/Title (AAAI Press Formatting Instructions for Authors Using LaTeX – A Guide) /Author (AAAI Press Staff, Pater Patel Schneider, Sunil Issar, J. Scott Penberthy, George Ferguson, Hans Guesgen, Francisco Cruz, Marc Pujol-Gonzalez) /TemplateVersion (2022.1)

3D Shape Analysis Using the CNN

Written by 周芃为23020211153995，周密23020211153994，郭思正 31520211154042，曾威 31520211154020

[1]Association for the Advancement of Artificial Intelligence
2275 East Bayshore Road, Suite 160
Palo Alto, California 94303
publications22@aaai.org

Abstract

Convolutional neural networks (CNNs) have made great breakthroughs in 2D computer vision. Consequently, CNN has been also applied to 3D geometry, inspired by 2D CNNs. There are several for 3D data, such as point cloud, voxel, mesh and some implicit representations. Among the representations above, polygonal meshes provide a relatively efficient representation for 3D shapes. Compared with point cloud, they explicitly capture topology of a shape, while compared with voxel, they only represent the boundary of an object, and does not have redundant elements representing the object's interior. Meshes leverage non-uniformity to represent large flat regions as well as intricate features. However, this nonuniformity and irregularity inhibits mesh analysis efforts using CNNs. Based on MeshCNN, which present special edge-based convolution and pooling, we attempt to improve the method of MeshCNN, since it has limited receptive field and can not capture global features. In this project, we still utilize the unique properties of the triangle mesh for a direct analysis of 3D shapes, including classification and segmentation, using an improved edition of MeshCNN, a convolutional neural network designed specifically for triangular meshes

Introduction

The great success of deep convolutional neural networks (CNNs) in 2D computer vision has led to their generalisation to various disciplines, including 3D geometry. For computational reasons, and to facilitate data processing, various discrete approximations for 3D shapes have been suggested and utilized to represent shapes in an array of applications. PointNet [1] is a pioneering and representative approach for learning a feature representation of a point cloud, followed by more successful work in this domain .Apart from point clouds,3D geometry learning has been extended to other forms of 3D data, such as voxels and meshes .

In this project, we consider the neural network MeshCNN, which aim to tap into the natural potential of the native mesh representation. The network adequately utilize the unique property, every edge is incident to exactly two faces (triangles), which defines a natural fixed-sized convolutional neighborhood of four edges. But we find a defect of the special convolutional operation. The convolution kernel of MeshCNN has a receptive field which is limited to the adjacent four edges of a given edge. We try to improve this defect. However, unlike conventional edge collapse, which removes edges that introduce a minimal geometric distortion, mesh pooling delegates the choice of which edges to collapse to the network in a task-specific manner. The purged edges are the ones whose features contribute the least to the used objective(see examples in figures 1 and 2).

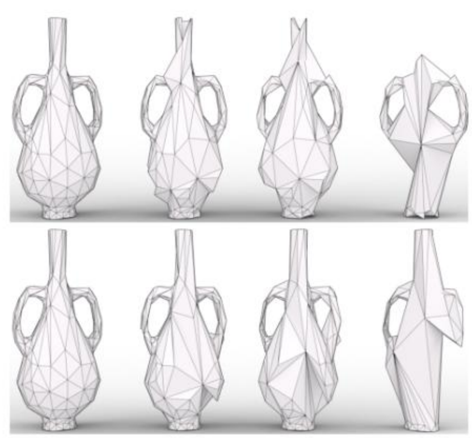

Figure1

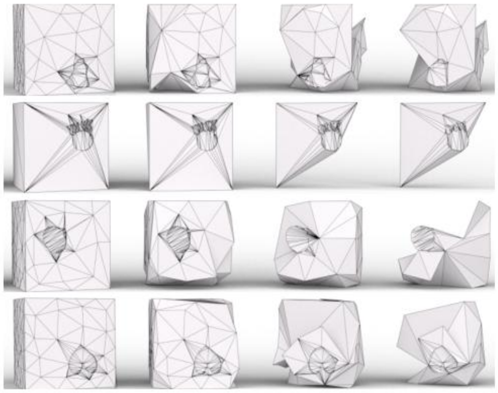

Figure2

Related Work

Many of the operators that MeshCNN presents or uses in their work are based on classic mesh processing techniques [8][9][10], or more specifically, mesh simplification techniques [11][12][13].In particular, they use the edge-collapse technique [13] for the task-driven pooling operator. While classic mesh simplification techniques aim to reduce the number of mesh elements with minimal geometric distortion[14][15], in MeshCNN, they use the mesh simplification technique to reduce the resolution of the feature maps within the context of a neural network. In the following, we briefly review relevant work on 3D geometric learning ,organized according to input representation type.

