GraphFit: Learning Multi-scale Graph-Convolutional Representation for Point Cloud Normal Estimation

Keqiang Li[1,3], MingYang Zhao[1,2*], Huaiyu Wu[1**], Dong-Ming Yan[1,3], Zhen Shen[1], Fei-Yue Wang[1] and Gang Xiong[1]*

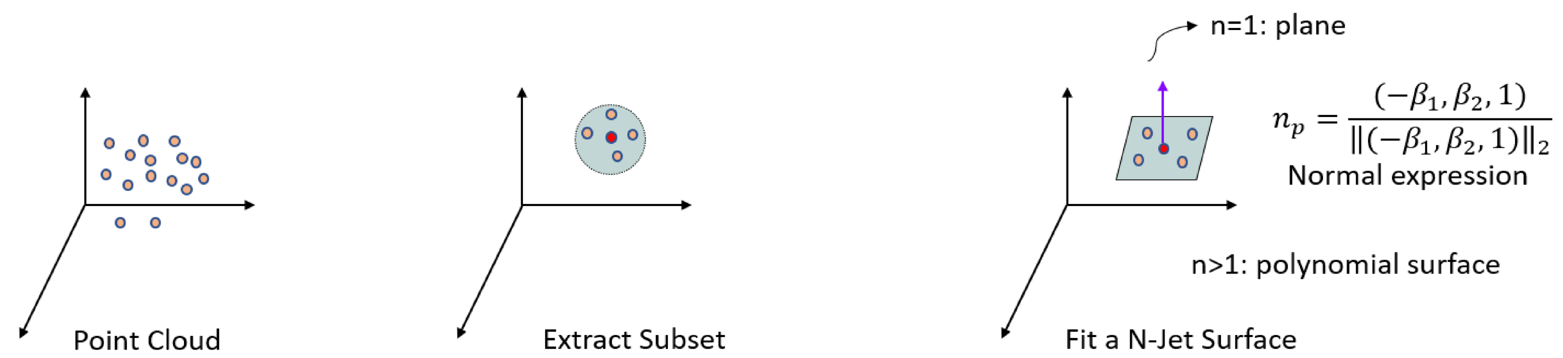
[1]Institute of Automation, CAS, [2]BAAI, [3]UCAS

ECCV TEL AVIV 2022

Introduction

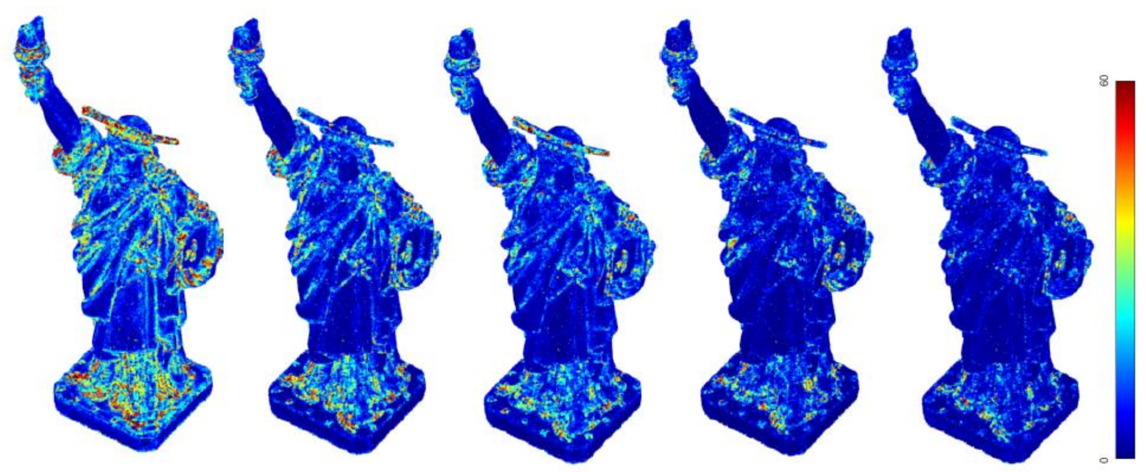
Goal

➢ Estimate the normal for unstructured point clouds



N-Jet: $f(x,y) = J_{\beta,n}(x,y) = \sum_{k=0}^{n}\sum_{j=0}^{k}\beta_{k-j,j}x^{k-j}y^{j}$

