

通过对比学习优化深度拟合的点云法向估计

史路冰*, 聂明辉, 张 杰

辽宁师范大学数学学院, 辽宁 大连

收稿日期: 2023年6月13日; 录用日期: 2023年7月19日; 发布日期: 2023年7月26日

摘 要

近年来, 以当前点的邻域作为输入, 利用深度网络估计逐点权重进行加权最小二乘曲面拟合的算法, 在法向估计上取得了当前领先的结果。本文算法在此基础上, 引入对比学习的思想, 即先通过真实法向的差异构造三元组, 再根据构造的三元组创建三元组损失, 用以对邻域内的每点权重进行约束, 从而提高权重的估计质量, 获得更优的法向估计结果。实验结果表明, 本文算法能够对已有的算法有较好的提高, 在PCPNet数据集上的均方误差平均值为11.71。

关键词

深度学习, 对比学习, 法向估计

Optimization of Point Cloud Normal Estimation for Deep Fitting via Contrastive Learning

Lubing Shi*, Minghui Nie, Jie Zhang

School of Mathematics, Liaoning Normal University, Dalian Liaoning

Received: Jun. 13th, 2023; accepted: Jul. 19th, 2023; published: Jul. 26th, 2023

Abstract

In recent years, algorithms that use the neighborhood of the current point as input and use deep networks to estimate point-wise weights for weighted least squares surface fitting have achieved leading results in normal estimation. On this basis, the algorithm in this paper introduces the idea of contrastive learning, which first constructs triplets based on the differences in real normal di-

*通讯作者。

rections, and then creates triplet losses based on the constructed triplets to constrain the weights of each point in the neighborhood, thereby improving the estimation quality of weights and obtaining better normal estimation results. The experimental results show that the proposed algorithm can significantly improve existing algorithms, with an average mean square error of 11.71 on the PCPNet dataset.

Keywords

Deep Learning, Contrastive Learning, Normal Estimation

Copyright © 2023 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

点云的法向估计是 3D 计算机视觉和计算机图形学中的一个基本问题。众多研究结果表明, 点云的可靠法向信息对于表面的重建、渲染、分割、抓取等任务具有重要意义。然而, 一方面由于扫描仪得到的点云数据缺乏连接关系, 另一方面由于受到噪声、遮挡等因素影响, 准确并快速地估计出点云可靠的法向依旧是一项具有挑战性的工作。

现有的法向估计的算法大致分为传统的算法[1]-[6]和基于深度学习的算法[7] [8] [9] [10]。传统的算法在很大程度上依赖于人工提取的特征和人为设定的规则, 因此无法自适应用于法向估计的公共数据集中的各种点云。在实际应用中, 传统算法的用户需要手动调节参数, 时间成本高昂。特别是在用户不清楚点云数据中所包含的噪声大小和采样密度时, 参数调节将更加耗时。随着神经网络在点云处理中的成功应用, 基于深度学习的点云法向估计算法取得了当前领先的结果。早期基于深度学习的算法将不规则的点云投射到规则的空间域中, 并用卷积神经网络(convolutional neural network, CNN)进行特征学习[7] [8]。然而, 将不规则的点云投射到规则的空间域中会有一些信息损失。为了克服这个问题, 近来诸多算法试图利用神经网络直接建立从不规则点云到法向的映射[9] [10]。PCPNet [9]是该类算法的先驱, 该工作受 PointNet [11]的启发, 提出了一种用于法向估计的深度多尺度框架。基于 PCPNet 框架, Zhou 等人[10]引入了一种额外的特征约束机制和多尺度邻域选择策略来提高性能。这种直接学习法向的算法, 泛化能力弱, 使得预测结果不稳定[12]。为了解决这个问题, DeepFit [13]利用网络预测邻域内的逐点权重, 然后通过加权最小二乘拟合曲面来估计法向。该工作取得当前领先的结果, 诸多工作对该框架做出改进[12] [14]。AdaFit [12]在 DeepFit 框架的基础上, 结合了 CSA (Cascaded Scale Aggregation)层来聚合来自多个邻域尺度的特征, 并添加额外的偏移量以使输出法向更加稳健和准确。G2N2 [14]在 DeepFit 框架的基础上, 提出了一种几何引导的神经网络用于法向估计。

