SHS-Net: Learning Signed Hyper Surfaces for Oriented Normal Estimation of Point Clouds — Supplementary Material —

Qing Li[1] Huifang Feng[2] Kanle Shi[3] Yue Gao[1] Yi Fang[4]
Yu-Shen Liu[1] * Zhizhong Han[5]
[1]School of Software, BNRist, Tsinghua University, Beijing, China
[2]School of Informatics, Xiamen University, Xiamen, China [3]Kuaishou Technology, Beijing, China
[4]Center for Artificial Intelligence and Robotics, New York University Abu Dhabi, Abu Dhabi, UAE
[5]Department of Computer Science, Wayne State University, Detroit, USA
{leoqli, gaoyue, liuyushen}@tsinghua.edu.cn fenghuifang@stu.xmu.edu.cn
shikanle@kuaishou.com yfang@nyu.edu h312h@wayne.edu

1. Evaluation Metrics

We adopt the same evaluation metrics as in [1, 5, 8, 14] to evaluate the estimated normal results of baseline methods and our method. Firstly, we employ the Root Mean Squared Error (RMSE) of normal angles between the ground-truth normals $\hat{\mathbf{n}}$ and the predicted normals $\mathbf{n}$, *i.e.*,

$$\mathrm{RMSE}_U = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\big(\arccos(|\hat{\mathbf{n}}_i \odot \mathbf{n}_i|)\big)^2}\ , \quad (1)$$

$$\mathrm{RMSE}_O = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\big(\arccos(\hat{\mathbf{n}}_i \odot \mathbf{n}_i)\big)^2}\ , \quad (2)$$

where RMSE_U and RMSE_O are used in unoriented and oriented normal evaluation, respectively. N is the number of evaluated normals in a point cloud. $|\cdot|$ represents the absolute value of the inner product $\odot$ of two normal vectors. Therefore, the normal angle errors are bounded between 0° and 90° in unoriented normal evaluation, and between 0° and 180° in oriented normal evaluation. Moreover, we adopt the Area Under the Curve (AUC) to show the distribution of normal errors. The AUC is plotted by the percentage of good points (PGP) whose normal angle errors are less than the angle thresholds. The PGP is computed by

$$\mathrm{PGP}_U(\tau) = \frac{1}{N}\sum_{i=1}^{N}\mathcal{I}\big(\arccos(|\hat{\mathbf{n}}_i \odot \mathbf{n}_i|) < \tau\big), \quad (3)$$

$$\mathrm{PGP}_O(\tau) = \frac{1}{N}\sum_{i=1}^{N}\mathcal{I}\big(\arccos(\hat{\mathbf{n}}_i \odot \mathbf{n}_i) < \tau\big), \quad (4)$$

where $\mathrm{PGP}_U(\tau)$ and $\mathrm{PGP}_O(\tau)$ are used in unoriented and oriented normal evaluation, respectively. $\mathcal{I}$ represents an indicator function that measures whether the error is less than a threshold τ.

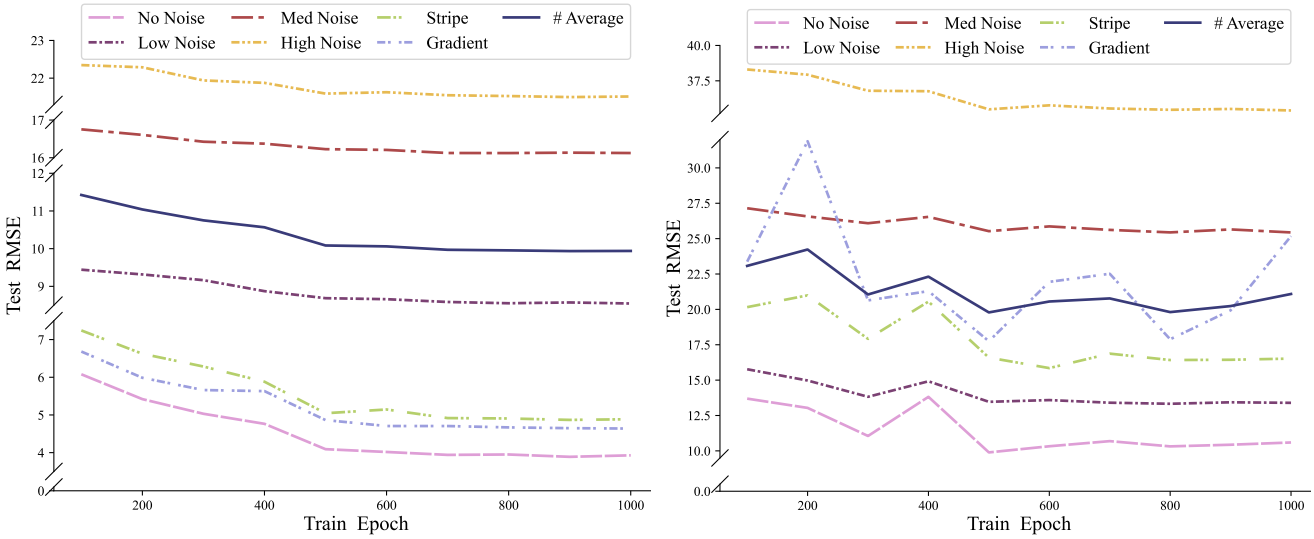
Table 1. Comparison of the normal angle RMSE, the learnable network parameter (million) and the average execution time (seconds per 100k points) of different learning-based methods for oriented normal estimation on the PCPNet dataset. Our method achieves a large performance gain with relatively few network parameters and execution time.

	Ours	PCPNet [5]	HSurf-Net [8] +ODP [11]	AdaFit [14] +ODP [11]
RMSE	19.79	37.66	31.07	30.93
Param.	3.27	22.36	2.16+0.43	4.87+0.43
Time	65.89	63.02	72.47+236.35	56.23+248.54

2. Tangent Plane of Explicit & Implicit Surface

Generally, an explicit surface $z = f(x, y)$ can be rewritten as an implicit surface, *i.e.*, $F(x, y, z) = z - f(x, y) = 0$. These two surface representations have the same tangent plane at a given point $p = (x_0, y_0, z_0)$, where the point normal is defined. We briefly prove this conclusion as follows.

*The corresponding author is Yu-Shen Liu. This work was supported by National Key R&D Program of China (2022YFC3800600), the National Natural Science Foundation of China (62272263, 62072268), and in part by Tsinghua-Kuaishou Institute of Future Media Data.



(a) Unoriented normal evaluation

(b) Oriented normal evaluation

Figure 1. Unoriented and oriented normal evaluation results on the PCPNet test set using our models trained for 100 to 1200 epochs. We report the results under different noise levels and density variations along with their average values. Note that the normals used in unoriented and oriented normal evaluation are the same, but evaluated using different metrics, *i.e.*, RMSE_U and RMSE_O.