Multi-view 2D projections. One way of applying deep learning to geometric data is to transform 3D shapes into images. Consequently, the method of transform 3D data into 2D projection has been exploited. Additionally, leveraging existing techniques and architectures from the 2D domain is made possible by representing 3D shapes through their 2D projections from various viewpoints. These sets of rendered images serve as input to subsequent processing by standard CNN models.[16] were the first to apply a multi-view CNN for the task of shape classification, however, this approach cannot perform semantic-segmentation. Then [17] presented a more comprehensive multi-view framework for shape segmentation which fix the defect above.

Volumetric. Transforming a 3D shape into voxels provides a grid-based representation that is analogous to the 2D grid of an image, so that operations that are applied on 2D grids can be extended to 3D grids in a straight-forward manner, thus allowing common image-based approaches apply to 3D. [18] pioneered this concept, and presented a CNN that processes voxelized shapes for classification and completion. Following that, [19] tackled shape reconstruction using a voxel-based variational autoencoder, and[20] combined trilinear interpolation and Conditional Random Fields (CRF) with a volumetric network to promote semantic shape segmentation.[21] used voxel to train a network to regress grid-based warp fields for shape alignment, and applied the estimated deformation on the original mesh.

Point clouds. The point cloud is a classic candidate for data analysis attributed to the close relationship to data acquisition and ease of conversion from other representations. [1] proposes to use 1x1 convolutions followed by global max pooling for order invariance. In its followup work, [3], points are partitioned to capture local structures better.

Meshes. A mesh representation is based on three types of geometric primitive: vertices, edges, and faces. We classify mesh deep learning methods according to which of these is treated as the primary data. The first is the vertex-based approach. One popular approach performs deep learning on 3D shapes by locally encoding in the neighborhood of each vertex into a regular domain, whereupon convolution operations (or kernel functions)can imitate those used for images, such as [22][23][24].The second is the edge-based method. In a 2-manifold mesh, every edge is adjacent to two faces, and the four other edges of those two triangles. This property is exploited by [25] to define an order invariant convolution. PD-MeshNet[26] first constructs a primal graph and a dual graph from the input mesh, then performs convolutions on these graphs using a graph attention network[27]. MeshWalker [29] employs random walks along edges to extract shape features, instead of exploiting regular neighborhood structures. The last one is face-based method. Face-based methods focus on how to efficiently and effectively gather information from neighboring faces.[30] propose a rotationally invariant face based method considering ring neighbors.[31] propose MeshNet. It adopts graph-constrained mesh-cell nodes to integrate local-to-global geometric features. DNF-Net [32] denoises mesh normals on cropped local patches using multi-scale embedding and a residual learning strategy. TextureNet [33] parameterizes mesh patches and high-resolution textures as quadrilaterals to employ grid convolution. SubdivNet[34],which is presented this year, achieves the state-of-art result on tasks of shape classification and segmentation.

Proposed Solution

Applying CNN on Meshes

The most fundamental and commonly used 3D data representation in computer graphics is the non-uniform polygonal mesh; large flat regions use a small number of large polygons, while detailed regions use a larger number of polygons. A mesh explicitly represents the topology of a surface: faithfully describing intricate structures while disambiguating proximity from nearby surfaces (see Figure 3).

Realizing our goal to apply the CNN paradigm directly onto triangular meshes, necessitates an analogous definition and implementation of the standard building blocks of CNN: the convolution and pooling layers. As opposed to images which are represented on a regular grid of discrete values, the key challenge in mesh analysis is its inherent irregularity and non-uniformity.

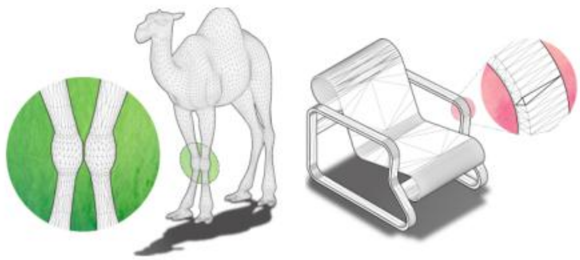

Figure3

Methods

Mesh Convolution. We define a convolution operator for edges, where the spatial support is defined using the four incident neighbors (Figure 3). Recall that convolution is the dot product between a kernel k and a neighborhood, thus the convolution for an edge feature e and its four adjacent edges is:

$$e \cdot k_0 + \sum_{j=1}^{4} k_j \cdot e^j,$$

Mesh Pooling. We extend conventional pooling to irregular data, by identifying three core operations that together generalize the notion of pooling:
1) define pooling region given adjacency
2) merge features in each pooling region
3) redefine adjacency for the merged features