本文算法是在 DeepFit 框架的基础上提出的改进措施。该算法利用对比学习的机制, 对 DeepFit 中的权重进行优化, 从而提升性能。为了实现这一点, 本文设计了三元组(三元组包括锚点 anchor, 正点 positive, 负点 negative)和三元组损失两部分, 使得锚点对应的权值远离负点对应的权值并与正点对应的权值相近。三元组损失主要用于优化锚点与正样本和负样本的相对距离。在本文中, 三元组损失通过计算锚点与正点和负点权重的相对差异, 优化每点权重。本文给出的策略适用于所有基于 DeepFit 框架的算法, 文中以文献[13]为基础给出结果, 对于 AdaFit 和 G2N2 也一样适用。

2. 算法

对于给定的点云 $P = \{p_i\}_{i=1}^N$ 和任一当前点 $p_i \in P$, 本文以 p_i 的邻域 $N_K(p_i)$, 即 $N_K(p_i) = \{p_i' | p_i' \text{ 是点 } p_i \text{ 的 } K \text{ 个近邻点}\}$, 作为输入, 在 DeepFit [13] 框架的基础之上, 引入对比学习来优化邻域内的每点权重, 进而输出当前点的法向。在对比学习中, 最核心的部分就是通过构造三元组来刻画点与点特征之间的相似程度。在三元组构造过程中, 首先要找到合适的锚点, 其次再构造与之对应的正点和负点, 其中和锚点属于同一类的点(特征相似的点)称为正点, 和锚点不属于同一类的点(特征不同的点)称为负点。本文算法流程如图 1 所示。

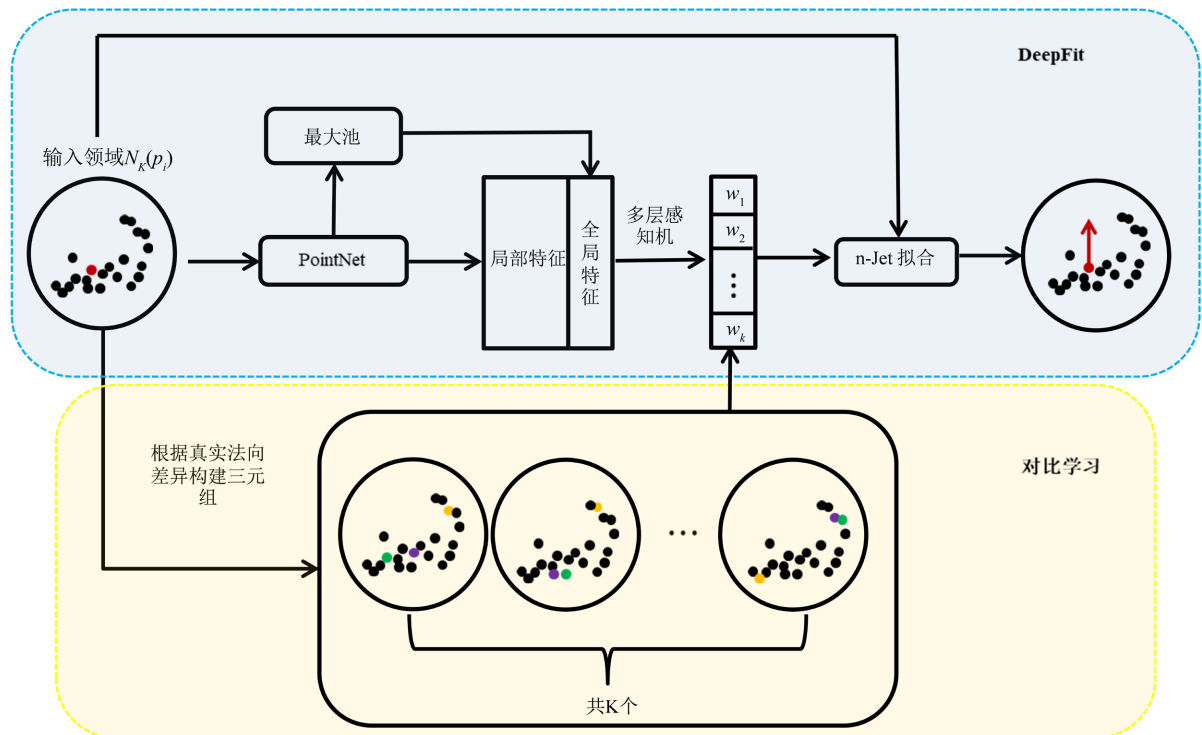


Figure 1. Algorithm flowchart. The upper part (light blue part) is the work of DeepFit, and the lower part (light yellow part) is the work of comparative learning introduced. The red dot represents the current point, the purple dot represents the anchor point, the orange dot represents a randomly selected negative point corresponding to the anchor point, and the green dot represents a randomly selected positive point corresponding to the anchor point