For the explicit surface $z = f(x, y)$, its tangent plane is given by

$$f'_x(x_0, y_0)(x - x_0) + f'_y(x_0, y_0)(y - y_0) + z_0 = z. \quad (5)$$

For the implicit surface $F(x, y, z) = 0$, its tangent plane is given by

$$F'_x(p)(x - x_0) + F'_y(p)(y - y_0) + F'_z(p)(z - z_0) = 0. \quad (6)$$

Since $F(x, y, z) = z - f(x, y)$, we have

$$F'_x(p) = -f'_x(x_0, y_0),\ F'_y(p) = -f'_y(x_0, y_0),\ F'_z(p) = 1.$$

Then, the Eq. (6) can be rewritten as

$$-f'_x(x_0, y_0)(x - x_0) - f'_y(x_0, y_0)(y - y_0) + (z - z_0) = 0.$$

Finally, we obtain

$$f'_x(x_0, y_0)(x - x_0) + f'_y(x_0, y_0)(y - y_0) + z_0 = z.$$

This equation is the same as the Eq. (5). Therefore, we prove that the explicit and implicit surface representations have the same tangent plane at a given point.

3. Evaluation of the Trained Models

We evaluate our method on the PCPNet test set using models trained for 100 to 1000 epochs. The estimated normals are measured using the evaluation metrics RMSE_U and RMSE_O of normal angles, and the evaluation results are shown in Fig. 1. We provide the results at different noise levels and different density variations along with their average results. It can be seen from the curves in Fig. 1 that the errors of unoriented normal evaluation keep decreasing, while the errors of oriented normal evaluation fluctuate greatly. In our training, we observe that the model is harder to converge in oriented normal estimation than in unoriented normal estimation. After about 800 epochs of training, the errors of oriented normal evaluation reach a minimum value, and the errors of unoriented normal evaluation also reach the lowest value and remain unchanged. Therefore, in all experiments of the paper, we use the model trained in 800 epochs.

4. Complexity & Efficiency

In the oriented normal evaluation experiments, we compare our method to PCPNet [5], DPGO [13], and other methods that are based on a two-stage paradigm. The trained model of PCPNet [5] is available. The source code of DPGO [13] is uncompleted and its results on the PCPNet dataset are taken from its paper. For other methods that are based on a two-stage paradigm, we choose three *unoriented* normal estimation methods (PCA [6], AdaFit [14] and HSurf-Net [8]) and three normal orientation methods (MST [6], QPBO [12] and ODP [11]). We make different combinations of them to estimate oriented normals, such as AdaFit+ODP and HSurf-Net+ODP. Among these baselines, AdaFit [14], HSurf-Net [8] and ODP [11] are learning-based methods, and the others are traditional methods.

In this evaluation experiment, we compare the learning-based methods on the same machine with NVIDIA 2080

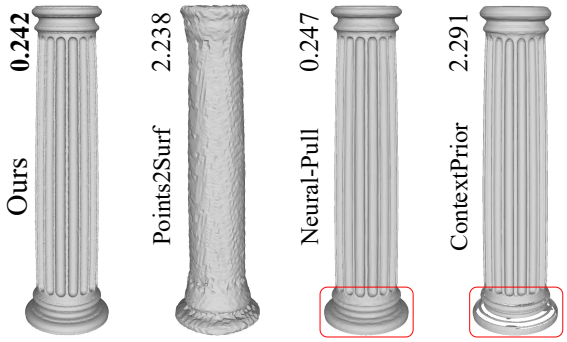


Figure 2. Comparison with surface reconstruction methods, *e.g.*, Points2Surf [3], Neural-Pull [9] and ContextPrior [10]. The chamfer distances ($\times 10^4$) are provided for the reconstructed surfaces.

Ti GPU. In Table 1, we report the normal angle RMSE, the number of the learnable network parameters, and the execution time of each method for oriented normal estimation. Our method achieves a large performance improvement with relatively fewer parameters and less running time.

5. Comparison with Surface Reconstruction Methods

For the normal estimation task, we do not simply use the skills of surface reconstruction methods. First, compared with the dual-stream structure of Points2Surf [3], we present novel task-specific patch/shape encoders and feature fusion strategy. Compared with POCO [2], we extend the attention mechanism with coplanarity weight τ and sign prediction to learn oriented normals from signed hyper surfaces. Second, we present a detailed mathematical derivation for the design of our algorithm. We make a comprehensive analysis of the problem in theory and make specific designs in technology. It is these new and important innovative designs that make our method outperforms the state-of-the-art methods in both unoriented and oriented normal estimation on the widely used benchmarks. Furthermore, we explore a new idea for learning local features and geometric properties from point clouds. It can better serve the community for point cloud processing and has a positive impact on the performance improvement of downstream tasks using normals.

As shown in Fig. 2, based on the accurate normals estimated by our method, Poisson reconstruction algorithm [7] reconstructs more complete detailed geometry from point clouds than baseline methods [3, 9, 10].

6. Comparison with HSurf-Net

In our work, we have made great innovations compared with HSurf-Net [8]. First, we design a brand new algorithm framework, including patch/shape encoder and attention-

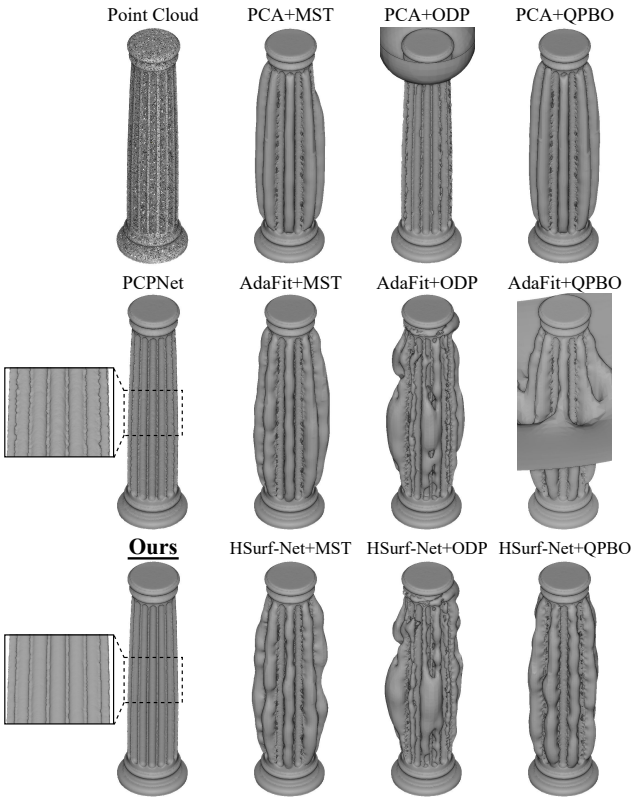


Figure 3. Comparison of surface reconstruction results using normals from different methods.

weighted normal prediction, and it achieves significant performance improvements in both oriented and unoriented normal estimation tasks. Second, estimating oriented normal is more difficult than unoriented normal, and it needs global information. In order to solve this challenging task, we use the theory of learning implicit surface to further expand the Hyper Surface, such that it is signed to be able to determine the orientation of normal. This is our advantage over HSurf-Net which can only be used to estimate unoriented normal. Our experimental results (Table 1, 2 and Fig. 5, 6 in the paper) show the effectiveness of novel designs in our method.

