Nesti-Net: Normal Estimation for Unstructured 3D Point Clouds using Convolutional Neural Networks

Supplementary Material

A. Scale selection methods: architecture details

In Section 4.4 we report the performance of several ablations of our method. Here we detail the architecture of the following ablations:

- ss - A single scale version which directly feeds a 3DmFV representation into a CNN architecture (a single-scale MuPS); see Table 3.
- ms - A multi-scale version which feeds the full MuPS representation into a CNN architecture; see Table 3.
- ms-sw - A multi scale version which first attempts to estimate the noise level and then feeds the 3DmFV representation of the corresponding input scale into two sub-networks for two noise levels (switching). Note that for this version, the noise level is provided during training and we use a predetermined threshold for the sub-network selection; see Table 4.

B. Normal estimation performance analysis

We show here additional results from section 4.3. Figure 8 shows normal vectors mapped to RGB color at each point. Figure 10 shows the angular error mapped to a heatmap between 0-60. The number above each point cloud is its RMS error. Figure 9 shows the expert (scale) prediction by assigning a color to each expert and visualizing the chosen expert color over the point cloud.

ss	**ms**
3D Inception(3,5,128)	3D Inception(3,5,128)
3D Inception(3,5,256)	3D Inception(3,5,256)
3D Inception(3,5,256)	3D Inception(3,5,256)
maxpool(2,2,2)	maxpool(2,2,2)
3D Inception(3,5,512)	3D Inception(3,4,512)
3D Inception(3,5,512)	3D Inception(3,4,512)
maxpool(2,2,2)	maxpool(2,2,2)
FC(1024)	FC(1024)
FC(256)	FC(256)
FC(128)	FC(128)
FC(3)	FC(3)

Table 3. Ablation architecture details for single-scale (ss) and multi-scale (ms).

In Section 4.5 we report the normal estimation of Nesti-Net and compare it qualitatively to the PCA results with medium scale. For additional comparison, 11 shows results of PCA for small and large scale. It shows that a small scale produces a noisy output and a large scale over-smooths fine details and corners.

In Section 4.3 we report the RMS error metric results for comparison to other methods. The RMS error favors averaging methods. For example, near corners, it will reward a method that estimates an average normal direction rather than a method that estimates the normal of the wrong plane. Therefore, a complimentary metric is required to negate this effect. We use the proportion of good points metric (PGPα), which computes the percentage of points with an error less than α; *e.g.*, PGP10 computes the percentage of points with angular error of less than 10 degrees. Table 5 reports the results of PGP10 and Table 6 the results of PGP5 for the baseline methods compared to Nesti-Net.

We further investigate the performance of all methods near sharp features. by subdividing the testset points into two subgroups according to their ground truth curvatures: (a)sharp features, (b)smooth surfaces. Table 7 shows that Nesti-Net outperforms state-of-the-art on both subgroups with significant PGP5 improvement of 19.6% vs prior best method for sharp features. We compared to the best performance of each method's variants as well as a multi-scale approach for both Jet and PCA (choosing the best radius).

We further demonstrate the normal smoothing improvement by depicting a Poisson reconstruction in Figure 12.

ms-sw	
noise estimation net	normal estimation net
3D Inception(3,5,128)	3D Inception(3,5,128)
3D Inception(3,5,256)	3D Inception(3,5,256)
3D Inception(3,5,256)	3D Inception(3,5,256)
maxpool(2,2,2)	maxpool(2,2,2)
3D Inception(3,5,512)	3D Inception(3,4,512)
3D Inception(3,5,512)	3D Inception(3,4,512)
maxpool(2,2,2)	maxpool(2,2,2)
FC(1024)	FC(1024)
FC(256)	FC(256)
FC(128)	FC(128)
FC(1)	FC(3)

Table 4. Ablation architecture details for multi-scale with switching. First the noise is estimated and then the input is fed into the corresponding scale network according to a threshold. The network for both scales is constructed identically.