图 1. 算法流程图。上半部分(浅蓝色部分)是 DeepFit 的工作, 下半部分(浅黄色部分)是引入的对比学习的工作。红点表示当前点, 紫点表示锚点, 橙点表示随机选出的与该锚点对应的一个负点, 绿点表示随机选出的与该锚点对应的一个正点

2.1. DeepFit 工作的简要回顾

首先, DeepFit 先对每个点的局部子集 $N_K(p_i)$ 进行预处理, 即对 $N_K(p_i)$ 进行中心化和对齐操作。具体来说, 对于中心化操作, DeepFit 先计算 $N_K(p_i)$ 的中心点, 之后将邻域内所有点的坐标减去中心点的坐标进行平移。对于对齐操作, DeepFit 先对点云进行主成分分析计算局部坐标系, 之后将整个邻域旋转到局部坐标系中。为了简便起见, 本文亦将预处理后的局部子集记为 $N_K(p_i)$ 。然后, DeepFit 再将预处理后的局部子集 $N_K(p_i)$ 输入到 PointNet 网络中, 得到局部特征表示和全局特征表示, 之后将得到的特征表示连接起来一起送入到多层感知机(Multilayer Perceptron, MLP)中, 得到该邻域内每点的权重 $W = \{w_j | j=1, \dots, K\}$ 。最后, DeepFit 通过加权最小二乘拟合获得法向 \hat{n}_i 。

DeepFit 的损失函数为:

$$L_D = |n_{GT} \times \hat{n}_i| + \alpha_1 L_{con} + \alpha_2 L_{reg} \quad (1)$$

其中, n_{GT} 是真实法向, \hat{n}_i 是估计的法向, L_{con} 是一致性损失, L_{reg} 是 PointNet [11] 的正则化项, α_1 和 α_2 是加权因子, 根据经验选择。一致性损失为 $L_{con} = \frac{1}{K} \left[-\sum_{j=1}^K \log(w_j) + \sum_{j=1}^K w_j |n_{GT} \times \hat{n}_j| \right]$ 。正则化项为 $L_{reg} = |I - \mathbf{A}\mathbf{A}^T|$, 其中 \mathbf{A} 表示 PointNet 中的旋转矩阵, 是正交矩阵。 \mathbf{A}^T 表示 \mathbf{A} 的转置。

2.2. 本文算法

为了提高 DeepFit 在权重学习上的准确率, 本文通过对比学习来优化邻域内每点的权重, 从而提高法向估计的结果。在对比学习中, 最核心的部分是三元组的构造方式。

2.2.1. 三元组构造

本文将每个数据样本都看成一个锚点, 试图在该点的邻域内找寻与该锚点相对应的一个正样本和一个负样本, 这样便构建了一个三元组。具体来说, 本文将输入的局部邻域 $N_K(p_i)$ 内的每个点都视为一个锚点, 通过计算锚点与邻域内其他点的真实法向角度差选取正点和负点, 构成一个三元组。对于锚点 p_j , 在给定的邻域 $N_K(p_i)$ 内, 满足与该锚点相对应的正、负点选取条件的点存在不止一个的情况, 故本文将满足选取正点条件的点组成一个集合, 称为正样本点集, 设为 S_j , 将满足负点条件的点组成负样本点集, 设为 T_j 。根据如下定义得到正、负样本点集:

$$S_j = \left\{ p_l \mid \theta(n_{p_j}, n_{p_l}) < \theta_{th1}, p_l \in N_K(p_i) \right\} \quad (2)$$

$$T_j = \left\{ p_l \mid \theta(n_{p_j}, n_{p_l}) > \theta_{th2}, p_l \in N_K(p_i) \right\} \quad (3)$$

其中, n_{p^*} 是 p^* 的真实法向, θ_{th1} 和 θ_{th2} 是用于区分正样本点和负样本点的角度阈值。根据经验, 本文将 θ_{th1} 设为 10 度, 将 θ_{th2} 设为 30 度。

