73.34	72.49
		MCNet-M	52.02	**50.59**	**63.16**	**61.39**	**73.67**	**72.45**
	Unknown	PointCN [4]	29.33	49.55	41.89	59.46	56.82	69.65
		OANet++ [8]	39.12	52.30	52.79	62.25	67.03	72.41
		CAT-C [9]	**44.52**	**53.90**	**57.54**	64.09	**69.99**	73.98
		MCNet-U	40.20	53.23	54.39	63.11	68.37	73.51
		MCNet-M	41.08	53.73	54.91	**64.09**	68.61	**74.17**
Super-point	Known	PointCN [4]	17.45	29.50	26.62	39.88	39.20	52.34
		OANet++ [8]	37.06	37.35	48.26	48.23	60.25	60.43
		CAT-C [9]	30.12	34.74	41.08	45.51	53.66	57.75
		MCNet-U	24.69	33.20	34.90	43.80	47.50	56.12
		MCNet-M	**44.35**	**40.60**	**55.25**	51.44	**66.33**	**63.30**
	Unknown	PointCN [4]	27.20	39.17	39.98	51.60	55.48	65.06
		OANet++ [8]	36.48	43.23	51.48	56.11	67.14	69.64
		CAT-C [9]	**37.00**	**44.35**	51.36	**56.94**	66.42	**69.86**
		MCNet-U	29.00	42.60	43.21	54.52	59.16	67.67
		MCNet-M	36.98	43.15	**52.50**	55.96	**68.03**	69.46

Table 2. Generalization Ability Evaluation on SUN3D Indoor Scenes

Model	SIFT			Superpoint		
	5°	10°	20°	5°	10°	20°
PointCN [4]	1.00	2.54	7.22	3.43	6.61	13.96
OANet++ [8]	3.41	6.95	10.92	5.30	9.78	18.96
CAT-C [9]	3.58	7.12	14.32	**5.72**	10.3	19.23
MCNet-U	**3.66**	**7.68**	**16.24**	4.07	8.21	16.66
MCNet-M	3.49	7.61	16.17	5.28	**10.3**	**20.28**

Results on feature matching: Table 3 reports the comparison results on YFCC100M with SIFT and Superpoint features. MCNet-M achieves the best average precision and F1-scores. OANet++ and CAT-C get better recall values. Our models get a slightly slower runtime than PointCN and OANet++, but faster than CAT-C. Fig. 2 shows the qualitative results of feature matching on unknown outdoor scenes, where MCNet-M provides more inliers, while RANSAC is sensitive with heavy outliers. Fig. 3 shows the cumulative distribution of inlier ratios, precision, recall, and F1-scores. The putative correspondences generated on the test set of YFCC100M have about 90% outliers, and deep learning models can establish more robust context information.

Impact on different scales: Table 4 presents an analysis of multiscale clustering (MCNet-U) using different scale parameters. Clusters are correlated with inliers as more information is lost when the scale is reduced. Multiscale $[\frac{N}{2}, \frac{N}{4}]$ is slightly less efficient than

Table 3. Feature Matching Results on YFCC100M With the Average Precision (Pr), Recall (Re), F1-Score (%), and Runtime (RT).

Model	SIFT				Superpoint			
	Pr	Re	F1	RT	Pr	Re	F1	RT
RANSAC [3]	51.25	44.22	46.38	0.057	58.14	58.18	57.32	0.064
MAGSAC [17]	53.57	58.67	54.59	0.039	58.35	73.49	63.58	0.035
LPM [18]	43.53	44.94	43.12	0.040	41.48	57.92	47.05	0.016
GLOF [19]	40.47	45.77	42.01	0.065	29.29	85.87	41.73	0.067
DFE [6]	50.79	84.67	60.62	0.042	60.10	87.62	68.64	0.019
N^3Net [7]	31.64	73.25	41.93	0.065	59.11	82.06	66.11	0.041
PointCN [4]	52.69	85.10	62.41	0.019	62.88	87.74	71.01	0.019
OANet++ [8]	56.05	82.78	64.71	0.028	65.81	87.48	73.13	0.029
CAT-C [9]	55.89	88.02	65.73	0.093	66.61	88.31	73.80	0.075
MCNet-U	56.38	83.85	65.17	0.042	64.41	87.15	71.84	0.041
MCNet-M	58.12	83.39	66.38	0.038	66.39	86.28	73.18	0.038

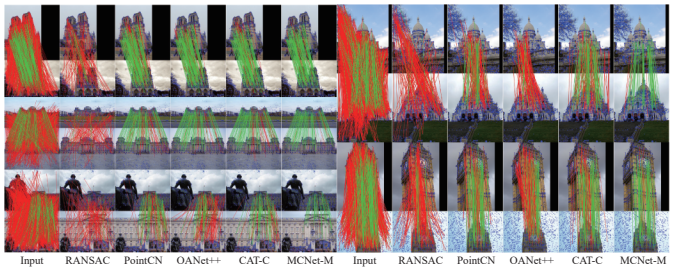

Fig. 2. Feature matching (using SIFT features) on YFCC100M.