Aug.	**PCPNet** [11]		**Jet** [7]			**PCA** [12]			**NestiNet**
Scale	ss	ms	small	med	large	small	med	large	MoE
None	0.8364	0.8404	0.8802	0.7509	0.6584	0.8686	0.7409	0.6606	**0.9120**
Noise									
$\sigma = 0.00125$	0.8013	0.8031	0.7346	0.7447	0.6575	0.7712	0.7378	0.6598	**0.8384**
$\sigma = 0.006$	0.6667	0.6294	0.1006	0.6397	0.6311	0.1101	0.6402	0.6301	**0.7164**
$\sigma = 0.01$	0.5546	0.5124	0.0377	0.3827	0.547	0.04063	0.394	0.5462	**0.6123**
Density									
Gradient	0.7801	0.8062	0.8848	0.7695	0.6401	0.8731	0.7624	0.6366	**0.9003**
Striped	0.7967	0.8076	0.8743	0.7504	0.6001	0.8609	0.7379	0.5879	**0.8929**
Average	0.7393	0.7332	0.5854	0.6730	0.6224	0.5874	0.6689	0.6202	**0.8120**

Table 5. Normal estimation results comparison using the PGP10 metric (higher is better).

Aug.	**PCPNet** [11]		**Jet** [7]			**PCA** [12]			**NestiNet**
Scale	ss	ms	small	med	large	small	med	large	MoE
None	0.7078	0.6986	0.7905	0.6284	0.5395	0.7756	0.6192	0.5361	**0.8057**
Noise									
$\sigma = 0.00125$	0.6245	0.5932	0.4132	0.6237	0.5377	0.4758	0.6157	0.5335	**0.6611**
$\sigma = 0.006$	0.4486	0.366	0.027	0.4152	0.4837	0.02998	0.42	0.4812	**0.5618**
$\sigma = 0.01$	0.3156	0.2482	0.0099	0.1462	0.3715	0.0104	0.154	0.3719	**0.399**
Density									
Gradient	0.6065	0.6254	**0.7883**	0.6442	0.4976	0.7743	0.647	0.4894	0.7749
Striped	0.6126	0.6231	**0.7753**	0.6321	0.4598	0.7575	0.6174	0.4415	0.7676
Average	0.5526	0.5257	0.4674	0.5150	0.4816	0.4706	0.5122	0.4756	**0.6617**

Table 6. Normal estimation results comparison using the PGP5 metric (higher is better).

		Ours	**pcpnet**	**jet**	**jet ms**	**pca**	**pca ms**	**Hough**	Improve (%)
PGP5	(a)	**0.39**	0.24	0.29	0.31	0.28	0.30	0.33	19.6
	(b)	**0.77**	0.65	0.63	0.74	0.62	0.73	0.60	3.05

Table 7. Nesti-Net normal estimation PGP5 performance for subsets containing (a) sharp features and (b) smooth surfaces. Percentage of relative improvement is given in the right column.

C. Time complexity and timing

We subdivide Nesti-Net's time complexity into its two main stages: MuPS computation and normal estimation. It was shown in [4] that the time complexity of 3DmFV is $O(KT)$. Here K is the number of Gaussians and T is the number of points in the point cloud. MuPS computes 3DmFV of n scales (point subsets) containing a maximum of T_{max} points. Therefore, its time complexity is $O(nKT_{max})$. The time complexity of the normal estimation network is constant and proportional to the number of operators in the network. Adding experts to the network increases training time but does not affect test time since only one expert is evaluated during test time. Adding additional scales, however, affects the scale manager network by introducing additional operators. Nevertheless, the normal estimation time is independent of the number of points. We report the time performance of our method and its ablations in Figure 13. It includes timing results for single-scale (ss), multi-scale (ms) and mixture-of-experts (Nesti-Net) using 8^3 Gaussians and 3^3 Gaussians in the 'light' versions. Timing is measured as a function of the number of points within each scale. Figure 13 shows that choosing a lower number of Gaussians for the MuPS representation significantly improves speed but introduces a tradeoff with accuracy. For example, the average RMS error of 'Nesti-Net light' is 13.5, which is still superior to all other methods but by a smaller margin. We also report the timing results of different methods in Figure 14 and compare to our 'light' version. Note that the methods were implemented using different frameworks; PCA and Jet were implemented as part of the CGAL library, PCPNet uses pytorch, HoughCNN uses Cuda code directly, and Nesti-Net was implemented using TensorFlow. All measurements were performed on the same machine with a quad-core Intel i7-4770 CPU, 16GB RAM, and an Nvidia GTX 1080 GPU. The figure shows that PCA and Jet are the fastest methods (PCA is slightly faster) and that the learning-based approaches are comparable. All of the results are outside the range of real-time performance. The figure also shows that our method's timing is not as sensitive to the number of points within each scale as the other methods.