7. Visualization of Weight and Attention

In Fig. 4, we visualize the learned attention weight in the normal prediction module $\mathcal{H}$, weight τ in Eq.(12) and weight w in Eq.(14) of the paper. They illustrate the points that the model focuses on at different stages of the normal estimation process. The weight w indicates that the model focuses on points closer to the center during the patch and shape encoding. The weight τ indicates that the model focuses on points coplanar with the query point during the final local feature modulation for normal prediction. The attention weight indicates that the model focuses on the

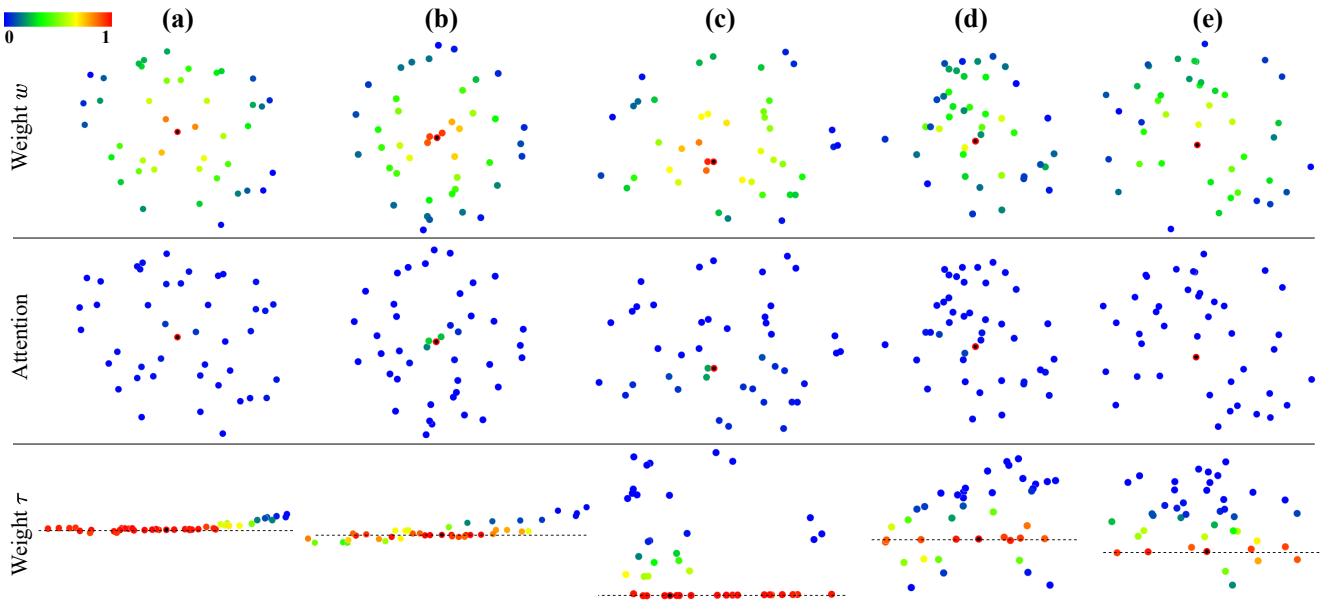


Figure 4. Visualization of the learned attention weight, weights w and τ in point cloud patches (a)-(e). They show the points that the model focuses on at different stages of the normal estimation process. Specifically, the points are the following three types: (top) the points closer to the center, (middle) the query point and its neighbors, and (bottom) the points coplanar with the query point. The red color indicates that the point has a large value, while the blue color indicates that it has a small value. The black point is the query point of the patch. The viewing angle of the third row is changed for better visualization.

query point during the final oriented normal prediction of the query point.

8. Limitation and Failure Case

We find that the task of estimating oriented normals with consistent orientations is more challenging than finding the perpendicular line of a local plane/surface. In our experiments, we observe that our method can deliver good unoriented normal results from various point clouds. In contrast, our method has the potential to fail on normal orientation in oriented normal evaluation. As we introduced in the paper, our method determines the normal orientation depending on the global latent code that is extracted from the global subsample set of the shape point cloud. Thus, the quality of the global subsample set will greatly affect the final normal orientation result. For our network model, the global subsample sets from different sampling strategies lead to significant performance differences, as we demonstrate in the ablation studies. At present, the sampling strategy we use is not optimal. Ideally, the errors in the unoriented and oriented normal evaluation should be the same, *i.e*. the normals are all oriented correctly. In practice, however, if the global subsample set cannot provide clear information to distinguish the spatial relationship between the inside and the outside, then the result of normal orientation may be in the opposite direction. This will lead to poor evaluation results for oriented normals even though their corresponding unoriented normals are very accurate. In summary, the bottleneck of oriented normal estimation is to estimate the correct normal orientation.

In Fig. 5, we show some failure cases of our method in oriented normal estimation. In these cases, the unoriented normal results are accurate, and the normal orientations of most points are also correct. We can see that the failure only occurs in the regions with special structures, such as small holes, complex concave structures, and the inner wall of the bending pipe with uneven density. In these areas, the information provided by globally sampled point clouds is insufficient or ambiguous. Moreover, we find that our oriented normal results are unsatisfactory on objects in indoor scenes, such as vertical walls inside the closed room, where our method cannot estimate normals with consistent orientations since they do not form internal and external structures. Future work includes exploring the transfer of context information between patches, such as using the orientation of adjacent patches to form a mechanism similar to LSTM or attention in NLP, and treating patches in point clouds as words in sentences. It is also important work to develop noise-adaptive techniques [4] to handle more diverse point clouds.

9. More Discussion

Loss for neighboring points. The loss for neighboring point normals makes the attention-weighted module shift

