

论文 Subdivision-Based Mesh Convolution Networks 复现 [2]

摘要

卷积神经网络在二维计算机视觉方面已经取得了巨大的突破。然而由于三维网格的不规则结构，很难将卷积神经网络直接应用于三维网格上。考虑到细分曲面提供了一个层次化的多分辨率结构，而且每一个面片都与三个面片相邻，这篇文章提出了 SubdivNet。这是一个新颖且多功能的 CNN 框架，能应用于满足 Loop 细分规则的三维网格。将网格面片与二维图像中的像素进行类比，本文提出网格卷积算子来聚合相邻面片的局部特征。利用面片邻域，这个卷积可以支持标准的二维卷积网络的相关概念，例如：可变内核大小、步幅和空洞卷积。基于多分辨率层次结构，本文利用池化层将四个面片合并为一个面片。相对的，利用采用上采样方法将一个面分割为四个面。由此，许多流行的二维 CNN 架构可以很容易地用于三维网格的处理。任意结构的三维网格可以通过自参数化使其满足 Loop 细分连接规则，这样 SubdivNet 便可以普适地应用于任意网格结构。

关键词：几何深度学习；卷积神经网络；细分曲面；网格处理。

1 引言

深度卷积网络 (CNN) 在二维计算机视觉方面取得了巨大的成功，导致其在各个学科中广泛应用，包括三维几何处理。本文考虑使用网格表示进行三维几何学习。多边形网格是最常见的三维数据表示形式之一，广泛应用于建模、渲染、动画、3D 打印等领域。与点云相比，网格包含拓扑信息。与体素相比，多边形网格可以有效地表示几何上下文。二维 CNN 在图像领域成功的基础在于其规则的网格结构允许类似金字塔的层次化图像特征，这使 CNN 能够通过下采样和上采样得到不同大小的特征。然而，不同于图像中像素的横平竖直结构，三维网格的结构是不规则的，这使得很难定义具有可变内核大小、步幅和空洞的标准卷积。因此图像领域中的成熟的卷积网络架构很难应用于三维网格。此外，顶点和面之间的非结构化连接使得很难为网格找到一个简单的池化操作。人们可以通过网格简化来考虑细节层次 (LOD)，但是每个层次之间的几何基元映射没有很好的定义，并且人们不能从 LOD 层次派生出直观的池化操作符。

本文提出 SubdivNet，它可以学习具有 Loop 细分序列连接性的网格的特征表示。利用面片的邻域，在网格面片上定义了一个新的卷积，支持可变的核大小、步幅和空洞。因此，我们的可以在一个大的感受野上操作。由于我们新的三角网格卷积的灵活性，在图像邻域表现优异的神经网络，如 VGG, ResNet, DeepLabv3+，可以自然地应用于三维网格上。

2 相关工作

三角形网格是计算机图形学中三维几何的主要表示形式。它们是电子游戏、电影、游戏和虚拟现实交互中使用的 3D 资产的主要代表。与其他 3D 形状表示 (如点云或体素) 相比, 网格提供了更连贯的表面表示; 它们更容易控制, 更容易操作, 更紧凑, 并直接适合现代渲染流水管线, 以更少的数据量获得高视觉质量。

将深度学习应用于几何数据的一种方法是将 3D 形状转换为图像, 例如一组无序的投影 [11], 全景图 [9] 或几何图像 [10], 然后在它们上运行 2D CNN。这类间接方法并不是几何不变的, 因为涉及了额外的与视图相关的投影步骤。另一种直接解决方案是表示其固有 3D 空间中的形状, 如体素数据, 这样可以直接应用 3D CNN [6, 14] 或适应更高分辨率 [3, 5, 12]。最近, 基于顶点的深度学习技术也是研究的热点 [4, 7, 13], 这是因为三维传感器很容易获取点云数据。然而, 体素数据的高计算需求和点云数据拓扑信息的缺失使得它们不适用于现有的渲染流水管线。在网格曲面上学习的方法克服了上述问题, 有望取代体素和点云成为几何学习的主要研究对象。当前基于网格的神经网络 [1] 通常假设网格是高质量的, 尤其要求网格是流形表面并且是水密的。然而, 无论是手工设计还是自动生成的网格模型, 都存在着各种各样的缺陷, 如非流形面、裂缝、断片、自相交面等。基于网格的神经网络需要在存在拓扑问题和其他网格质量问题时具有鲁棒性: 即使在不完美的输入下, 特征融合和层次结构生成也应该有效。网格神经网络的主要挑战是为不规则的输入数据生成层次结构。各种工作 [1, 2, 8] 采用消去边的方法来生成层次结构, 但这种朴素的方法不能保证放大感受野的一致性和拓扑结构恒定。此外, 简化可能会失败, 层次与层次之间的映射可能定义不清。因此, 本文提出一种基于 Loop 细分的架构——SubdivNet, 以确保卷积和池化过程感受野的一致性。

