Supplementary Material of “DeMatch: Deep Decomposition of Motion Field for Two-View Correspondence Learning”

Shihua Zhang, Zizhuo Li, Yuan Gao, Jiayi Ma*
Electronic Information School, Wuhan University, Wuhan 430072, China
suhzhang001@gmail.com, zizhuo_li@whu.edu.cn, {ethan.y.gao, jyma2010}@gmail.com

We give the structure details of some components in DeMatch, and provide more experimental results to show the superior performance and promising properties, especially the piecewise smoothness. We also conduct further analysis to determine the structure of DeMatch.

1. Structure Details

In this section, we give structure details of the initial module, the graph attention network $\mathcal{G}(\cdot,\cdot)$, and the inlier predictor, all of them are not presented in the main paper.

1.1. Initial Module

Once getting the putative motion vectors $\{\boldsymbol{m}_i = (\boldsymbol{x}_i, \boldsymbol{d}_i) | i = 1, \dots, N\}$, we try to map them into high dimensional sapce as $\boldsymbol{F} = \{\boldsymbol{f}_i\}$ with initial module. The dimensions of coordinates $\boldsymbol{X} = \{\boldsymbol{x}_i\}$ and displacements $\boldsymbol{D} = \{\boldsymbol{d}_i\}$ are upgraded, respectively, then the results are summed together to achieve positional embedding [7]. The initial module totally consists of 1×1 convlutional layers (1×1 Conv), batch normalization [3] (Batch Norm) and Relu activation function [2] (ReLU). Details are shown in Figure 1.

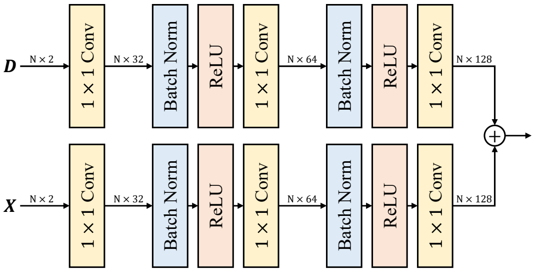

Figure 1. Structure details of the initial module.

1.2. Graph Attention Network

Graph attention network is performed in the paper with:

$$\boldsymbol{Z} = \mathcal{G}(\boldsymbol{B}, \boldsymbol{F}) = \boldsymbol{B} + \text{FFN}(\boldsymbol{B} \| \mathcal{A}(\boldsymbol{B}, \boldsymbol{F})), \tag{1}$$

where $\mathcal{A}(\boldsymbol{B}, \boldsymbol{F})$ is the standard attention mechanism:

$$\mathcal{A}(\boldsymbol{B}, \boldsymbol{F}) = \text{Softmax}\left(\frac{(\boldsymbol{W}_Q \boldsymbol{B})(\boldsymbol{W}_K \boldsymbol{F})^T}{\sqrt{C}}\right) \boldsymbol{W}_V \boldsymbol{F}. \tag{2}$$

The structure of $\mathcal{G}(\boldsymbol{B}, \boldsymbol{F})$ is shown in Figure 2, where $\boldsymbol{Q}$, $\boldsymbol{K}$, $\boldsymbol{V}$ indicate the query, key and value in attention mechanism, respectively. Note that we perform multi-head attention practically for $\mathcal{A}(\cdot,\cdot)$ in Eq. (2) as the same as [7]. And the function FFN$(\cdot)$ (*i.e.* feed-forward network) consists of 1×1 convolutional layers, batch normalization, and Relu activation function.

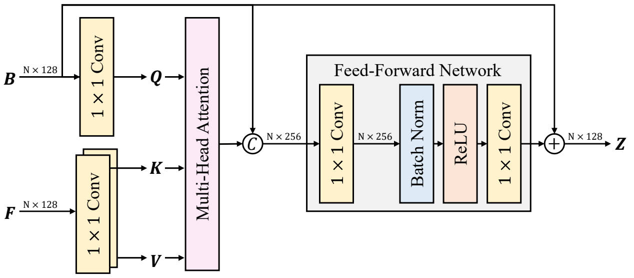

Figure 2. Structure details of the graph attention network.