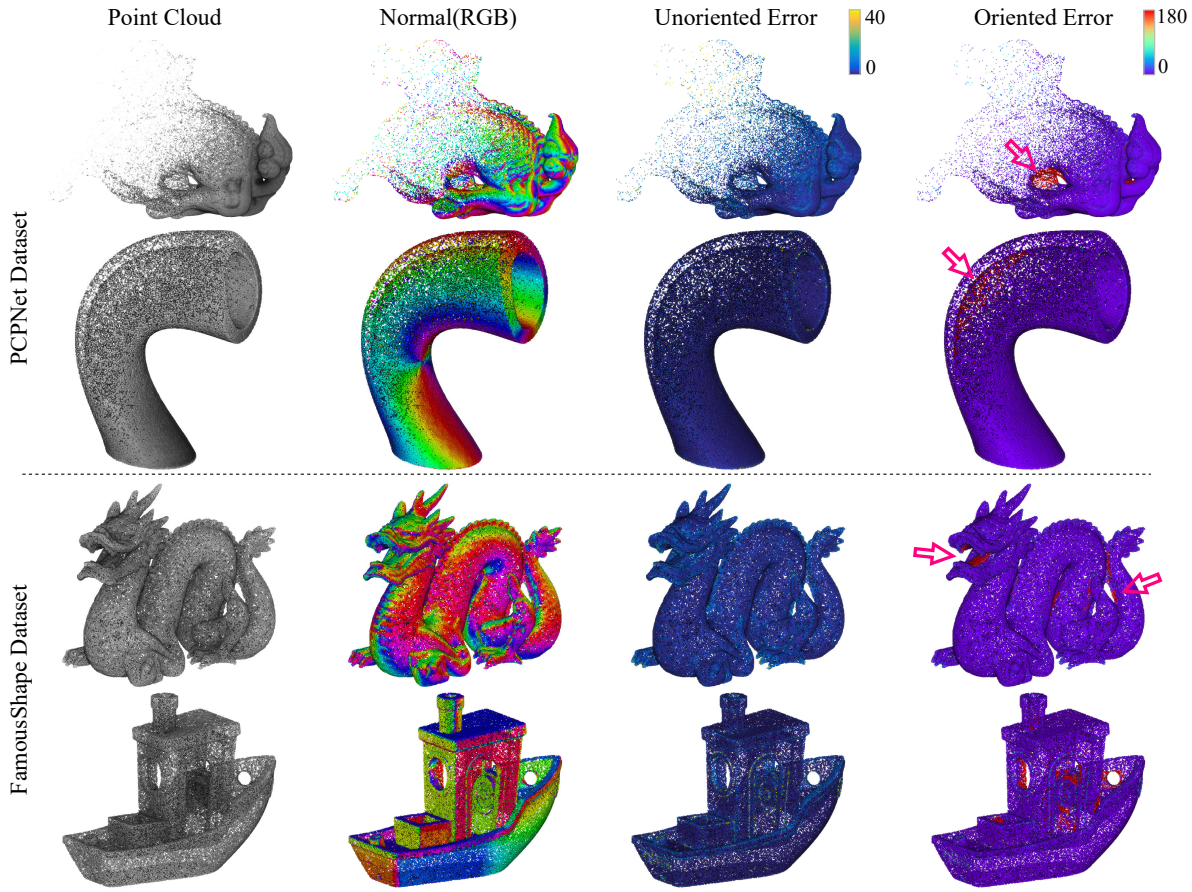


Figure 5. Visualization of failure cases on datasets PCPNet (top) and FamousShape (bottom). Our method can estimate the unoriented normals with high accuracy. However, the normal orientations in some local areas are not correctly determined. We map the normal vector and normal angle error to RGB colors for visualization.

some attention from the query point to the neighborhood points, which is beneficial for the patch/shape encoder to extract more useful neighborhood features. Thus, it helps the network learn to fit the Signed Hyper Surface in the feature space, thereby accurately describing the spatial geometry around the query point.

The network predicts the normal and its sign. Compared with directly predicting an oriented 3D normal vector by MLP, decomposing it into unoriented normal and its sign can reduce the search space of the model (from the whole spherical space to the hemisphere), and improve the accuracy of estimation results. Please see part (c) of ablation studies in the paper.

10. More Results

In Fig. 3, We show a visual comparison of surfaces reconstructed by the Poisson algorithm [7] based on the estimated normals. In Fig. 6, we show a visual comparison of the oriented normal angle RMSE of different methods. The point clouds are rendered in RGB colors generated from the error values.

In Fig. 7, we show the oriented normal AUC of the evaluated methods on different data categories of the FamousShape dataset. It can be seen that our method achieves significant performance improvements at all thresholds. In Fig. 8 and Fig. 9, we show the unoriented normal AUC of various methods on the datasets PCPNet and FamousShape, respectively. We can see that our method maintains a performance advantage at the vast majority of thresholds.

In Fig. 10 and Fig. 11, we provide more visual comparison results of the reconstructed surfaces on the KITTI dataset. The surfaces are generated by Poisson surface reconstruction algorithm [7] based on the oriented normals estimated by different methods. It can be seen that the reconstruction algorithm can benefit from the oriented normals estimated by our method to generate better scene surfaces.

In Fig. 12, we show the oriented normals estimated by our method on different shapes of the PCPNet dataset [5]. We map the normal vectors to RGB colors to render the point cloud. In Fig. 13, we visualize the angle RMSE of the oriented normals estimated by our method on different

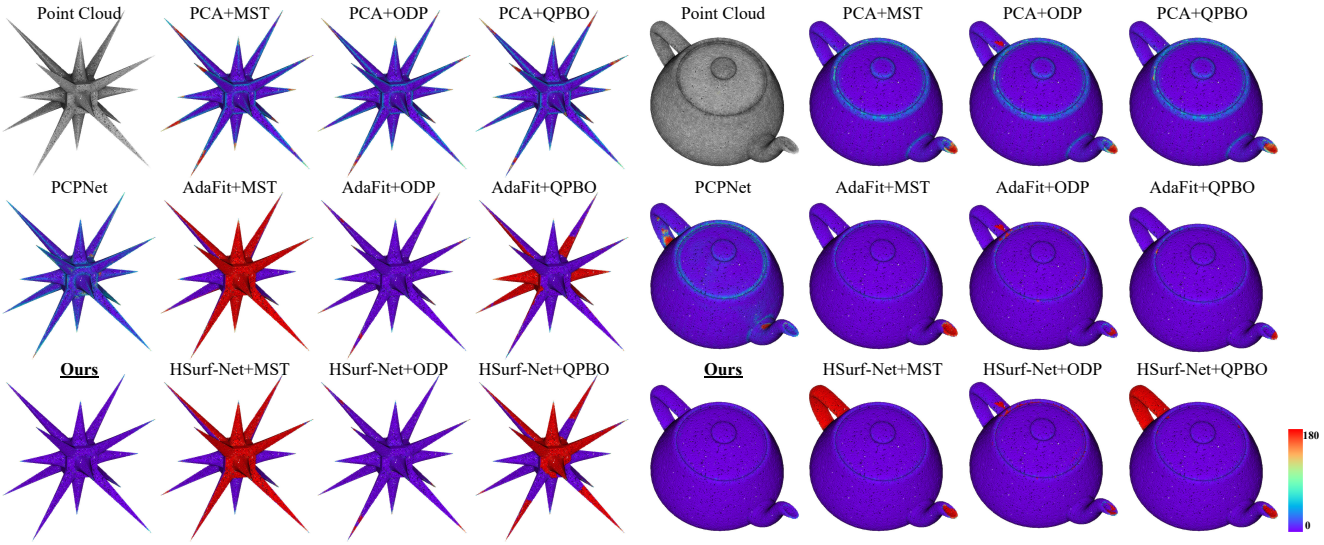


Figure 6. Visualization of the oriented normal error on datasets PCPNet (left) and FamousShape (right). The angle error is mapped to a heatmap ranging from 0° to 180°. The purple color indicates the same direction as the ground-truth, while the red color is the opposite.

shapes of the PCPNet dataset. We map the errors to RGB colors to render the point cloud.

Our FamousShape dataset is generated from mesh data following the same preprocessing steps as the PCPNet dataset. It contains 12 shapes with different structures. As shown in Fig. 14, we visualize the angle RMSE of the oriented normals estimated by our method on all shapes of the FamousShape dataset.