3 本文方法

3.1 符号定义

三角网格 $\mathcal{M} = (V, F)$ 由点集 $V = \{v_i | v_i \in \mathbb{R}^3\}$ 和面集 $F = \{f_i | f_i \in \{1, \dots, |V|\}^3\}$ 定义。每个面 f_i 上定义有特征向量 e_i 。

若两个面 f_i, f_j 有公共边, 则称两个面相邻。 f_i 和 f_j 之间的距离 $D(f_i, f_j)$ 定义为通过边从一个面到另一个面的任意路径所遍历的面的最小数量。 f_i 的 k 领域定义为

$$\mathcal{N}_k(f_i) = \{f_j | D(f_i, f_j) = k.\}$$

设将 L 为 \mathcal{M} 的细分深度, \mathcal{M}_0 作为 \mathcal{M} 的基网格。显然, \mathcal{M} 的面数为 $|F| = 4^L |F_0|$ 。

3.2 概述

给定一个具有 Loop 细分序列连通性的水密二维流形三角形网格, 旨在学习 3D 形状的全局表示, 或每个面片的局部几何特征向量。类似二维图像金字塔, $(\mathcal{M}_L \cdots \mathcal{M}_0)$ 提供了一个层次结构。通过细分序列连通性提供的 4-1 面片映射, 我们还可以逐步建立从 \mathcal{M}_L 到 \mathcal{M}_0 的面片对应, 允许从局部到全局的特征聚合。在此基础上, SubdivNet 以小批量具有细分序列连通性的封闭二维流形三角网格作为输入。

3.3 卷积

本节讨论对面片 f_i 进行卷积过程中那些面片应该被考虑，称这样的面片为卷积核对象，记为 $\Omega(f_i)$ 。

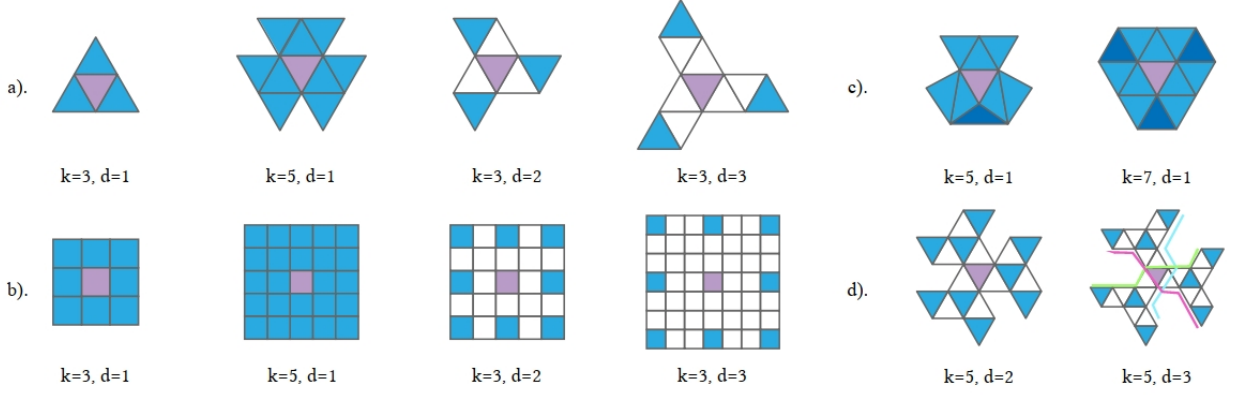


图 1. 网格卷积核对象。(a) 具有不同核大小 k 和空洞 d 的网格卷积核。(b) 对应的 2D 卷积核 (c) 一些核大小大于 3 的卷积操作可能会重复访问: 深蓝色的面被访问两次。(d) 更复杂的卷积核, 具有更大的核尺寸和空洞。

3.3.1 基础卷积

为给定面片定义卷积的关键是指定其邻域。由于二维流形三角形网格中没有边界，因此网格上的每个面恰好有 3 个邻接面。这种 3 相邻属性类似于 2D 图像中像素的连接。形式上，对于每个面片 f_i ，基本卷积核对象由其 1 邻居 $\Omega(f_i) = \mathcal{N}_1(f_i)$ 组成，如图1(a) 所示。