从采集到的正、负样本点集中分别随机选择一个点作为 p_j 的一个正、负样本点, 记为 s_j 和 t_j 。当 p_i 的邻域采样自同一个曲面时, 邻域内所有点的角度差异都较小, 因此会出现邻域内的所有点与锚点同属于一类, 即负样本点集个数为 0 的情况。对于邻域内未产生负样本点集的锚点, 为了方便计算损失, 本文将自身看作该锚点的唯一负样本点。

2.2.2. 三元组的损失

对于定义每个三元组, 本文通过权重差异计算三元组损失:

$$L_{th} = \max \left\{ \left| w_{p_j} - w_{s_j} \right| - \left| w_{p_j} - w_{t_j} \right| + m, 0 \right\} \quad (4)$$

其中 w_{p_j} 、 w_{s_j} 、 w_{t_j} 分别是锚点的权重和其对应的正、负样本点的权重。根据经验, 本文将 m 设为 0.5。

本文的损失函数是在 DeepFit 的基础上额外增加了一个三元组损失, 定义如下:

$$L = L_D + \alpha_3 L_{th} \quad (5)$$

其中, α_3 是加权因子, 根据经验设置为 0.01。

3. 实验

首先描述用于实验的数据库和评价指标, 然后与基于深度学习的算法和经典法向估计算法进行比较, 给出定性和定量的结果。

3.1. 数据库和评价指标

为了验证本文算法的有效性,本文在 PCPNet [9]形状数据集上进行测试。该数据集遵循与[12]完全相同的实验设置,包括训练数据增强以及在测试数据上添加的噪声或不同的采样机制。对于训练数据集,PCPNet 使用了 8 种形状,所有形状都以三角网格的形式给出。对于每个网格,PCPNet 对其添加标准差分别为形状边界框对角线长度的 0.12%、0.6%和 1.2%的高斯噪声,分别生成三种不同噪声的点云。对于测试数据,使用了 22 种形状,每个形状都是由 100 k 个点组成的,其处理算法与训练数据一样。与以前算法类似,本文也使用每个形状的 5000 点构成的子集进行评估。

本文与深度学习算法(DeepFit、PCPNet)和经典法向估计算法(PCA [1])进行比较。为了体现定量比较结果,本文遵循[12]的设置,使用预测法向与真实法向之间的角度均方根误差(RMSE)为评价指标。算法在 PyTorch 上实现,并在一个 Nvidia GTX 1080Ti 上进行训练。

3.2. 定量比较

与 DeepFit 类似,本文分析了在两种数据损坏(非均匀采样和点扰动)下算法的鲁棒性。非均匀采样是指对点子集选择梯度和条纹两种采样机制,点扰动是指将高斯噪声添加到点坐标中。对于 PCPNet 算法,本文展示了单尺度和多尺度版本的结果。对于 PCA 算法,本文展示了三种不同尺度的结果,分别是小、中和大,它们对应于 18、112、450 个最近邻。表 1 显示了采用不同算法法向估计的结果。结果表明,本文算法在所有设置中均优于其他算法,这证明了对比学习对于深度拟合算法起到一定的提升作用。

Table 1. The comparison of RMSE angle error between this algorithm and other algorithms shows the best result in bold font
表 1. 本文算法与其他算法的 RMSE 角度误差的比较,最佳结果用粗体表示

算法	本文算法	DeepFit	PCPNet		PCA		
尺度	单尺度	单尺度	单尺度	多尺度	小	中	大
无噪声	6.45	6.51	9.68	9.62	8.31	12.29	16.77
0.00125	9.05	9.21	11.46	11.37	12.00	12.87	16.87
0.006	16.66	16.72	18.26	18.87	40.36	18.38	18.94
0.012	23.02	23.12	22.80	23.28	52.63	27.50	23.50
梯度	7.27	7.31	13.42	11.70	9.14	12.81	17.26
条纹	7.84	7.92	11.74	11.16	9.42	13.66	19.87
平均	11.71	11.80	14.56	14.34	21.97	16.25	18.87