Mesh Unpooling. Each mesh unpooling layer is paired with a mesh pooling layer, to upsample the mesh topology and the edge features. The unpooling layer reinstates the upsampled topology (prior to mesh pooling),by storing the connectivity prior to pooling. Note that upsampling the connectivity is a reversible operation (just as in images). For unpooled edge feature computation, we retain a graph which stores the adjacencies from the original edges (prior to pooling) to the new edges (after pooling). Each unpooled edge feature is then a weighted combination of the pooled edge features. The case of average unpooling is demonstrated in Figure 4.

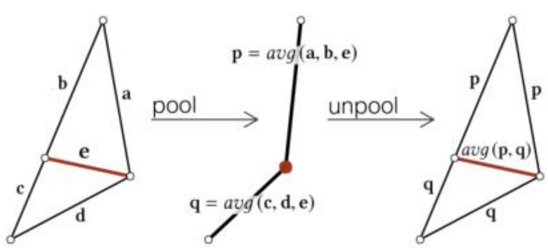

Figure4

Plan

Our project is based on MeshCNN. Considering the limitation of the convolution in MeshCNN which is mentioned above, we attempt to modify the original convolution pattern or the pipeline of the network to obtain better experiment results on the tasks of 3D shape classification and segmentation.

Experiments

Data Processing

Geometric mesh decimation helps to reduce the input resolution and with it the network capacity required for training , so we simplidied each mesh to roughly the same number of edges.Since our task is shape classification,the requirement of the shape resolution is relatively low(about 750 edges).

Augmentation. Since our input features are similarity-invariant, applying rotation, translation and isotropic scaling does not generate new input features.However , applying anisotropic scaling on the vertex locations in x , y and z can generate new features.Moreover, we shift vertices of each mesh to different locations and augment the tessellation of each object by performing random edge flips.

Mesh Classification

SHREC. We performed classification on 30 classes from the SHREC dataset , with 20 examples per class. Split 16 and 10 of the SHREC dataset are the number of training examples per class and we use the split 16 in our task.We stop training after 200 epochs.Figure 5 shows the test results.

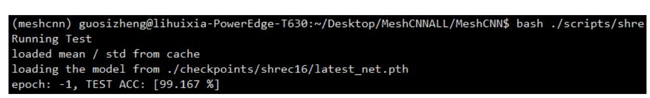

Figure5

We also visualize some examples of mesh pooling simplifications of this dataset in Figure 6.

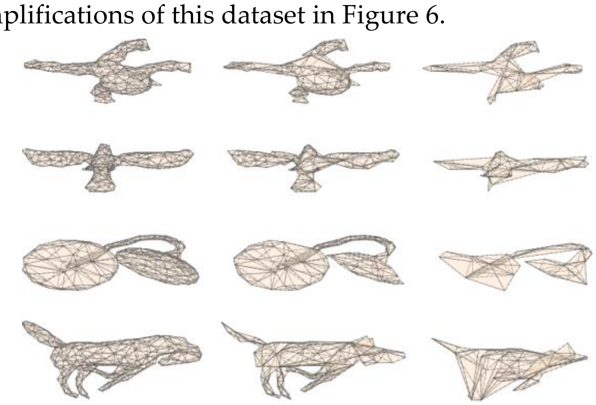

Figure6

Cube engraving. The dataset of cubes is modeled with shallow icon engravings(see Figure 7)

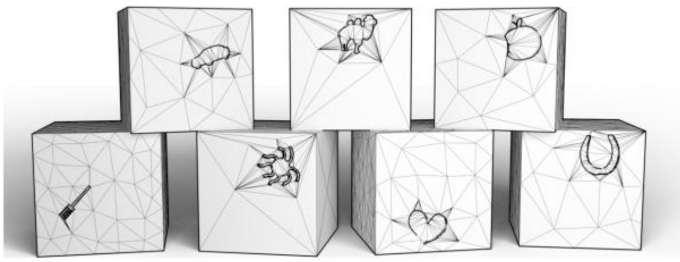

Figure7

We train our network to classify the cubes.We show the test accuracy in Figure 8 and visualize the effect of mesh pooling in Figure 9.