3.3.2 不同核大小的卷积

为了使卷积具有更大的感受野，2D 图像中的卷积被设计为支持可变的内核大小。这在网格处理中也至关重要，以学习每个顶点和面片的更具判别力的表示。我们考虑更多的邻接面来定义核大小为 k 的卷积对象，

$$\Omega(f_i, k) = \bigcup_{i=1}^{\hat{k}} \mathcal{N}_{\hat{k}}(f_i), \hat{k} = \frac{k-1}{2}, k = 1, 3, 5, \dots \quad (1)$$

卷积核对象总共有 $3 \times (2^{\hat{k}} - 1)$ 个面。例如，图1(a) 描述了一个核大小为 5 的卷积核对象。然而，当 k 大于 3 时，相邻的面可能被访问了不止一次，导致 Ω 中的面片数比预期的少，如图1(c) 所示。当 k 大于 7 时，面可能被访问两次以上。

3.3.3 空洞卷积

空洞卷积通过在卷积核中插入空洞 (见图1(b))，可以在不消耗更多计算或内存资源的情况下扩大感受野。为了将这个概念扩展到三角形网格，我们定义空洞步长 d ，若面与其最近的卷积对象之间的距离是 d 。特别地，基础卷积的空洞步长为 1。在二维图像网格中，通过跳过行和列可以很容易地获得带空洞的卷积核。本文提出折线策略来定义空洞卷积，如图2(a) 所

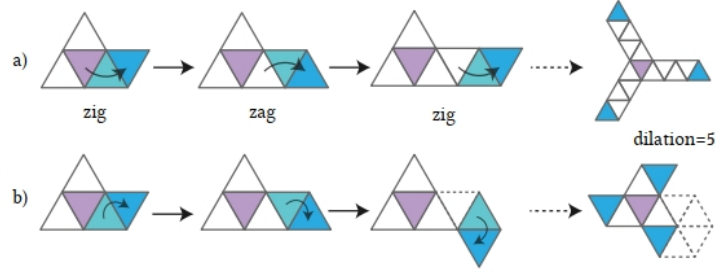


图 2. (a) 采取折线策略从基础卷积生成一个均匀扩张的空洞卷积。(b) 若不采取折线方案，将导致空洞卷积选择的三角形过于靠近中心三角形。

示。从 f_i 出发，按照顺时针或逆时针交替的方向访问 $\Omega(f_i, d)$ 的面，注意对 f_i 也算一次访问，在第 d 次访问的面即空洞卷积对象。

3.3.4 顺序不变的卷积计算

面片的三个邻接面是无序的，但邻接面选取的顺序应该不改变卷积计算的结果，即卷积计算应该是顺序不变的。虽然 $\Omega(f_i)$ 是一个无序的集合，但我们围绕 f_i 逆时针重新排列该集合，从而得到一个序列 $\hat{\Omega}(f_i)$ 。因此， $\hat{\Omega}(f_i)$ 是一个闭合的环。尽管如此，环排序从哪里开始仍然是不明确的，因此计算顺序不变的特性就显得尤为重要。面片 f_i 上的卷积操作定义为

$$\text{Conv}(f_i) = w_0 e_i + w_1 \sum_j e_j + w_2 \sum_j |e_{j+1} - e_j| + w_3 \sum_j |e_i - e_j|. \quad (2)$$

其中 e_j 是面 f_i 的特征向量， (w_0, w_1, w_2, w_3) 是可学的参数。由于面片排序不影响求和结果，因此也不影响卷积操作。

3.4 池化和上采样

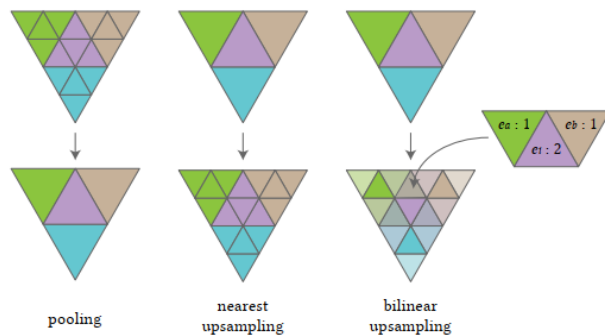


图 3. 池化和上采样

有了输入网络的细分金字塔，三角形网格上的池化和上采样就像在常规 2D 图像网格上一样简单，如图3所示：较细网格中的四个细分面被池化到较粗网格中的父面。上采样只是将面片分割为 4 张面片，子面的特征是从父面复制或者双线性插值得到。