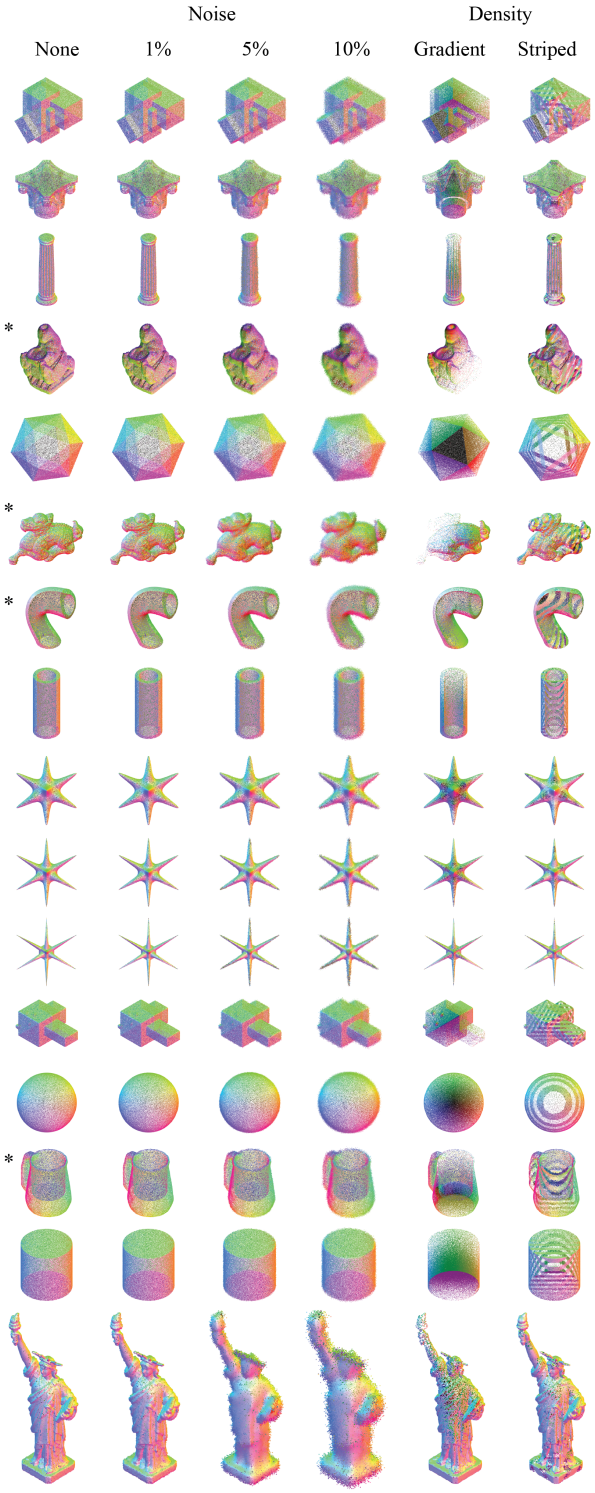

Figure 8. Nesti-Net normal prediction results for different noise levels (columns 1-4) and density distortions (columns 5-6). The point colors are normal vectors mapped to RGB. Point clouds in rows marked with * were rotated for a better view angle.