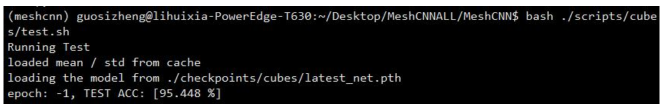

Figure8

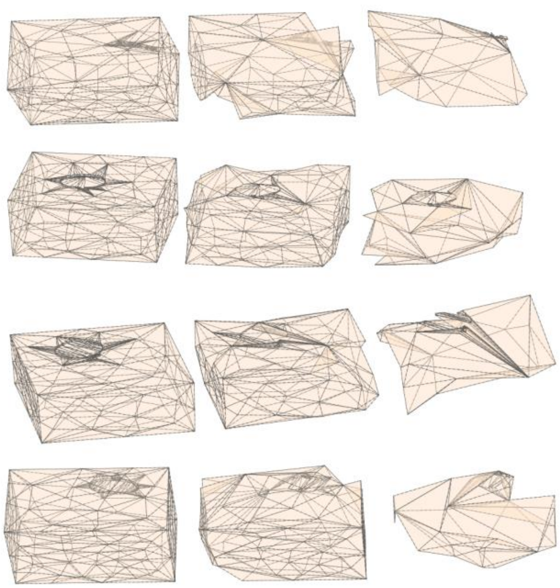

Figure9

Conclusion

DISCUSSION AND FUTURE WORK

We have presented MeshCNN, a general method for employing neural networks directly on irregular triangular meshes. The key contribution of our work is the definition and application of convolution and pooling operations tailored to irregular and non-uniform structures. These operations facilitate a direct analysis of shapes represented as meshes in their native form, and hence benefit from the unique properties associated with the representation of surface manifolds with non-uniform structures.

References

[1]Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. Proc. Computer Vision and Pattern Recognition (CVPR), IEEE 1, 2 (2017), 4.

[2]Yangyan Li, Rui Bu, Mingchao Sun, and Baoquan Chen.2018. PointCNN. CoRRabs/1801.07791 (2018).

[3]Charles R. Qi, Li Yi, Hao Su, and Leonidas J Guibas. 2017b. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In Advances in Neural Information Processing Systems (NIPS). 5105âĂŞ5114.

[4]Roman Klokov and Victor S. Lempitsky. 2017. Escape from Cells: Deep Kd-Networks for the Recognition of 3D Point Cloud Models. In ICCV. IEEE Computer Society,863–872.

[5]Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun,and Xin Tong. 2017. O-CNN:octree-based convolutional neural networks for 3D shape analysis. ACM Trans.Graph.36, 4 (2017), 72:1–72:11.

[6]Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. 2019. MeshCNN: a network with an edge. ACM Trans. Graph. 38, 4 (2019),90:1–90:12.

[7]Alon Lahav and Ayellet Tal. 2020. MeshWalker: deep mesh understanding by random walks. ACM Trans. Graph.39, 6 (2020), 263:1–263:13.

[8]Hugues Hoppe. 1999. New quadric metric for simplifying meshes with appearance attributes. In Visualization’99.Proceedings. IEEE, 59–510.

[9]Szymon Rusinkiewicz and Marc Levoy. 2000. QSplat: A Multiresolution Point Rendering

[10]System for Large Meshes. In Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH ’ 00). ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 343–352.https://doi.org/10.1145/344779.344940

[11]Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle.1993. Mesh optimization. , 19–26 pages.

[12]Michael Garland and Paul S Heckbert. 1997. Surfacesimplification using quadric error metrics. In Proceedings of the 24th annual conference onComputer graphics and interactive techniques. ACM Press/Addison-Wesley Publishing Co., 209–216.

[13]Hugues Hoppe. 1997. View-dependent refinement of progressive meshes. In Proceedings of the 24th annual conference on Computer graphics and interactive techniques.ACM Press/Addison-Wesley Publishing Co., 189-198.

