End-to-End Learning Local Multi-view Descriptors for 3D Point Clouds

Lei Li[1*] Siyu Zhu[2] Hongbo Fu[3†] Ping Tan[2,4] Chiew-Lan Tai[1]
[1]HKUST [2]Alibaba A.I. Labs [3]City University of Hong Kong [4]Simon Fraser University

1. Supplementary Material

1.1. CNN

In Sec. 3.2 of the main text, we adopt a CNN architecture similar to L2-Net [6] to extract feature maps for each view patch. The detailed configuration of the network is listed in Table 1. Note that the network input is of size 64×64 with a single depth channel, and the final output is of size 8×8 with 128 feature channels.

#	Layer	Kernel	Stride	Padding
1	Conv - Norm - ReLU	3×3×32	2	1
2	Conv - Norm - ReLU	3×3×32	1	1
3	Conv - Norm - ReLU	3×3×64	2	1
4	Conv - Norm - ReLU	3×3×64	1	1
5	Conv - Norm - ReLU	3×3×128	2	1
6	Conv - Norm - ReLU	3×3×128	1	1

Table 1: CNN backbone for feature extraction of each view patch. In the *Kernel* column, the first two numbers represent the kernel size, and the third number is the number of output feature channels.

1.2. Multi-view Rendering

In Fig. 1, we visualize the optimizable viewpoints after training. We also show the viewpoints obtained by a clustering scheme similar to the one in [3]. Specifically, 150 spherical coordinates (θ, ϕ) are randomly sampled on the hemisphere where point normals reside, and then the k-medoids clustering algorithm is applied to select three viewing directions. For each viewing direction, a virtual camera is placed at distances of 0.3m, 0.6m, 0.9m to the points of interest, and each rendered view patch is augmented with four in-plane rotations.

As shown in Fig. 1, there are mainly two differences between the hand-crafted rule and our method. First, the hand-crafted rule places some viewpoints far from points of interest, while the learnt viewpoints have more concentrated distance range, indicating the relatively low importance of broader global context. Second, the hand-crafted rule selects some dominant viewing directions through clustering, whereas the learnt viewpoints have more distributed viewing directions around the points of interests, which can help to capture more local geometry variance. In sum, the learnt viewpoints effectively balance the extent of context-awareness and local details in extracted descriptors, challenging the design wisdom of hand-crafted rules.

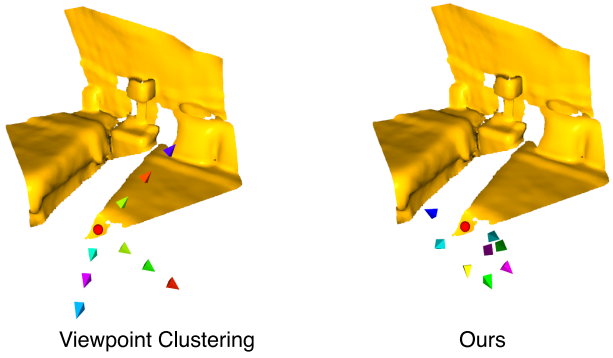

Figure 1: Visualization of viewpoints obtained by a clustering scheme and our method. The red spheres denote the points of interest, and the pyramids represent virtual cameras.

1.3. Multi-view Fusion

In Sec. 4.4 of the main text, we compared the proposed soft-view pooling with alternative fusion approaches including max-view pooling [3, 5, 4], Fuseption [7], and NetVLAD [1]. Fuseption has two branches: in the first branch, the feature maps of all the views are first channelwise concatenated together in a specific order and then fed into a convolutional block; in the second branch, max-pooling is applied to the inputs and the results are added to the output of the first branch, serving as a shortcut connection. NetVLAD is a descriptor pooling method that summarizes the residuals of each input w.r.t. several learnable cluster centers. The number of cluster centers is a hyper

*L. Li was an intern at Alibaba A.I. Labs.
†H. Fu is the corresponding author. E-mail: hongbofu@cityu.edu.hk

parameter, which is set to eight in our experiments. The network f is trained with the alternative fusion approaches, while the other stages are kept unchanged. The descriptor dimension d is set to 32, and the optimizable viewpoint number n is set to 8.