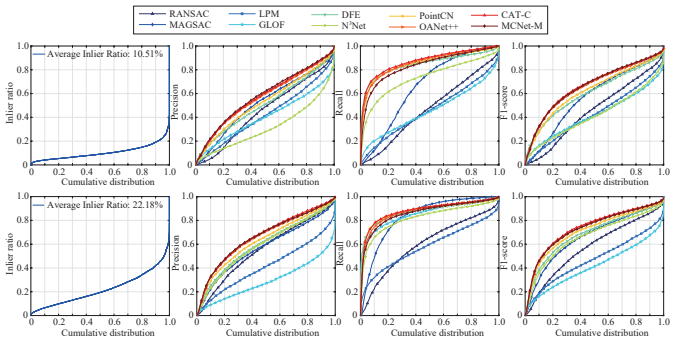

Fig. 3. Comparison results of feature matching using cumulative distribution of inlier ratio, precision, recall, and F1-score. Top: SIFT; Bottom: Superpoint.

Table 4. Analysis of Multiscale Clustering on YFCC100M and SIFT Features With Different Scales

$\frac{N}{2}$	$\frac{N}{4}$	$\frac{N}{8}$	$\frac{N}{16}$	mAP@**5°**		mAP@**10°**		mAP@**20°**	
				w/o R	w/ R	w/o R	w/ R	w/o R	w/ R
√	√			38.65	52.62	52.81	63.05	67.62	73.16
√	√	√		38.17	52.20	52.76	62.91	67.26	73.52
√	√	√	√	40.20	53.23	54.39	63.11	68.37	73.51

OANet++, but if all four scales are taken into consideration, it performs better. As a result of its pooling operations being not independent on each scale, MCNet-U performs unstably. MCNet-M has better performance with multi-head approach than UNet-like structures because it captures context independently from each scale.

Conclusion: Using classification and regression, we propose a multiscale graph clustering network that can deal with correspondence learning. Our approach entails the use of a multiscale cluster layer combined with a Pooling-OAFilter-Unpooling technique, so that local and global contextual information can be captured without setting a fixed cluster number. In order to achieve a better feature fusion with multiscale graph clustering, it is necessary to adjust the model to accommodate different inliers and outliers. As a result of analyzing the network with the UNet and Pyramid approaches, we recommend using a stable multi-scale graph clustering approach for correspondence learning, and it gets the state-of-the-art performance on challenging scenes.

Acknowledgments: This work was supported by the National Natural Science Foundation of China (61703260, 62173252).

References

[1] D. G. Lowe, " Distinctive image features from scale-invariant keypoints," *Int. J. Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[2] D. DeTone, T. Malisiewicz, and A. Rabinovich, " Superpoint: Self-supervised interest point detection and description," in *Proc. IEEE Conf. Computer Vision Pattern Recognition Workshops*, 2018, pp. 224–236.

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

[4] K. M. Yi, E. Trulls, Y. Ono, V. Lepetit, M. Salzmann, and P. Fua, "Learning to find good correspondences," in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, 2018, pp. 2666–2674.

[5] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, 2017, pp. 652–660.

[6] R. Ranftl and V. Koltun, " Deep fundamental matrix estimation," in *Proc. European Conf. Computer Vision*, 2018, pp. 284–299.

[7] T. Plötz and S. Roth, " Neural nearest neighbors networks," *Advances Neural Information Processing Syst*, vol. 31, pp. 1–12, 2018.

[8] 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 *Proc. IEEE Int. Conf. Computer Vision*, 2019, pp. 5845–5854.

[9] J. Ma, Y. Wang, A. Fan, G. Xiao, and R. Chen, "Correspondence attention transformer: A context-sensitive network for two-view correspondence learning," *IEEE Trans. Multimedia*, pp. 1–16, 2022. DOI: 10.1109/TMM.2022.3162115

[10] W. Sun, W. Jiang, E. Trulls, A. Tagliasacchi, and K. M. Yi, "ACNe: Attentive context normalization for robust permutation-equivariant learning," in *Proc. IEEE/CVF Conf. Computer Vision Pattern Recognition*, 2020, pp. 11286–11295.

[11] Z. Ying, J. You, C. Morris, X. Ren, W. Hamilton, and J. Leskovec, "Hierarchical graph representation learning with differentiable pooling," *Advances Neural Information Processing Systems*, vol. 31, pp. 1–11, 2018.

[12] O. Ronneberger, P. Fischer, and T. Brox, " U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Medical Image Computing Computer-Assisted Intervention*, Springer, 2015, pp. 234–241.

[13] 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 ACM*, vol. 59, no. 2, pp. 64–73, 2016.

[14] J. Heinly, J. L. Schonberger, E. Dunn, and J.-M. Frahm, "Reconstructing the world in six days," in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, 2015, pp. 3287–3295.

[15] J. Xiao, A. Owens, and A. Torralba, "Sun3D: A database of big spaces reconstructed using SFM and object labels," in *Proc. IEEE Int. Conf. Computer Vision*, 2013, pp. 1625–1632.

[16] C. Wu, " Towards linear-time incremental structure from motion," in *Proc. IEEE Int. Conf. 3D Vision*, 2013, pp. 127–134.

[17] D. Barath, J. Matas, and J. Noskova, " Magsac: Marginalizing sample consensus," in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, 2019, pp. 10197–10205.

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

[19] G. Wang and Y. Chen, " Robust feature matching using guided local outlier factor," *Pattern Recognition*, vol. 117, p. 107986, 2021.