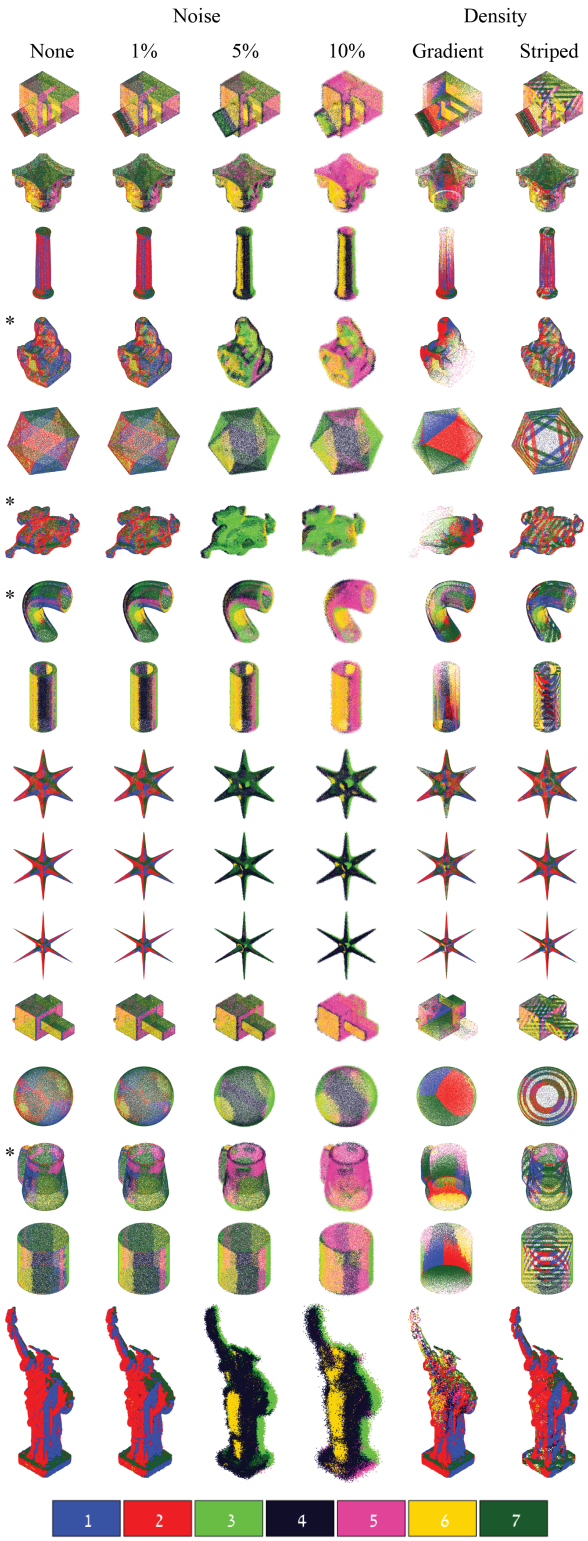

Figure 9. Nesti-Net predicted experts (scales). Each color represents the predicted expert for optimal normal estimation. Color coding is given at the bottom. Point clouds in rows marked with * were rotated for a better view angle.