In Fig. 2, we visualize the rendered multi-view inputs to CNNs, extracted feature maps for each view, and fused feature maps across views. It is observed that the CNN is influenced by multi-view fusion for feature extraction. Before fusion, for soft-view pooling and NetVLAD, the feature maps of each view extracted by the CNN tend to have more response, compared to max-view pooling and Fuseption. After fusion, the feature maps produced by max-view pooling and NetVLAD tend to have more high response than soft-view pooling and Fuseption. Note that for each location in the fused feature maps, max-view pooling only selects the strongest input response across views and discards the rest, while our soft-view pooling collectively considers all the inputs in an attentive manner for integration.

1.4. Comparisons with 3DSmoothNet

In Fig. 3, we visualize the color-coded local descriptors for all the points in the point clouds. Specifically, we project the high dimensional descriptors with PCA and keep the first three components, which are color-coded. It is observed that the descriptors of 3DSmoothNet and our method are both geometry-aware. Particularly, our method is able to capture more geometric changes in the point clouds (see the highlighted wall, pillow and floor regions of the point clouds in Fig. 3). In Fig. 4, we show additional geometric registration results of point cloud pairs, which further demonstrate the above advantage of our method.

For the running time of 3DSmoothNet in Sec. 4.2 of the main text, we observed some gap between our experiment results (input prep: 39.4ms; inference: 0.2ms) and the performance reported by the authors (input prep: 4.2ms; inference: 0.3ms). We used the source code[1] of 3DSmoothNet released by the authors, and the running time gap of input preparation is likely due to the difference of hardware configurations. In [2], they used a PC with an Intel Xeon E5-1650, a 32GB RAM and an NVIDIA GeForce GTX 1080 GPU, while we used a PC with an Intel Core i7 @ 3.6GHz, a 32GB RAM and an NVIDIA GTX 1080Ti GPU. Their input preparation stage involving LRF computation and SDV voxelization runs on CPU, which may be accelerated with GPU for further improvement.

References

[1] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *Proc. IEEE CVPR*, June 2016. 1

[2] Zan Gojcic, Caifa Zhou, Jan D. Wegner, and Andreas Wieser. The perfect match: 3d point cloud matching with smoothed densities. In *Proc. IEEE CVPR*, 2019. 2

[3] Haibin Huang, Evangelos Kalogerakis, Siddhartha Chaudhuri, Duygu Ceylan, Vladimir G. Kim, and Ersin Yumer. Learning local shape descriptors from part correspondences with multiview convolutional networks. *ACM TOG*, 37(1):6:1–6:14, Nov. 2017. 1

[4] Charles R. Qi, Hao Su, Matthias Niessner, Angela Dai, Mengyuan Yan, and Leonidas J. Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *Proc. IEEE CVPR*, 2016. 1

[5] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proc. IEEE ICCV*, 2015. 1

[6] Y. Tian, B. Fan, and F. Wu. L2-Net: Deep learning of discriminative patch descriptor in euclidean space. In *Proc. IEEE CVPR*, pages 6128–6136, 2017. 1

[7] Lei Zhou, Siyu Zhu, Zixin Luo, Tianwei Shen, Runze Zhang, Mingmin Zhen, Tian Fang, and Long Quan. Learning and matching multi-view descriptors for registration of point clouds. In *Proc. ECCV*, 2018. 1