1.3. Inlier Predictor

Similar to many other methods [10, 11], inlier predictor is feeded with the difference of motion vector features ${}^{\ell+1}\boldsymbol{F} - {}^{\ell}\boldsymbol{F} = \{{}^{\ell+1}\boldsymbol{f}_i - {}^{\ell}\boldsymbol{f}_i\}$ and outputs the predicted logits ${}^{\ell}\hat{\boldsymbol{\omega}} = \{{}^{\ell}\hat{\omega}_i\}$ for inliers/outliers classification. Except for the operations that the initial module has used, the predictor contains context normalization [9] (Context Norm) to identify inliers better. Details are shown in Figure 3.

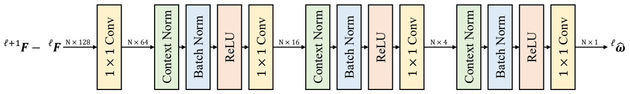

Figure 3. Structure details of the inlier predictor.

*Corresponding author

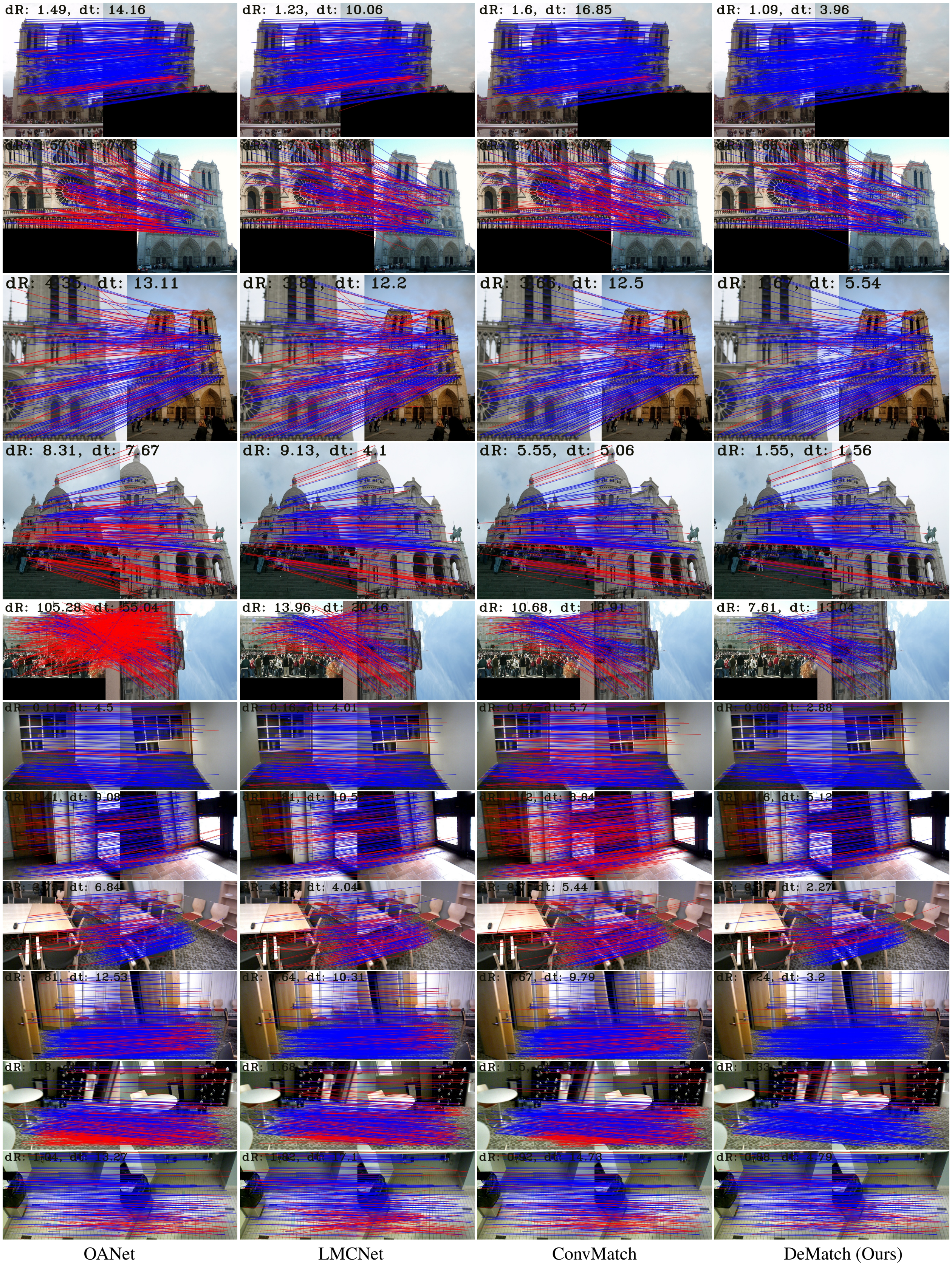

Figure 4. Qualitative illustration of outlier rejection and relative pose estimation. False matches are marked with red while correct matches are marked with blue. The relative pose estimation results (error of rotation and translation) are provided in the top left corner.

2. Experiments

2.1. Relative Pose Estimation

We show more visualization results (including OANet [10], LMCNet [4], ConvMatch [11] and DeMatch) of outlier rejection and relative pose estimation for outdoor scenes (the 1-st row to the 5-th row) and indoor scenes (the 6-th row to the 11-th row) in Figure 4. Note that ConvMatch seems to perform worse than others in some indoor scenes (the 6-th row to the 8-th row), because the local filters in ConvMatch over-smooth the discontinuities in the case of large scene disparities, resulting in wrong matches. However, DeMatch can handle the problem of piecewise smoothness naturally with the decomposition of the motion field, leading to better results.

2.2. Robustness Test

We try to verify the robustness of the proposed approach in challenging scenarios. In fact, YFCC100M [6] and SUN3D [8] contain many difficult cases with less than 10% of inliers. Thus to further demonstrate DeMatch's excellent robustness, we reduce the inlier ratio from 10% to only 2%, repeating the outdoor pose estimation experiment with these different inlier ratios. Results are shown in Figure 5. The models of DeMatch and other algorithms (OANet [10], CLNet [12], MS^{2}DGNet [1] and ConvMatch [11]) trained on YFCC100M with SIFT [5] are used for evaluation. Although the accuracy of all methods decreases as the inlier ratio declines, the proposed DeMatch is less volatile and consistently performs better than others, displaying better robustness.