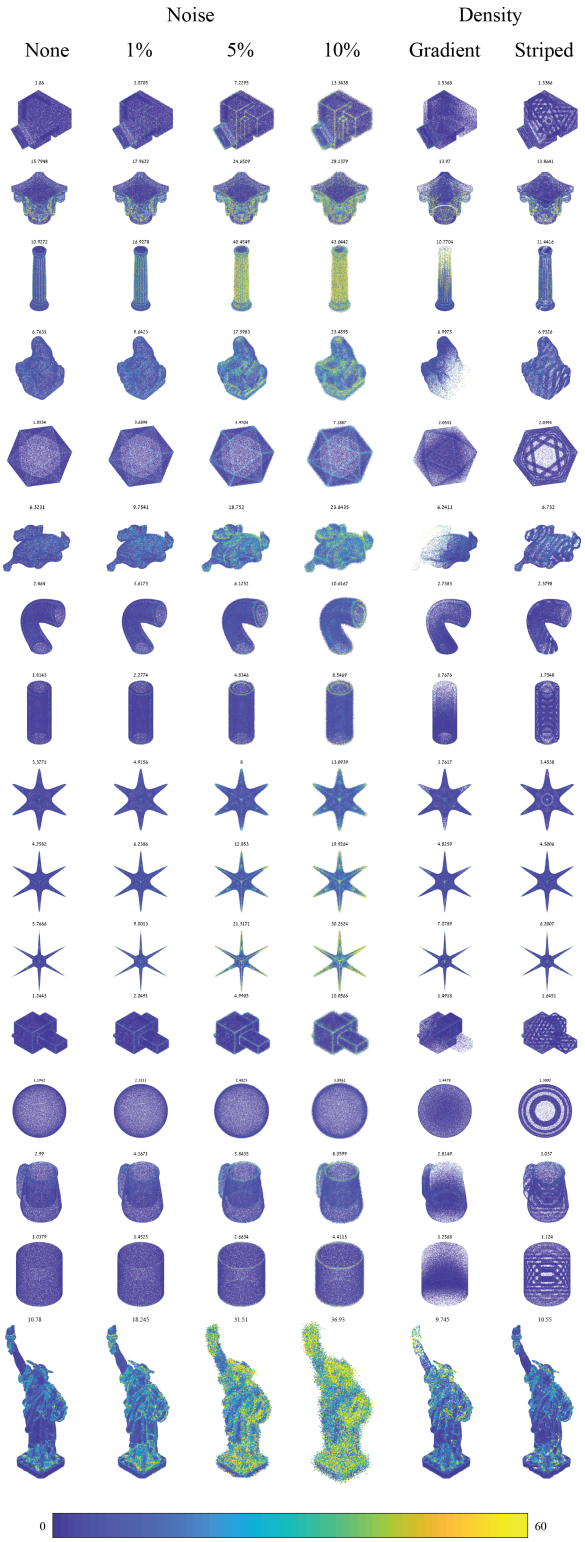

Figure 10. Normal estimation error results for NestiNet compared to other methods for different noise levels (columns 1-4) and density distortions (columns 5-6). The point colors correspond to angular difference, mapped to a heatmap between 0-60; see bottom color bar. The number above each point cloud is its RMS error.

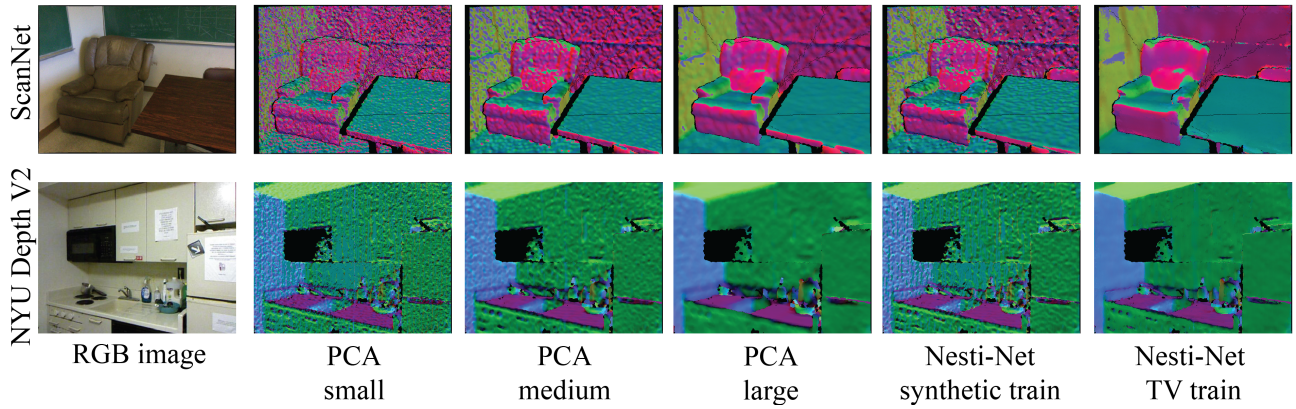

Figure 11. Normal estimation results on scanned data from the NYU Depth V2 [19] dataset and the ScanNet [8] dataset.