[1] https://github.com/zgojcic/3DSmoothNet

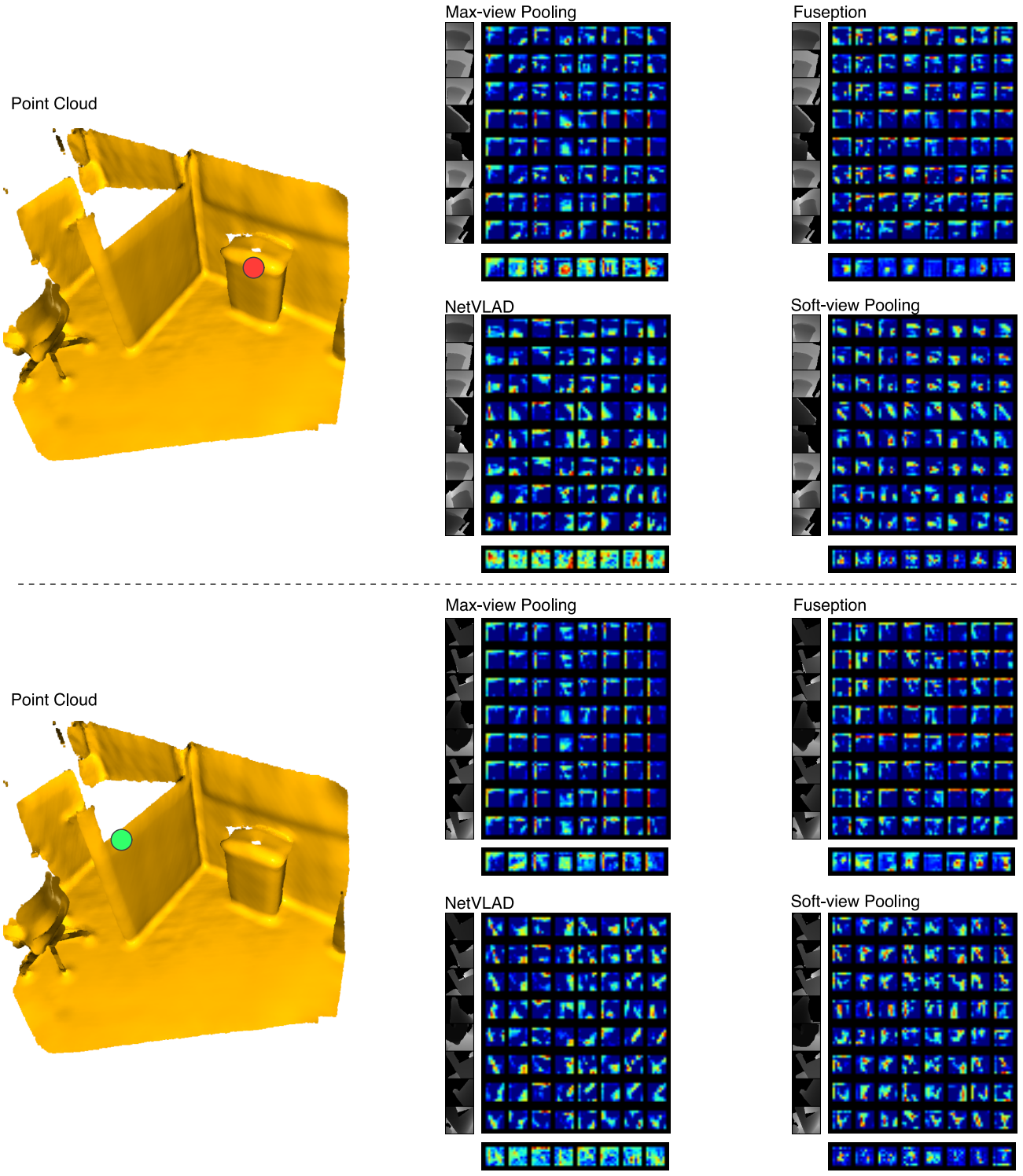

Figure 2: Visualizations for multi-view fusion by different methods. The top part is for the red keypoint while the bottom part is for the green keypoint. In each block, we visualize the view patches (depth) rendered with eight optimizable viewpoints on the left. On the right are the corresponding convolutional feature maps (with channel indices $\{1, 2, 4, 8, 16, 32, 64, 128\}$) before fusion, and each row is for a specific view. Fused feature maps across views are placed on the bottom.