➢ Recent methods DeepFit and AdaFit employ a DNN to learn point-wise weights for WLS fitting, leading to good normal estimation quality, precise normal estimation in complex regions are still difficult



(a) PCPNet (b) NestiNet (c) DeepFit (d) AdaFit (e) Ours

Insights

➢ Previous methods directly adopt patches for normal estimation, and usually ignore intrinsic relationship between points in the same patch

Contribution

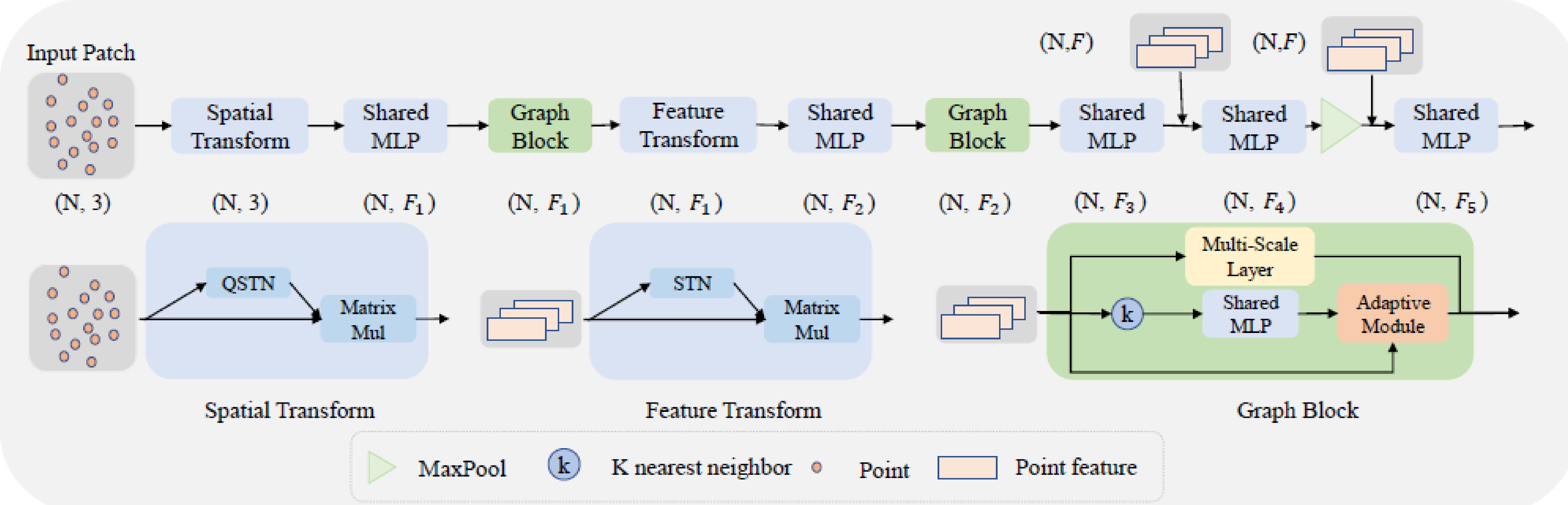
➢ We propose a new method for accurate and robust normal estimation via the graph-convolutional feature learning

➢ We design an adaptive module using the attention mechanism to fuse the point features with its neighboring features

➢ We introduce a multi-scale representation module to extract more expressive features