3.5 网络架构

得益于本文提出的卷积操作的普适性，复现可以轻松地将热门的二维卷积网络应用于三维网格。复现采用 DeepLabv3+ 用于分割任务。

3.6 输入特征

每个三角形面片的输入特征是一个 13 维向量，由一个 7 维形状描述符和一个 6 维姿态描述符组成。形状描述符包括面片面积，三角形的三个内角，以及面片法线与三个顶点法线的内积（表征曲率）。姿态描述符给出面片中心的位置和面片法向，帮助网络识别具有相似形状和位置的面片。

4 复现细节

4.1 与已有开源代码对比

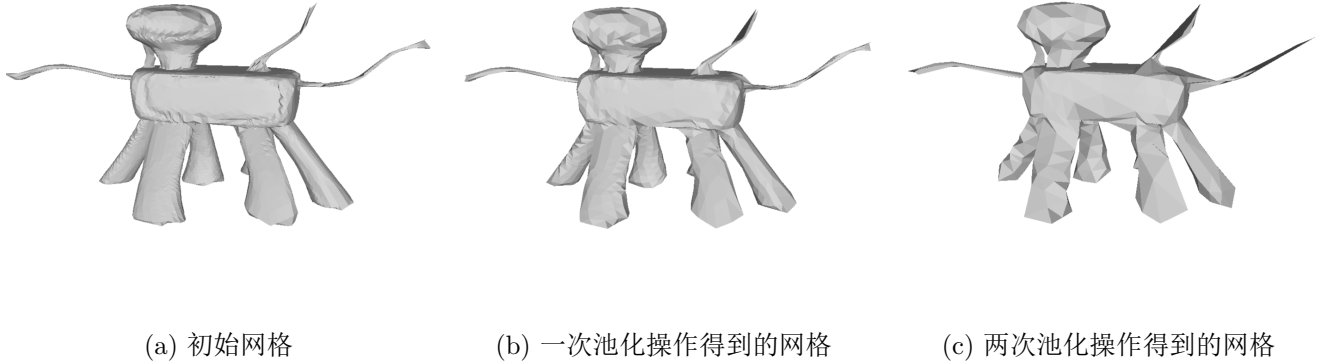


图 4. 网格特征可视化

复现在源代码基础上增加了由面片特征计算顶点位置的功能，成功实现了网络计算过程的可视化。

面片 f_i 的输入特征 e_i 提供了面片中心位置 (x_i, y_i, z_i) 以及面片法向 (n_i^1, n_i^2, n_i^3) ，因此很容易能求出面片所在平面的表达式为

$$n_i^1(x - x_i) + n_i^2(y - y_i) + n_i^3(z - z_i) = 0. \quad (3)$$

要求某个顶点的坐标，有三种可能性：

1. 若该顶点所在的三个不在同一平面上的面片，则联立这三个面片的表达式求解即可得到该顶点位置坐标；
2. 若该顶点的所有邻接面都在同一平面上，则该顶点的位置可以直接取所有邻接面的中心位置的平均；