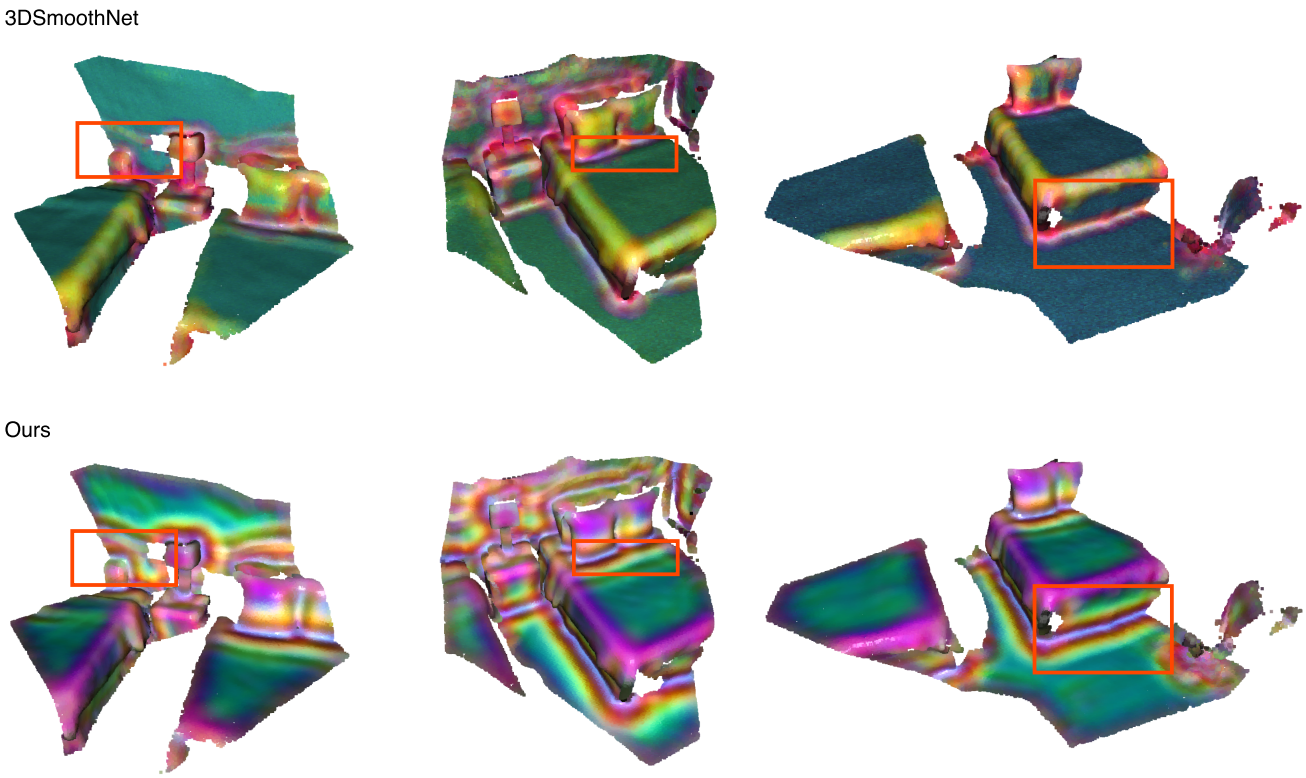

Figure 3: Visualization of local descriptors for 3DSmoothNet and our method. The high dimensional descriptors are projected with PCA to 3D space and color-coded. The highlighted regions show that our method can better capture geometric changes in the point clouds.