References

[1] Yizhak Ben-Shabat and Stephen Gould. DeepFit: 3D surface fitting via neural network weighted least squares. In *European Conference on Computer Vision*, pages 20–34. Springer, 2020. 1

[2] Alexandre Boulch and Renaud Marlet. POCO: Point convolution for surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6302–6314, 2022. 3

[3] Philipp Erler, Paul Guerrero, Stefan Ohrhallinger, Niloy J Mitra, and Michael Wimmer. Points2Surf: learning implicit surfaces from point clouds. In *European Conference on Computer Vision*, pages 108–124. Springer, 2020. 3

[4] Simon Giraudot, David Cohen-Steiner, and Pierre Alliez. Noise-adaptive shape reconstruction from raw point sets. In *Computer Graphics Forum*, volume 32, pages 229–238. Wiley Online Library, 2013. 4

[5] Paul Guerrero, Yanir Kleiman, Maks Ovsjanikov, and Niloy J Mitra. PCPNet: learning local shape properties from raw point clouds. In *Computer Graphics Forum*, volume 37, pages 75–85. Wiley Online Library, 2018. 1, 2, 5

[6] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. In *Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques*, pages 71–78, 1992. 2

[7] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics*, 32(3):1–13, 2013. 3, 5

[8] Qing Li, Yu-Shen Liu, Jin-San Cheng, Cheng Wang, Yi Fang, and Zhizhong Han. HSurf-Net: Normal estimation for 3D point clouds by learning hyper surfaces. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 1, 2, 3

[9] Baorui Ma, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. Neural-Pull: Learning signed distance functions from point clouds by learning to pull space onto surfaces. *International Conference on Machine Learning*, 2021. 3

[10] Baorui Ma, Yu-Shen Liu, Matthias Zwicker, and Zhizhong Han. Surface reconstruction from point clouds by learning predictive context priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6326–6337, 2022. 3

[11] Gal Metzer, Rana Hanocka, Denis Zorin, Raja Giryes, Daniele Panozzo, and Daniel Cohen-Or. Orienting point clouds with dipole propagation. *ACM Transactions on Graphics*, 40(4):1–14, 2021. 1, 2

[12] Nico Schertler, Bogdan Savchynskyy, and Stefan Gumhold. Towards globally optimal normal orientations for large point clouds. In *Computer Graphics Forum*, volume 36, pages 197–208. Wiley Online Library, 2017. 2

[13] Shiyao Wang, Xiuping Liu, Jian Liu, Shuhua Li, and Junjie Cao. Deep patch-based global normal orientation. *Computer-Aided Design*, page 103281, 2022. 2

[14] Runsong Zhu, Yuan Liu, Zhen Dong, Yuan Wang, Tengping Jiang, Wenping Wang, and Bisheng Yang. AdaFit: Rethinking learning-based normal estimation on point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6118–6127, 2021. 1, 2

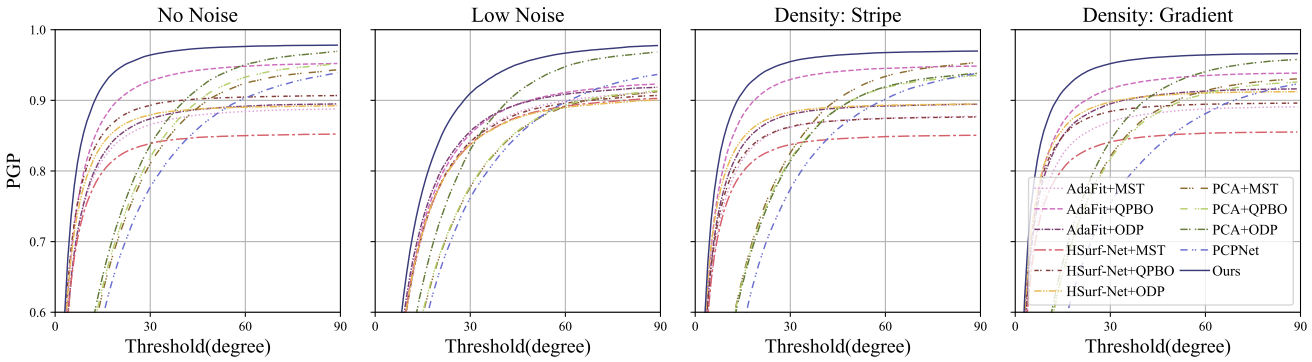


Figure 7. Oriented normal AUC on the FamousShape dataset. Our method achieves significant performance improvements at all thresholds. The X-axis is the angle threshold and the Y-axis is the percentage of good point normals (PGP) whose errors are less than the given threshold.

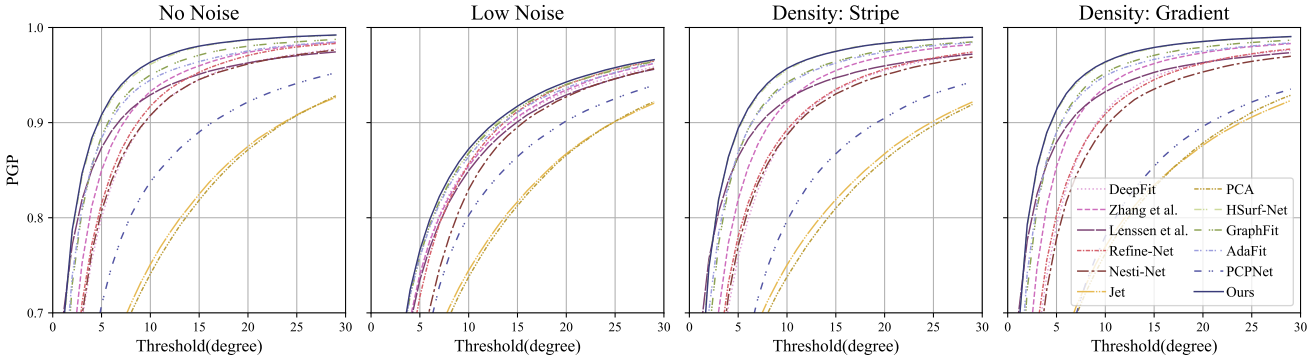


Figure 8. Unoriented normal AUC on the PCPNet dataset. Our method maintains a performance advantage at the vast majority of thresholds. The X-axis is the angle threshold and the Y-axis is the percentage of good point normals (PGP) whose errors are less than the given threshold.

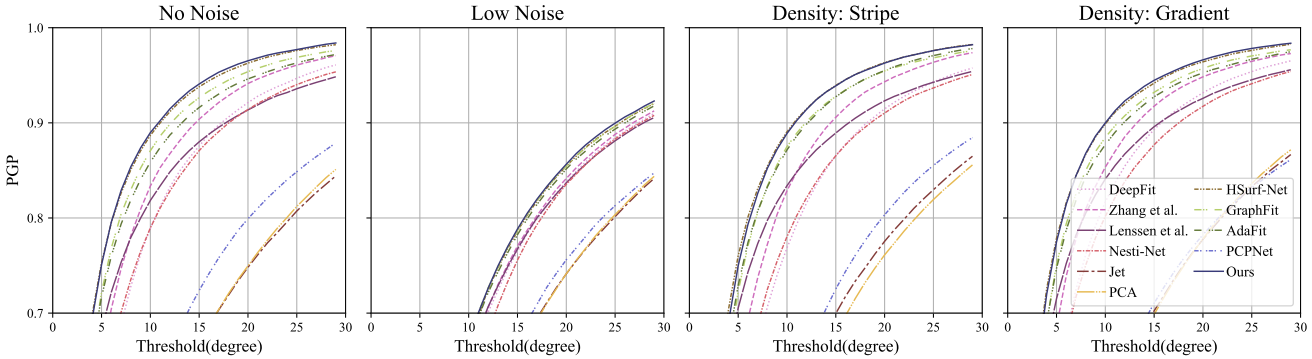


Figure 9. Unoriented normal AUC on the FamousShape dataset. Our method maintains a performance advantage at the vast majority of thresholds. The X-axis is the angle threshold and the Y-axis is the percentage of good point normals (PGP) whose errors are less than the given threshold.