Method

Overview: GraphFit

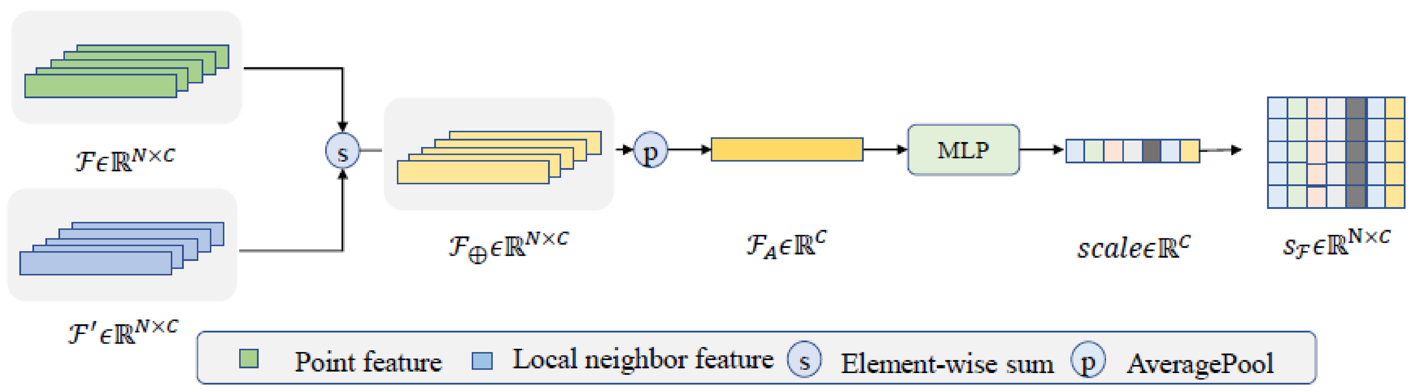


Graph Block

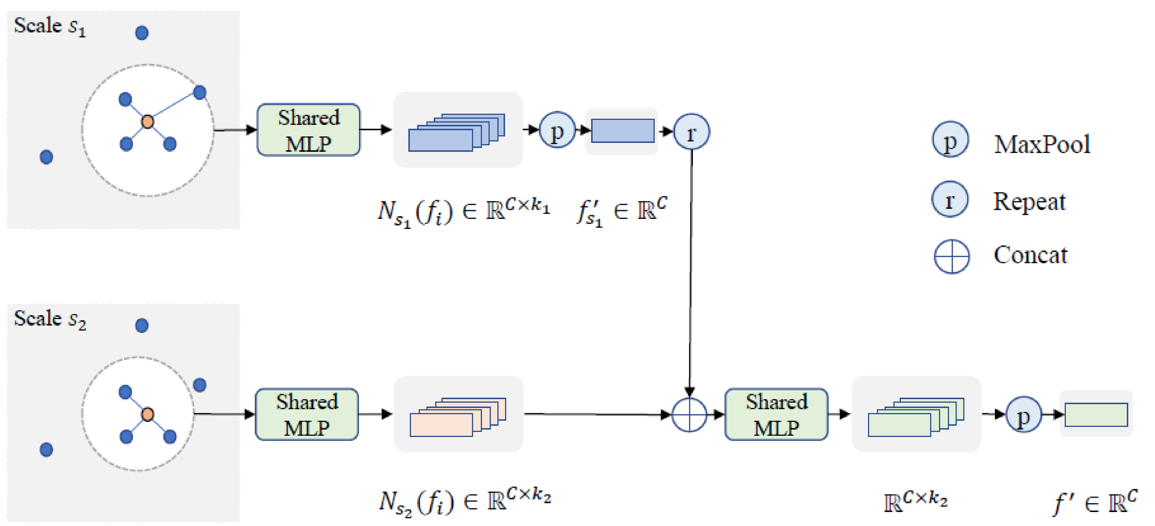
➢Feature $F = \{f_i | i = 1,2,\ldots,N_p\} \in \mathbb{R}^{N_p\times C}$ correspond to the input patch $N_k(p_i) \in \mathbb{R}^{N_p\times 3}$. Like DGCNN[1], we get local neighborhood information f_i' for each input feature f_i

$$g_{ijc} = \phi_c(\Delta f_{ij}), j \in N(i) \quad f_i' = \max_{j\in N(i)} g_{ij}$$