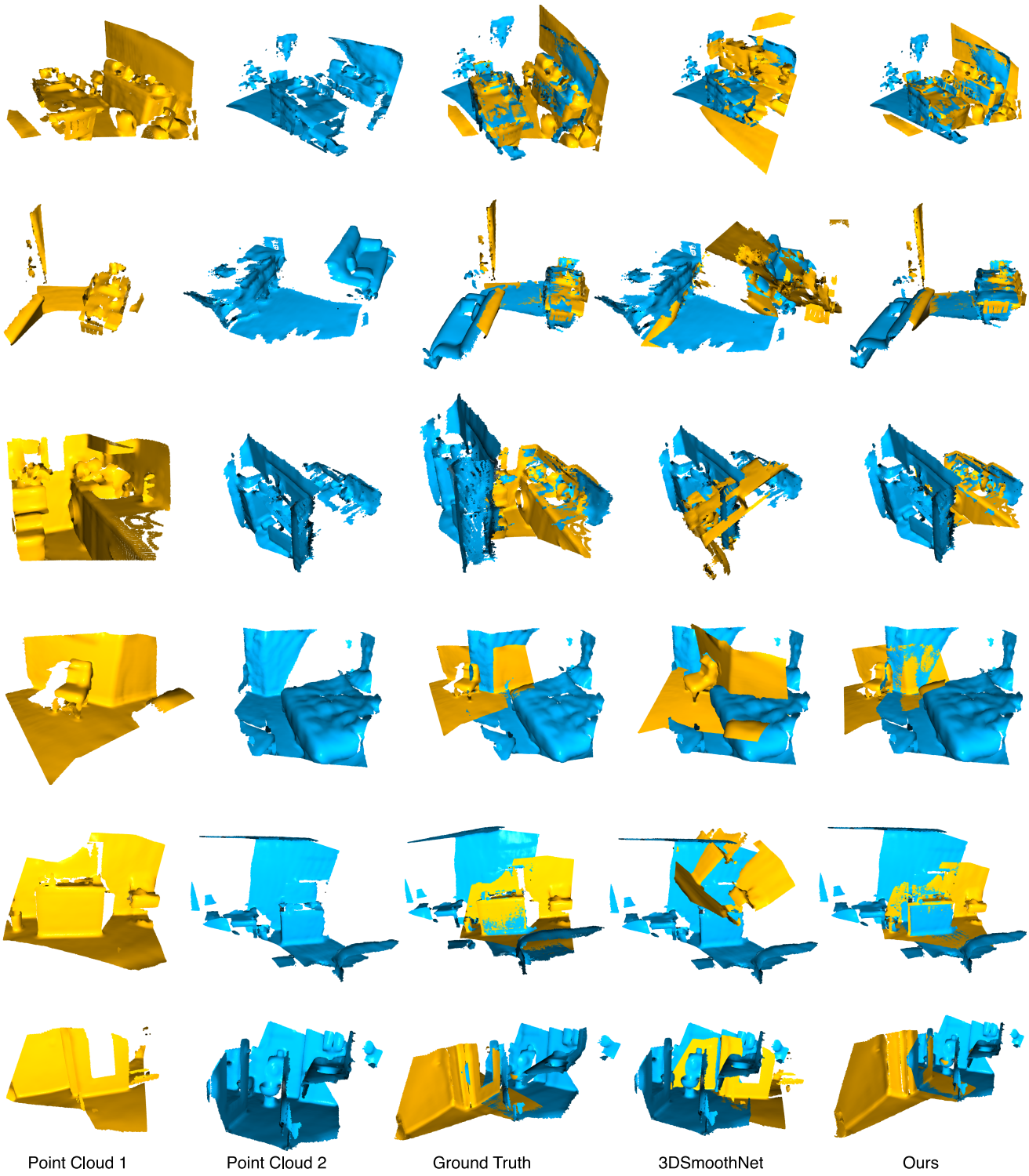

Figure 4: More geometric registration results with RANSAC for 3DSmoothNet and our method.