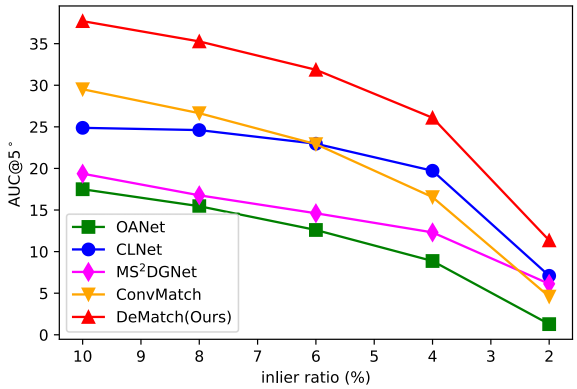

Figure 5. Robustness test. **AUC@5°** with the weighted eight-point algorithm is reported.

2.3. Piecewise Smoothness Property

We have demonstrated the piecewise smoothness property of DeMatch in the paper. We will provide more visualization results here in Figure 6. The first column shows the correspondence clusters, while the second column shows the motion clusters. Each cluster is highly consistent and subordinates to a particular motion pattern. We draw the top-4 clusters with the number of correspondences in different colors for better visualization, where the clusters with less than 5 correspondences are deemed as meaningless noise. The third column shows the highly smooth sub-fields that are formed by corresponding clusters on the "low-frequency" basis. In this way, we decompose the motion field into several sub-fields, constraining the smoothness of sub-fields respectively instead of smoothening the whole motion field directly. Therefore, piecewise smoothness can be guaranteed naturally, and discontinuities at the edge of different sub-fields can be maintained correctly. As shown in the 6-th to the 8-th row of Figure 4, DeMatch handles the problem of piecewise smoothness well. Note that the correspondences and clusters in Figure 6 are predicted totally by DeMatch, hence wrong matches may exist in the visualization. And in the 5-th to the 7-th rows, significant clusters are very limited for the simple scenes and small transformations, so that original motion field is decomposed into only a few (less than 4) sub-fields.

Furthermore, we visualize the effect of sub-fields on the final performance in Figure 7. As the clusters of motion vectors, *i.e.*, sub-fields, are progressively removed, the errors in rotation (dR) and translation (dt) become bigger and bigger. This shows the great influence of sub-fields on the performance of pose estimation.