Adaptive Module



Multi-scale Layer

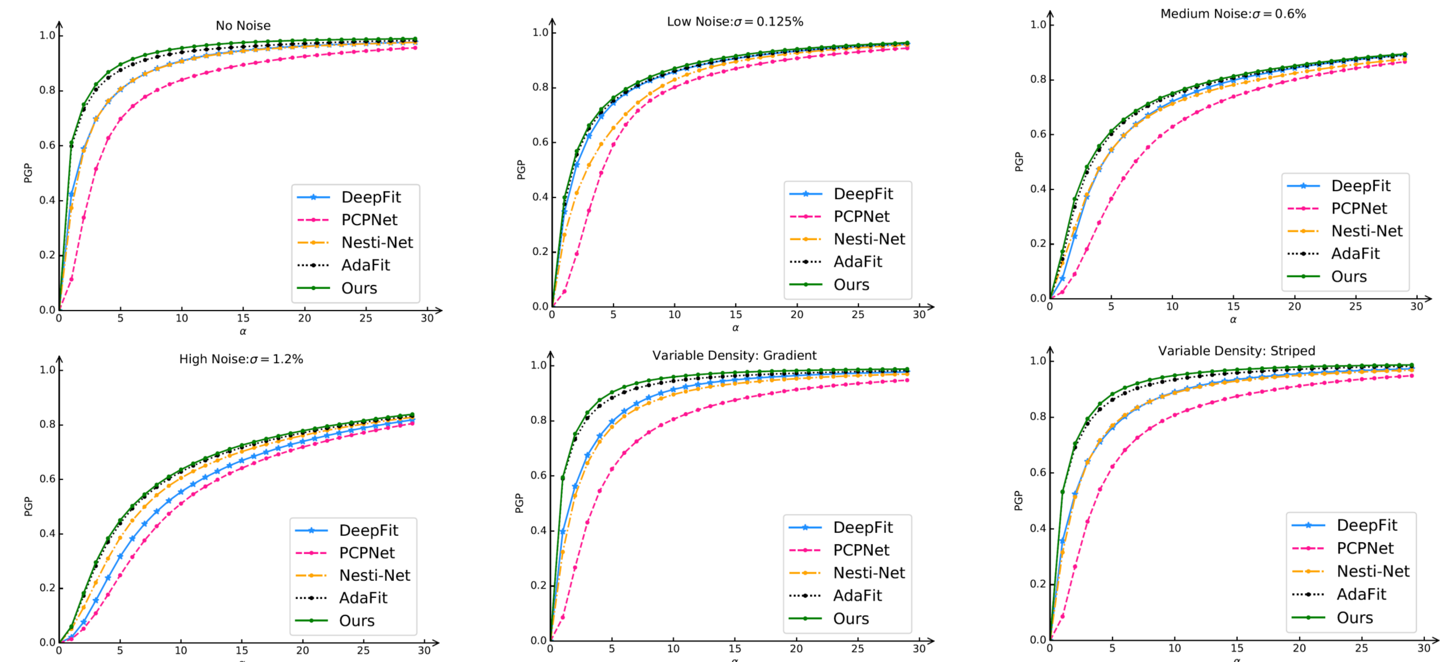


[1] Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph CNN for learning on point clouds. ACM Transactions On Graphics. 38(5), 1–12 (2019)