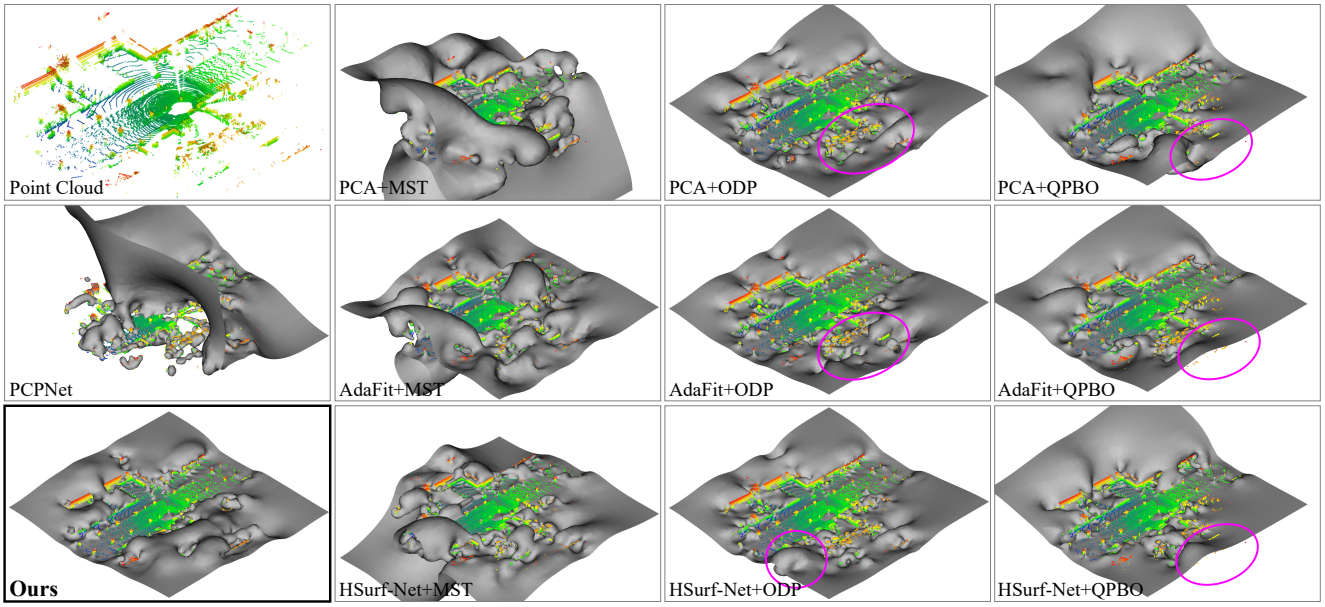


Figure 10. The reconstructed surfaces using oriented normals estimated by different methods on the KITTI dataset.

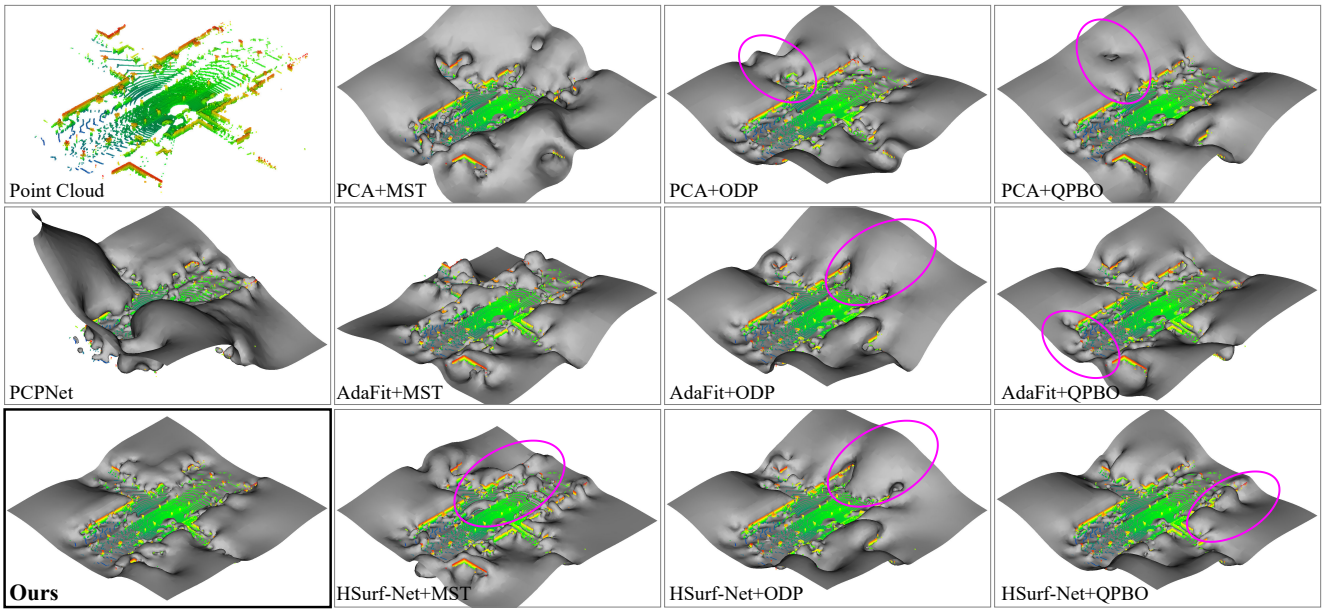


Figure 11. The reconstructed surfaces using oriented normals estimated by different methods on the KITTI dataset.