2.4. Parameters of Network Structure

In the paper, we have chosen the number of layers $L = 5$ and the number of motion patterns $K = 48$ as default. Although the growth of L leads to better performance, the high computational usage limits a large L, and $L = 5$ is a good performance and cost balance. But for K, the rule that bigger is better is not suitable. As K increases, more low-frequency information of the motion field $\mathcal{F}$ is retained, and the new field constructed by K sub-fields $\{\mathcal{F}^k\}$ becomes closer to the real clean field $\widetilde{\mathcal{F}}$. However, when K is larger than a certain value, some high-frequency noise is also retained, which can negatively affect the smoothness of the new field and result in worse performance. The experimental results in the paper also demonstrate this concept. Thus, we choose a proper $K = 48$ which is not too small or too large. Furthermore, we attempt different numbers of times

Table 1. Parameter settings. The metric is **AUC@10°** with the weighted eight-point algorithm. Note that one of the parameters is fixed while another changes.

Metric	α=1	$\boldsymbol{\alpha}$**=2**	α=3	β=2	$\boldsymbol{\beta}$**=4**	β=6
AUC@10°	48.95	52.67	51.45	49.15	52.67	52.95
Flops (G)	1.987	2.346	2.706	2.267	2.346	2.426
Params (M)	5.027	5.853	6.679	4.200	5.853	7.505