3.3. 定性比较

图 2 展示了在三种噪声尺度下本文算法与 DeepFit 算法的可视化结果。添加的噪声为高斯噪声,其标准差为原始模型边界框对角线的 0.12%、0.6%和 1.2%。本文计算每个形状的均方根值(RMS)以及每个点的真实法向与预测法向的角度差,根据计算出的角度差将点进行不同颜色的标记。本文将角度差在 0~10 度的点标记为蓝色,10~20 度标记为青色,20~90 度标记为红色。在图 2 中,PGP_10 代表真实法向与预测法向角度误差在 10 度以下的点所占的比例,PGP_20 代表角度误差在 20 度以下的点所占的比例。由此可见,PGP_10 和 PGP_20 的数值越高,实验结果越好。针对图 2,能够观察到,本文算法对于 DeepFit 算法具有一定的提升作用,尤其是当噪声尺度较大时,提升效果较为明显(如图 2 中最后一行第一个形状)。

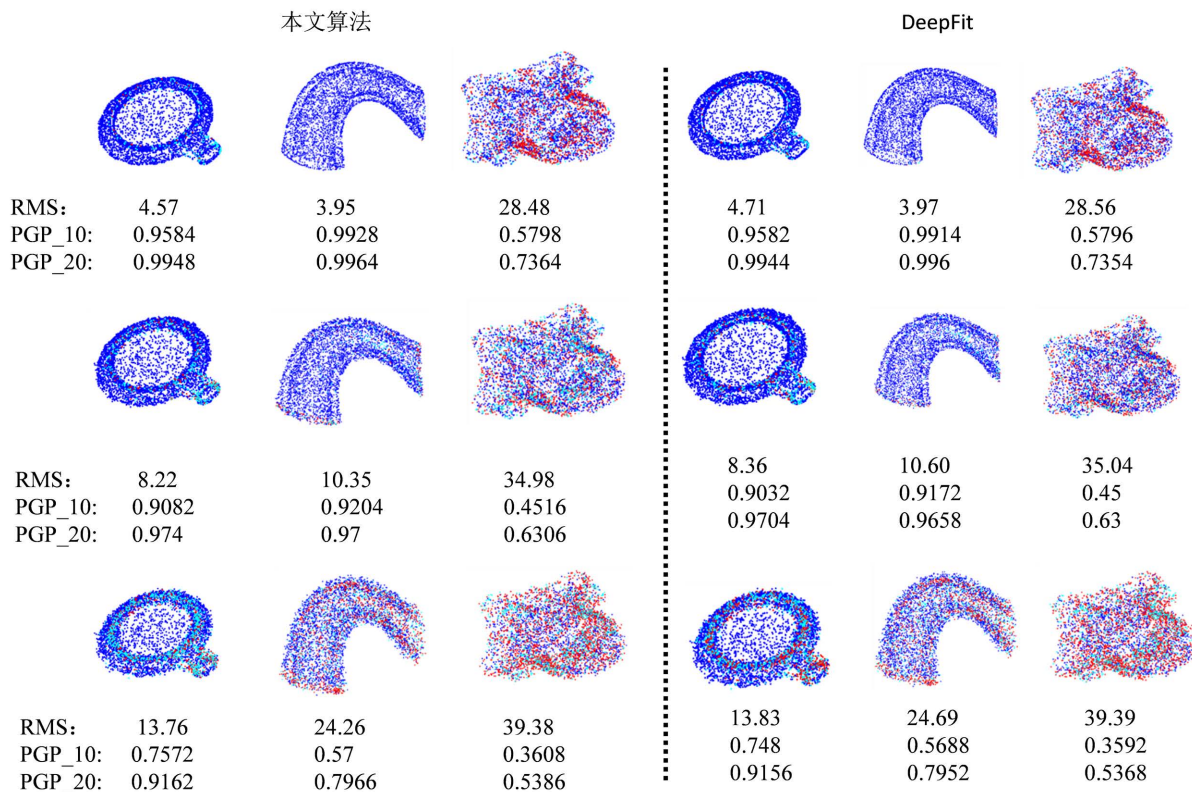


Figure 2. Visualization comparison of three shapes, with 0.12% noise added to the first row shape, 0.6% noise added to the second row shape, and 1.2% noise added to the third row shape