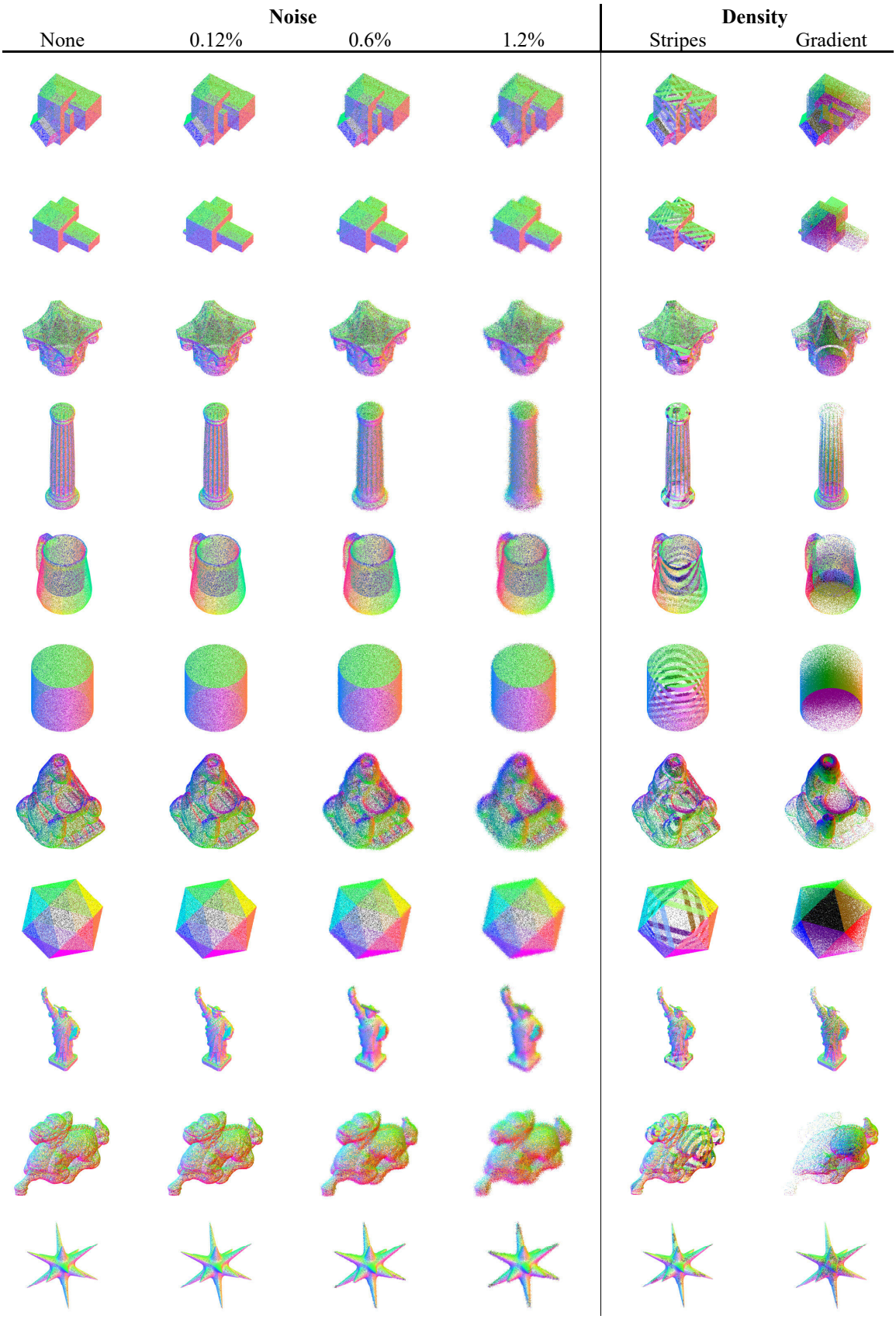


Figure 12. The normal estimation results of our method on the PCPNet dataset. The 3D point cloud normals are mapped to RGB colors for visualization.

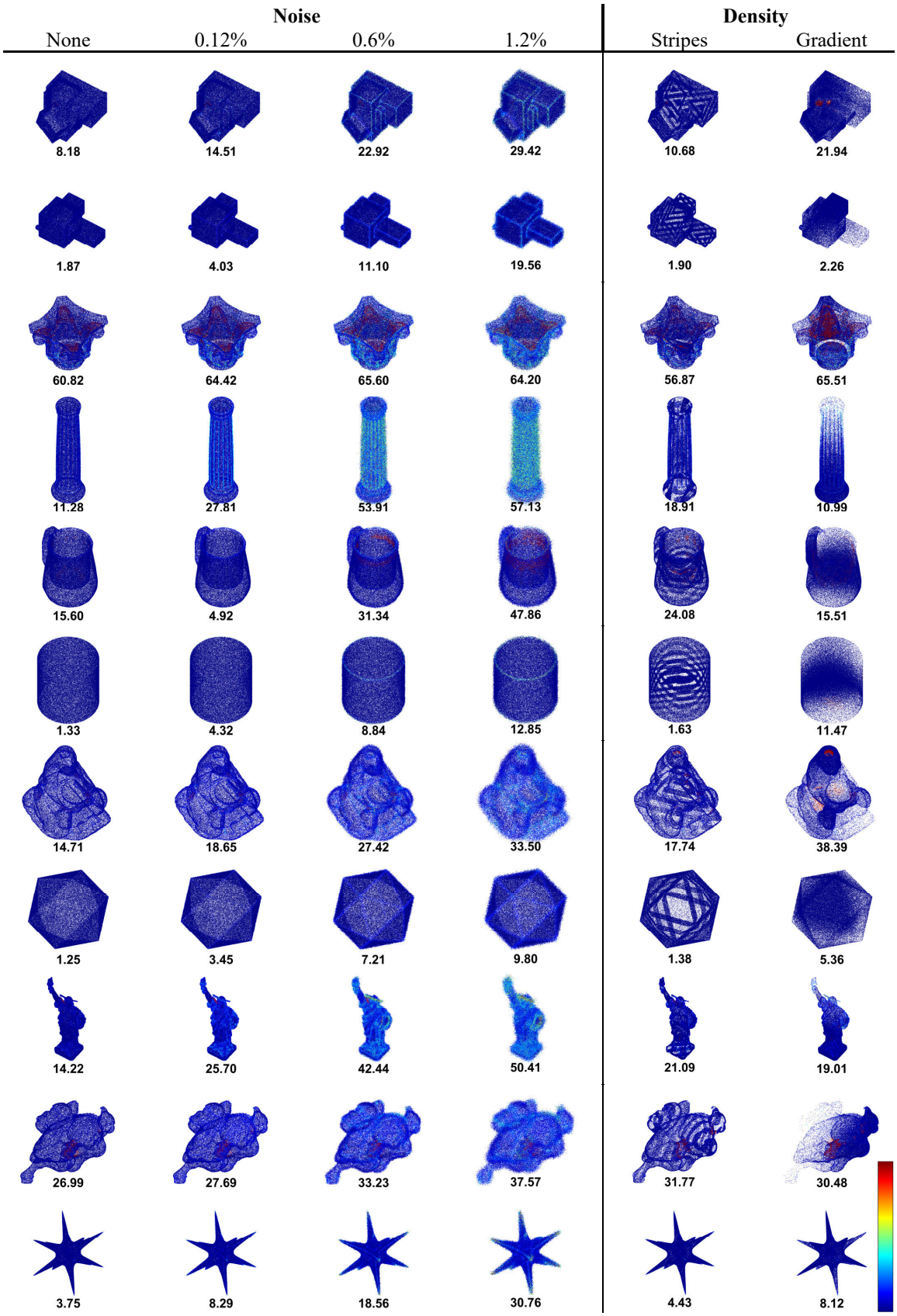
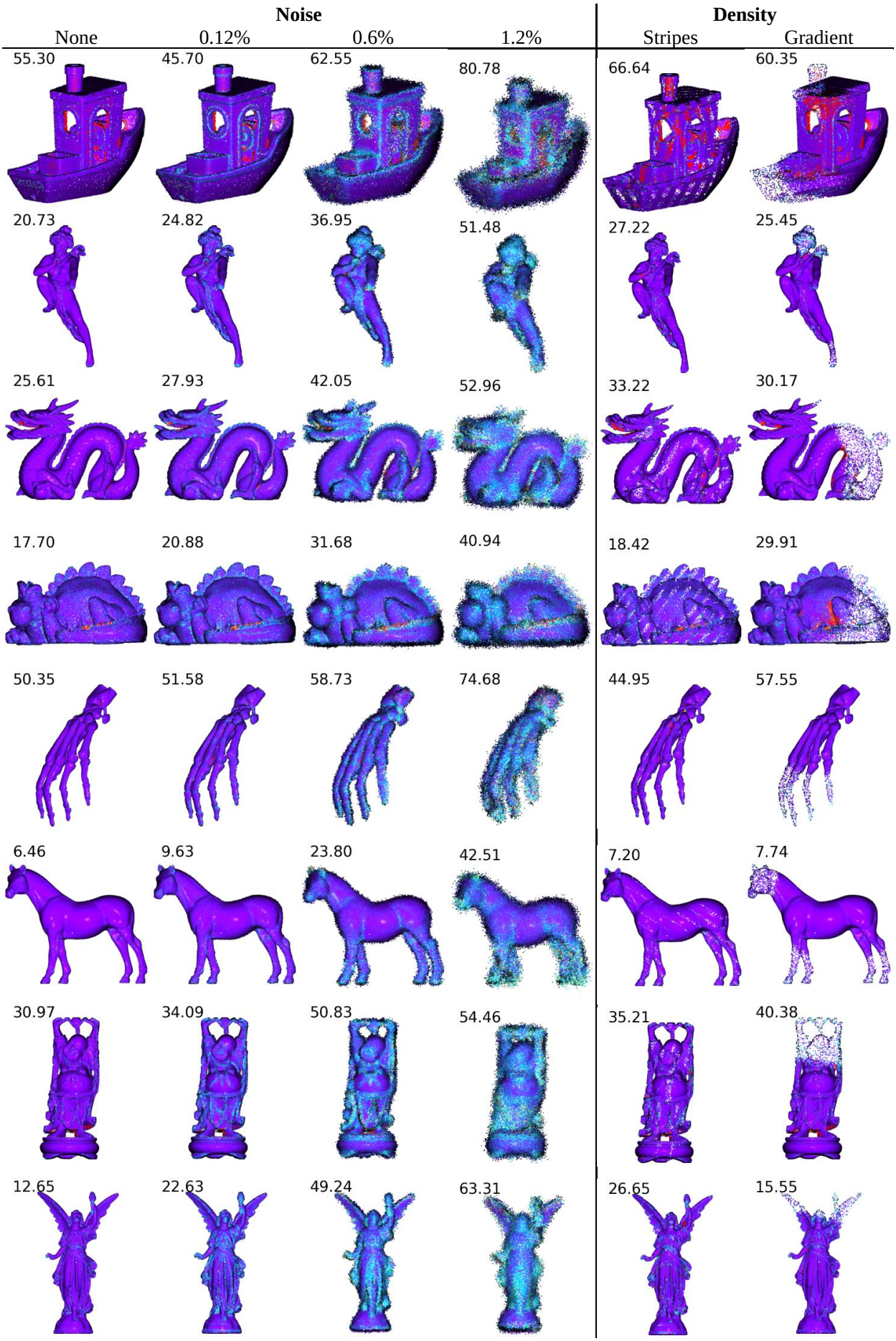