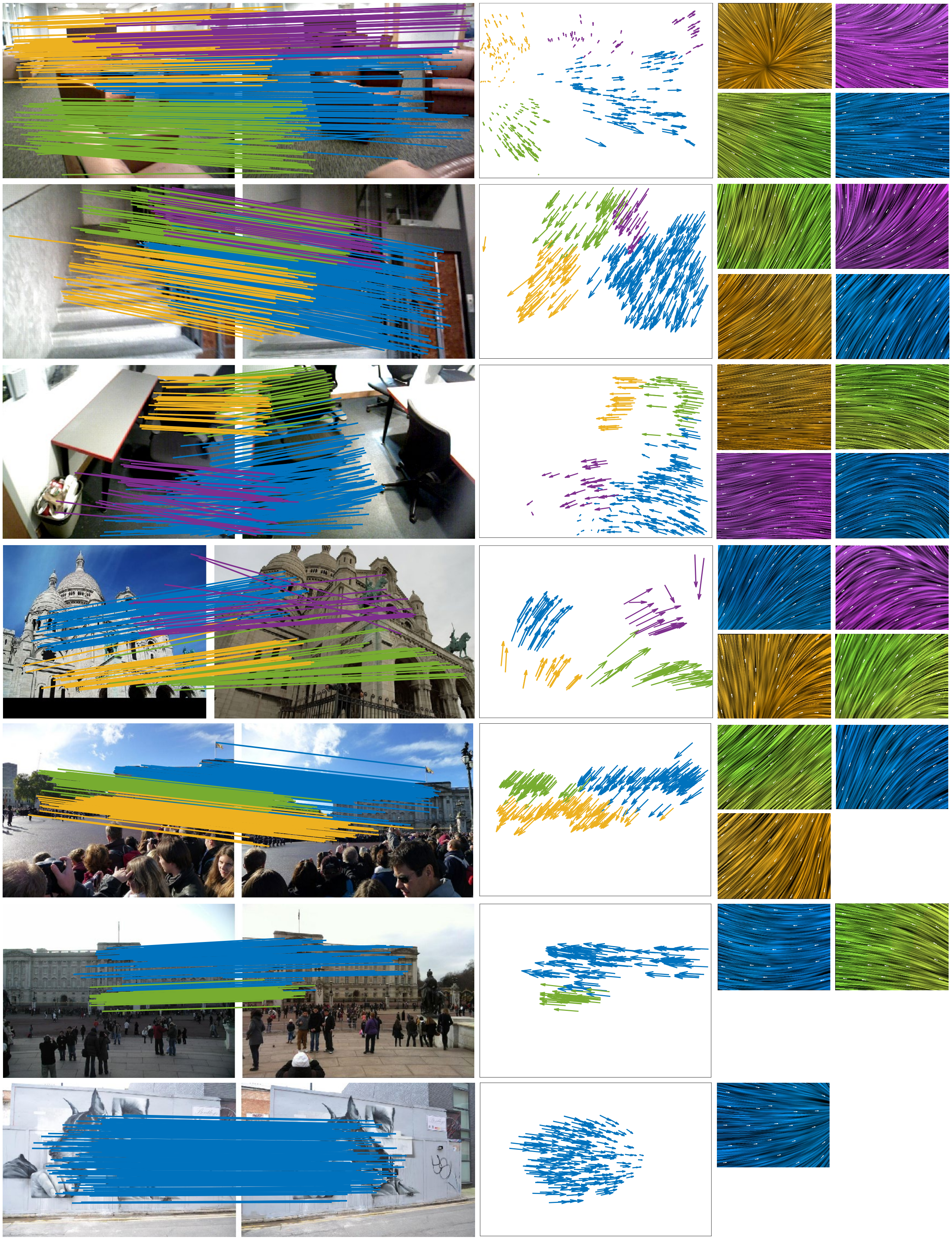

Figure 6. Illustration for piecewise smoothness property. Clusters are marked with different colors to represent different motion patterns, and the sub-field is formed by the cluster that is painted with a similar color. Zoom in for better visualization.

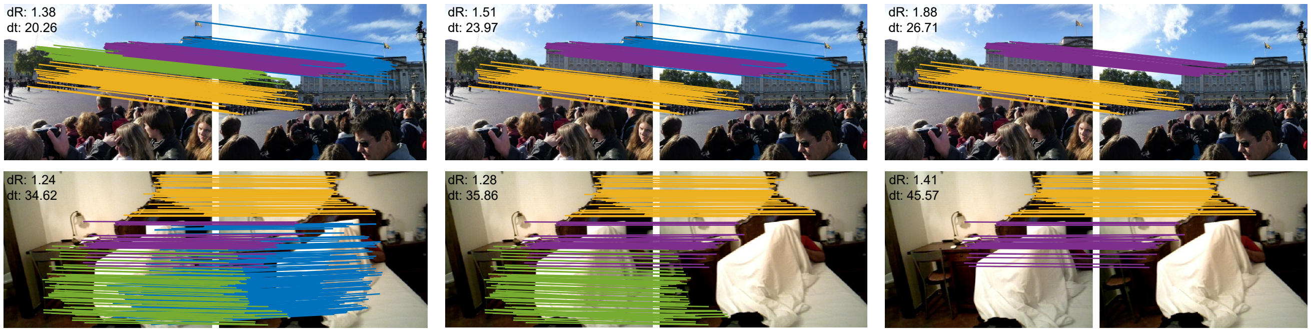

Figure 7. The effect of sub-fields on the final performance. Motion clusters that form smooth sub-fields are removed gradually. The errors of rotation and translation are shown in the top-left corner. Zoom in for better visualization.

the decomposition and the global enhancement are reused (noted as α and β, respectively) to find a better network structure of DeMatch. Results are shown in Table 1, and we choose $\alpha = 2$ and $\beta = 4$ finally to achieve unsurpassed performance while reducing computational usage.

References

[1] Luanyuan Dai, Yizhang Liu, Jiayi Ma, Lifang Wei, Taotao Lai, Changcai Yang, and Riqing Chen. Ms2dg-net: Progressive correspondence learning via multiple sparse semantics dynamic graph. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8973–8982, 2022. 3

[2] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 315–323, 2011. 1

[3] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the International Conference on Machine Learning*, pages 448–456, 2015. 1

[4] Yuan Liu, Lingjie Liu, Cheng Lin, Zhen Dong, and Wenping Wang. Learnable motion coherence for correspondence pruning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3237–3246, 2021. 3

[5] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 3

[6] 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. 3

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the Advances in Neural Information Processing Systems*, 2017. 1

[8] 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. 3

[9] 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

[10] 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. 1, 3

[11] Shihua Zhang and Jiayi Ma. Convmatch: Rethinking network design for two-view correspondence learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3472–3479, 2023. 1, 3

[12] Chen Zhao, Yixiao Ge, Feng Zhu, Rui Zhao, Hongsheng Li, and Mathieu Salzmann. Progressive correspondence pruning by consensus learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6464–6473, 2021. 3