3. 若该顶点的所有邻接面相较于一条直线，则取该顶点的所有邻接面中心的平均到交线的垂足位置作为该顶点的位置。

可视化结果如图4所示。

4.2 复现结果

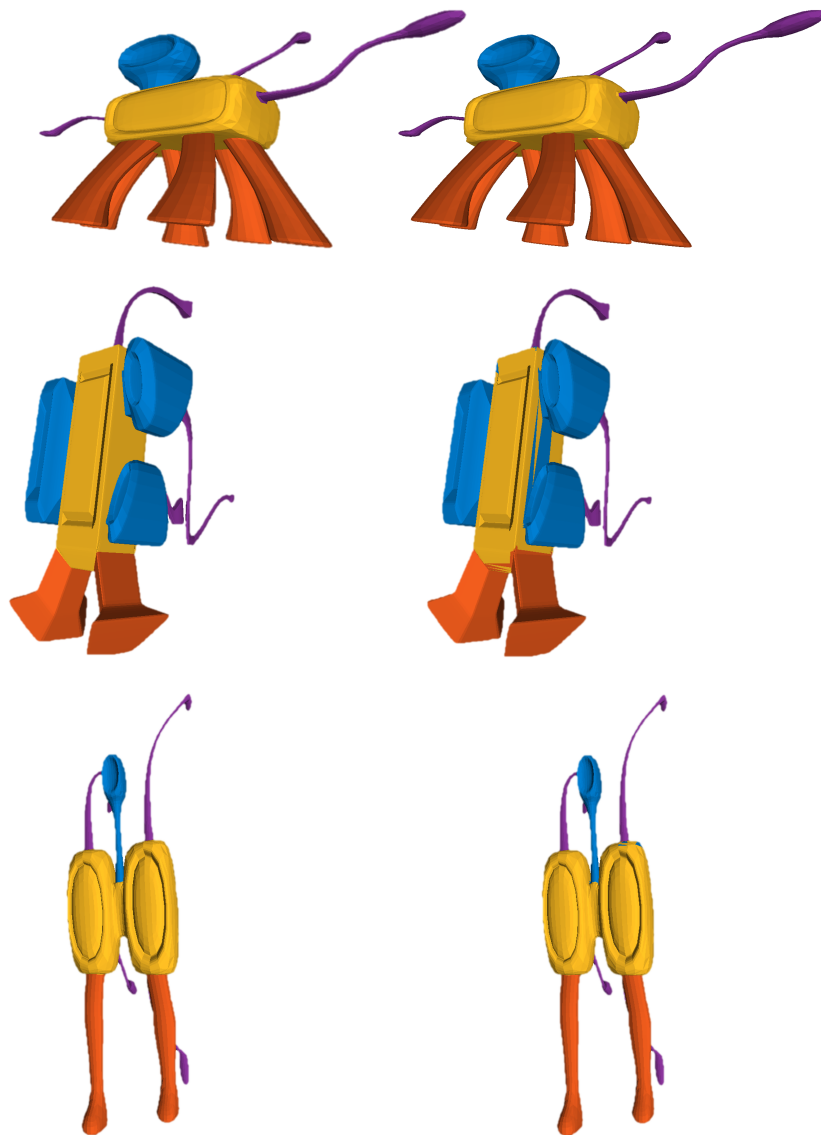


图 5. 复现结果

训练 100 轮后，训练精度为 0.989，测试准确率为 0.974，部分测试结果如图5所示。部分边界区域分割略有误差，大部分区域分割效果不错。

5 总结与展望

未来可以探索如何进行反卷积操作，进而实现三维网格的生成。

参考文献

- [1] Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. Meshcnn: a network with an edge. *ACM Transactions on Graphics*, 38(4):1–12, July 2019.
- [2] Shi-Min Hu, Zheng-Ning Liu, Meng-Hao Guo, Jun-Xiong Cai, Jiahui Huang, Tai-Jiang Mu, and Ralph R. Martin. Subdivision-based mesh convolution networks. *ACM Transactions on Graphics*, 41(3):1–16, March 2022.
- [3] Roman Klokov and Victor Lempitsky. Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. In *Proceedings of the IEEE international conference on computer vision*, pages 863–872, 2017.
- [4] Xianzhi Li, Ruihui Li, Lei Zhu, Chi-Wing Fu, and Pheng-Ann Heng. Dnf-net: A deep normal filtering network for mesh denoising. *IEEE Transactions on Visualization and Computer Graphics*, 27(10):4060–4072, 2020.
- [5] Zheng-Ning Liu, Yan-Pei Cao, Zheng-Fei Kuang, Leif Kobbelt, and Shi-Min Hu. High-quality textured 3d shape reconstruction with cascaded fully convolutional networks. *IEEE Transactions on Visualization and Computer Graphics*, 27(1):83–97, 2019.
- [6] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 922–928. IEEE, 2015.
- [7] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.
- [8] Jonas Schult, Francis Engelmann, Theodora Kontogianni, and Bastian Leibe. Dualconvmesh-net: Joint geodesic and euclidean convolutions on 3d meshes, 2020.
- [9] Baoguang Shi, Song Bai, Zhichao Zhou, and Xiang Bai. Deeppano: Deep panoramic representation for 3-d shape recognition. *IEEE Signal Processing Letters*, 22(12):2339–2343, 2015.
- [10] Ayan Sinha, Jing Bai, and Karthik Ramani. Deep learning 3d shape surfaces using geometry images. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VI 14*, pages 223–240. Springer, 2016.
- [11] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition, 2015.

- [12] Peng-Shuai Wang, Chun-Yu Sun, Yang Liu, and Xin Tong. Adaptive o-cnn: A patch-based deep representation of 3d shapes. *ACM Transactions on Graphics (TOG)*, 37(6):1–11, 2018.
- [13] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5):1–12, 2019.
- [14] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.