图 2. 3 种形状的可视化比较, 其中第一行形状添加 0.12% 噪声, 第二行形状添加 0.6% 噪声, 第三行形状添加 1.2% 噪声

4. 结论

本文提出了针对 DeepFit 框架的改进算法。该算法在 DeepFit 框架的基础之上, 利用对比学习的机制, 即通过构建三元组和三元组损失, 对深度网络提取的权重进行优化, 提高了法向估计的结果。本文在 PCPNet 数据集上进行了实验。由实验结果得知, 本文的算法对法向估计的结果有一定的提升, 尤其是对有噪声的模型会产生更好的结果。此外, 本文算法在无噪声的模型上提高效果并不明显, 针对无噪声数据如何提高法向估计结果, 是未来的一项重要工作。

基金项目

辽宁省教育厅科学技术研究项目(LJKMZ20221434)。

参考文献

- [1] Hoppe, H., DeRose, T., Duchamp, T., et al. (1992) Surface Reconstruction from Unorganized Points. *ACM SIGGRAPH Computer Graphics*, **26**, 71-78. <https://doi.org/10.1145/142920.134011>
- [2] Pauly, M., Keiser, R., Kobbelt, L.P., et al. (2003) Shape Modeling with Point-Sampled Geometry. *ACM Transactions on Graphics*, **22**, 641-650. <https://doi.org/10.1145/882262.882319>
- [3] Amenta, N. and Bern, M. (1999) Surface Reconstruction by Voronoi Filtering. *Discrete & Computational Geometry*, **22**, 481-504. <https://doi.org/10.1007/PL00009475>
- [4] Dey, T.K. and Goswami, S. (2006) Provable Surface Reconstruction from Noisy Samples. *Computational Geometry Theory*, **35**, 124-141. <https://doi.org/10.1016/j.comgeo.2005.10.006>

-
- [5] Feng, H.Y. and Ouyang, D. (2005) On the Normal Vector Estimation for Point Cloud Data from Smooth Surfaces. *Computer-Aided Design*, **37**, 1071-1079. <https://doi.org/10.1016/j.cad.2004.11.005>
- [6] Jones, T.R., Durand, F. and Zwicker, M. (2004) Normal Improvement for Point Rendering. *IEEE Computer Graphics and Applications*, **24**, 53-56. <https://doi.org/10.1109/MCG.2004.14>
- [7] Boulch, A. and Marlet, R. (2016) Deep Learning for Robust Normal Estimation in Unstructured Point Clouds. *Computer Graphics Forum*, **35**, 281-290. <https://doi.org/10.1111/cgf.12983>
- [8] Lu, D.N., Lu, X.Q., Sun, Y.X., *et al.* (2020) Deep Feature-Preserving Normal Estimation for Point Cloud Filtering. *Computer-Aided Design*, **125**, Article ID: 102860. <https://doi.org/10.1016/j.cad.2020.102860>
- [9] Guerrero, P., Kleiman, Y., Ovsjanikov, M., *et al.* (2018) PCPNet: Learning Local Shape Properties from Raw Point Clouds. *Computer Graphics Forum*, **37**, 75-85. <https://doi.org/10.1111/cgf.13343>
- [10] Zhou, J., Huang, H., Liu, B., *et al.* (2020) Normal Estimation for 3D Point Clouds via Local Plane Constraint and Multi-Scale Selection. *Computer-Aided Design*, **129**, Article ID: 102916. <https://doi.org/10.1016/j.cad.2020.102916>
- [11] Qi, C.R., Su, H., Mo, K., *et al.* (2017) PointNet: Deep Learning on Point Sets for 3d Classification and Segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, 21-26 July 2017, 652-660. <https://doi.org/10.1109/CVPR.2017.16>
- [12] Zhu, R.S., Liu, Y., Dong, Z., *et al.* (2021) AdaFit: Rethinking Learning-Based Normal Estimation on Point Clouds. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Montreal, 10-17 October 2021, 6118-6127. <https://doi.org/10.1109/ICCV48922.2021.00606>
- [13] Ben-Shabat, Y. and Gould, S. (2020) DeepFit: 3d Surface Fitting via Neural Network Weighted Least Squares. *European Conference on Computer Vision*, Glasgow, 23-28 August 2020, 20-34. https://doi.org/10.1007/978-3-030-58452-8_2
- [14] Zhang, J., Cao, J.J., Zhu, H.R., *et al.* (2021) Geometry Guided Deep Surface Normal Estimation. *Computer-Aided Design*, **142**, Article ID: 103119. <https://doi.org/10.1016/j.cad.2021.103119>