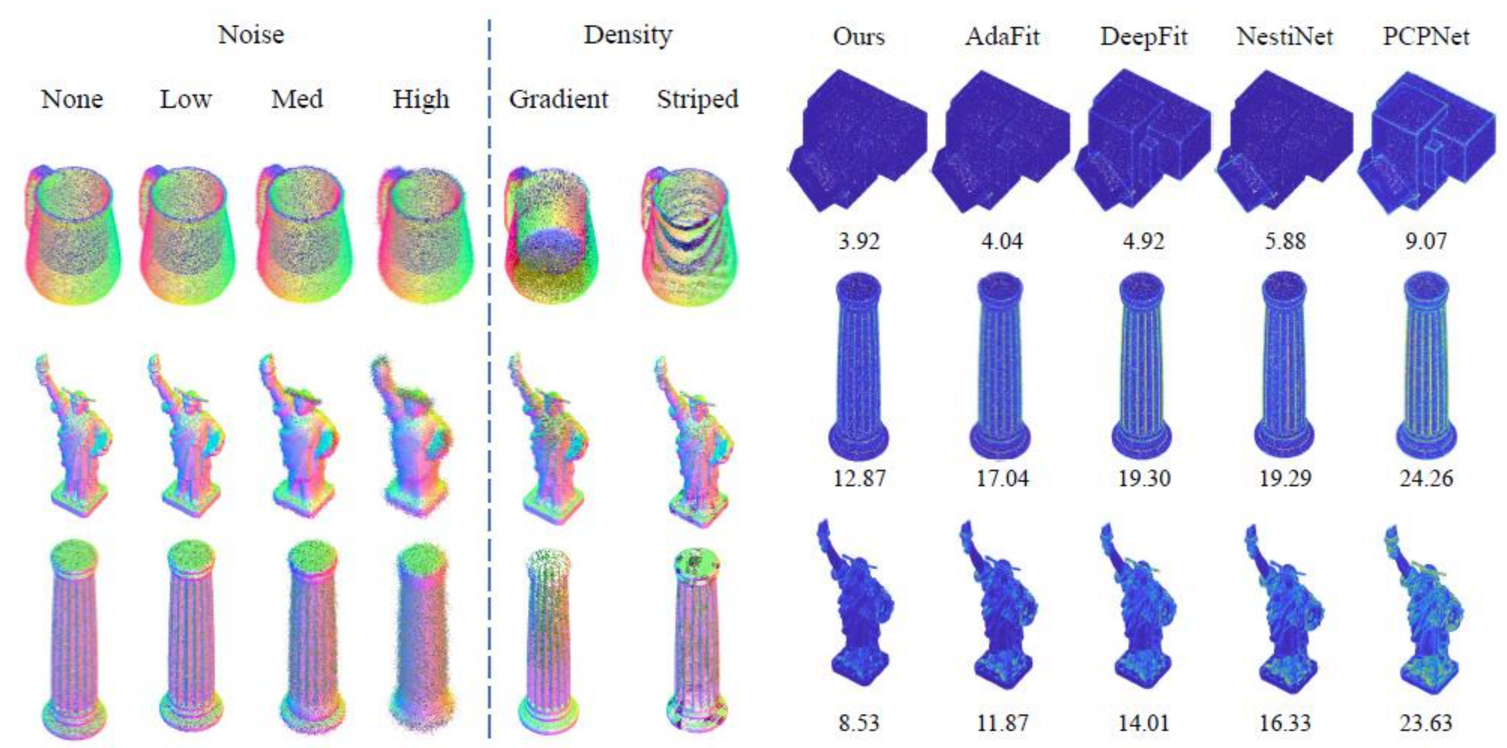
Experiments

➢GraphFit achieves the SOTA on common benchmarks.

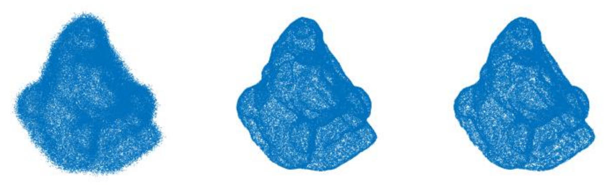
Aug.	Ours	AdaFit	DeepFit	IterNet	Nesti-Net	PCPNet	Jet	PCA
w/o Noise	**4.45**	5.19	6.51	6.72	6.99	9.62	12.25	12.29
$\sigma = 0.125\%$	**8.74**	9.05	9.21	9.95	10.11	11.37	12.84	12.87
$\sigma = 0.6\%$	**16.05**	16.44	16.72	17.18	17.63	18.87	18.33	18.38
$\sigma = 1.2\%$	**21.64**	21.94	23.12	21.96	22.28	23.28	27.68	27.5
Gradient	**5.22**	5.90	7.31	7.73	9.00	11.70	13.13	12.81
Striped	**5.48**	6.01	7.92	7.51	8.47	11.16	13.39	13.66
Average	**10.26**	10.76	11.80	11.84	12.41	14.34	16.29	16.25



Visualization Results



Applications



(a) Input (b) Ours (c) GT

noise removal

(a) Input (b) Ours (c) Groundtruth

surface reconstruction

CASIA BAAI

Paper and Code

https://arxiv.org/abs/2207.11484

https://github.com/UestcJay/GraphFit