Figure 13. Error visualization of oriented normals estimated by our method on the PCPNet dataset. We map the normal errors to a heatmap ranging from 0° to 180° for visualization. The average RMSE values are reported under each point cloud.


Noise
Density
None
0.12%
0.6%
1.2%
Stripes
Gradient
55.30
45.70
62.55
80.78
66.64
60.35
20.73
24.82
36.95
51.48
27.22
25.45
25.61
27.93
42.05
52.96
33.22
30.17
17.70
20.88
31.68
40.94
18.42
29.91
50.35
51.58
58.73
74.68
44.95
57.55
6.46
9.63
23.80
42.51
7.20
7.74
30.97
34.09
50.83
54.46
35.21
40.38
12.65
22.63
49.24
63.31
26.65
15.55

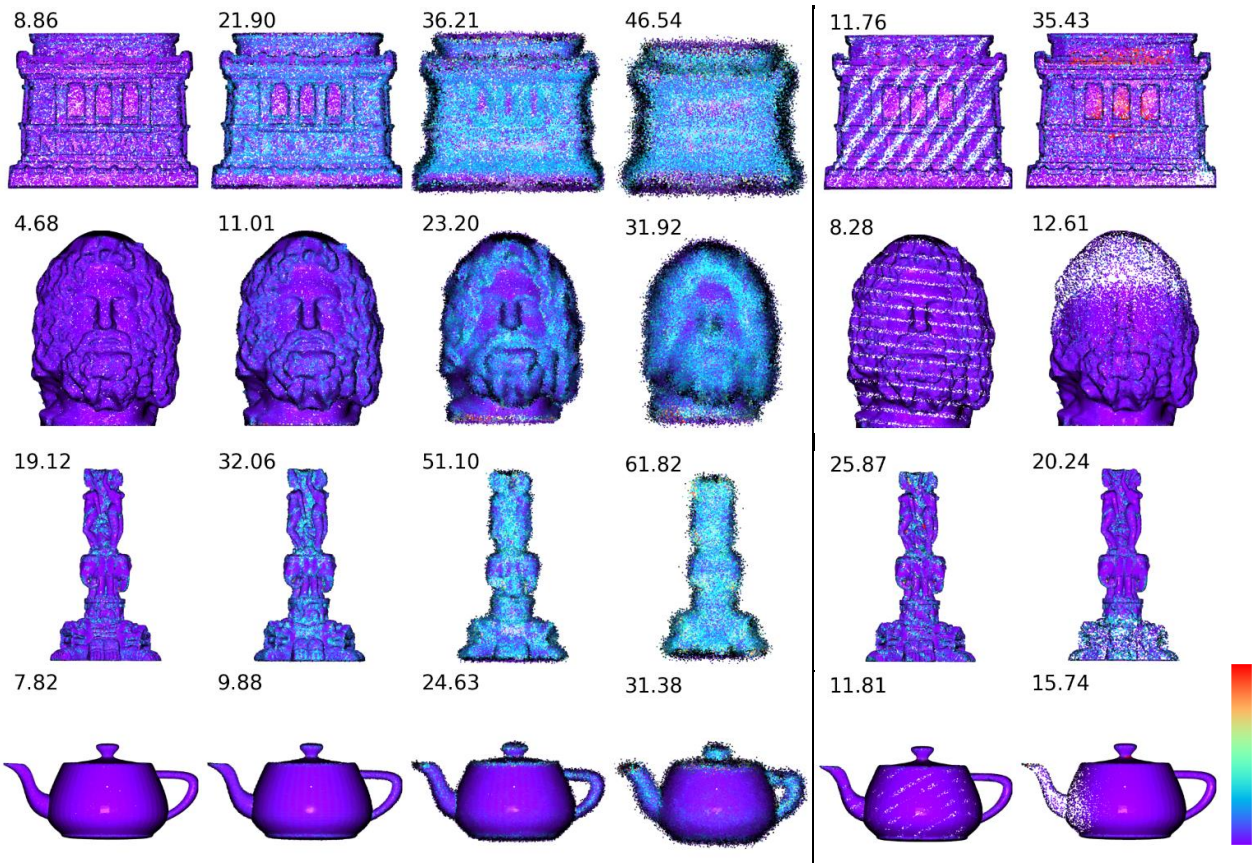


Figure 14. (Same as the previous figure.) Error visualization of oriented normals estimated by our method on the FamousShape dataset. We map the normal errors to a heatmap ranging from 0° to 180° for visualization. The average RMSE values are reported for each point cloud.