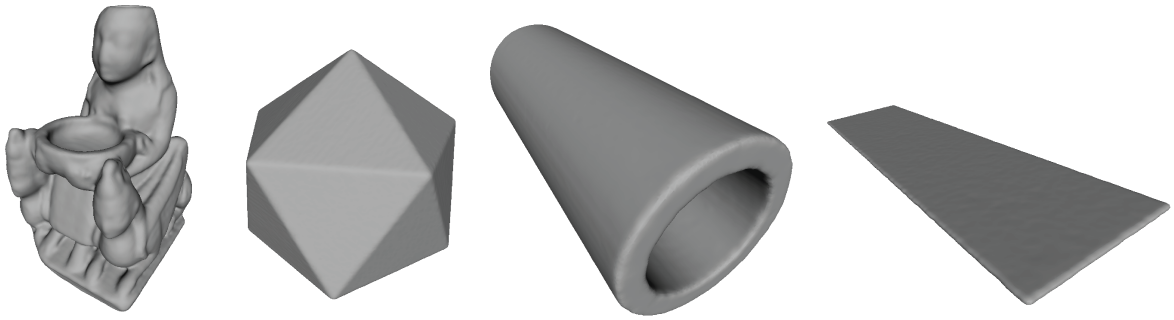

Figure 12. Poisson Reconstruction using normals estimated by Nesti-Net and oriented by MST [12] on PCPNet dataset point clouds.

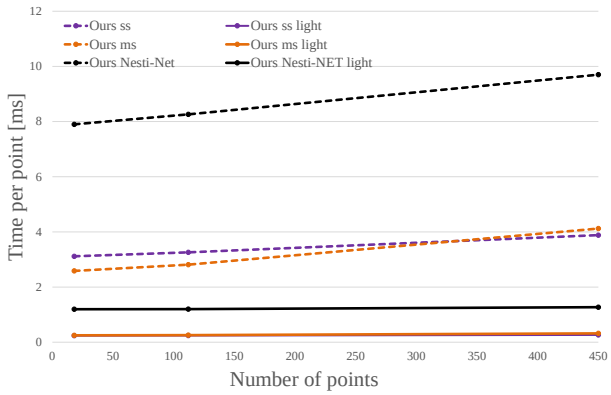

Figure 13. Timing results for our method and its ablations using 8^3 Gaussians and 3^3 Gaussians in the 'light' version. Ablations include single-scale (ss), multi-scale (ms) and mixture-of-experts (Nesti-Net). Time is measured in ms per point

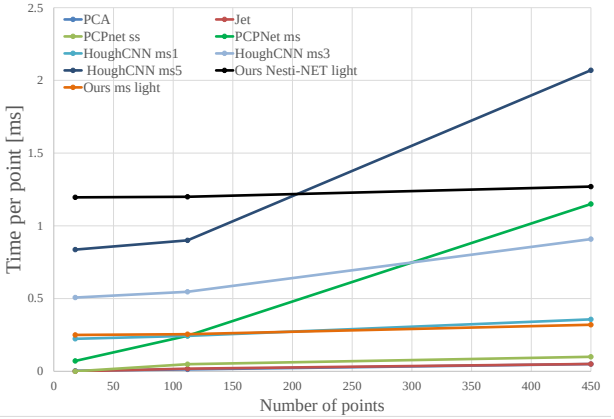

Figure 14. Timing results for normal estimating methods measured in ms per point.