[14]Marco Tarini, Nico Pietroni, Paolo Cignoni, Daniele Panozzo, and Enrico Puppo. 2010.Practical quad mesh simplification. In Computer Graphics Forum, Vol. 29. Wiley Online Library, 407 – 418.
[15]Xifeng Gao, Daniele Panozzo, Wenping Wang, Zhigang Deng, and Guoning Chen. 2017.Robust structure simplification for hex re-meshing. ACM Transactions on Graphics 36, 6 (2017).
[16]Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Millers. 2015. Multiview Convolutional Neural Networks for 3D Shape Recognition. In InternationalConference on Computer Vision (ICCV).
[17]Evangelos Kalogerakis, Melinos Averkiou, Subhransu Maji, and Siddhartha Chaudhuri.2017. 3D shape segmentation with projective convolutional networks. In Proc. CVPR,Vol. 1. 8.
[18]Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Li-guang Zhang, Xiaoou Tang,and Jianxiong Xiao. 2015. 3D shapenets: A deep representation for volumetric shapes. In Computer Vision and Pattern Recognition (CVPR).1912âĂŞ1920.
[19]Andrew Brock, Theodore Lim, J.M. Ritchie, and Nick Weston. 2016. Generative and Discriminative Voxel Modeling with Convolutional Neural Networks. In NIPS 3D Deep Learning Workshop.
[20]Lyne P. Tchapmi, Christopher B. Choy, Iro Armeni, JunYoung Gwak, and Silvio Savarese.2017. SEGCloud: Semantic Segmentation of 3D Point Clouds. In 3DV.
[21]Rana Hanocka, Noa Fish, Zhenhua Wang, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. 2018. ALIGNet: Partial-Shape Agnostic Alignment via Unsupervised Learning. ACM Trans. Graph. 38, 1, Article 1 (Dec. 2018), 14 pages. https://doi.org/10.1145/3267347
[22]Jonathan Masci, Davide Boscaini, Michael M. Bronstein,and Pierre Vandergheynst.2015. Geodesic Convolutional Neural Networks on Riemannian Manifolds. In ICCV Workshops. IEEE Computer Society, 832 – 840.
[23]Davide Boscaini, Jonathan Masci, Emanuele Rodolà, and Michael M. Bronstein. 2016.Learning shape correspondence with anisotropic convolutional neural networks. InNIPS. 3189 – 3197.
[24]Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodolà, Jan Svoboda, and Michael M. Bronstein. 2017.Geometric Deep Learning on Graphs and Manifolds Using Mixture Model CNNs. In CVPR. IEEE Computer Society, 5425 – 5434.
[25]Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Sh-char Fleishman, and Daniel Cohen-Or. 2019. MeshCNN: a network with an edge. ACM Trans. Graph. 38, 4 (2019),90:1 – 90:12.
[26]Francesco Milano, Antonio Loquercio, Antoni Rosinol,Davide Scaramuzza, and Luca Carlone. 2020. Primal-Dual Mesh Convolutional Neural Networks. In NeurIPS.
[27]Federico Monti, Oleksandr Shchur, Aleksandar Bojchevski, Or Litany, Stephan Günnemann, and Michael M. Bronstein. 2018. Dual-Primal Graph Convolutional Networks.CoRR abs/1806.00770 (2018). arXiv:1806.00770
[28]Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In ICLR (Poster). OpenReview.net.
[29]Alon Lahav and Ayellet Tal. 2020. MeshWalker: deep mesh understanding by random walks. ACM Trans. Graph. 39, 6 (2020), 263:1 – 263:13.
[30]Haotian Xu, Ming Dong, and Zichun Zhong. 2017. Directionally Convolutional Networks for 3D Shape Segmentation. In ICCV. IEEE Computer Society, 2717 – 2726.
[31]Chunfeng Lian, Li Wang, Tai-Hsien Wu, Mingxia Liu, Francisca Durán, Ching-Chang Ko, and Dinggang Shen. 2019. MeshSNet: Deep Multi-scale Mesh Feature Learning for End-to-End Tooth Labeling on 3D Dental Surfaces. In MICCAI (6) (Lecture Notes in Computer Science), Vol. 11769. Springer, 837 – 845.
[32]Xianzhi Li, Ruihui Li, Lei Zhu, Chi-Wing Fu, and Pheng-Ann Heng. 2020. DNF-Net: a Deep Normal Filtering Network for Mesh Denoising. CoRR abs/2006.15510 (2020).
[33]Jingwei Huang, Haotian Zhang, Li Yi, Thomas A. Funkhouser, Matthias Nießner, and Leonidas J. Guibas. 2019. TextureNet: Consistent Local Parametrizations for Learning From High-Resolution Signals on Meshes. In CVPR. Computer Vision Foundation /IEEE, 4440 – 4449.
[34]Shi-Min Hu, Zheng-Ning Liu, Meng-Hao Guo, Jun-Xiong Cai, Jiahui Huang, Tai-Jiang Mu, Ralph R. Martin.2021. Subdivision-Based Mesh Convolution Networks. https://arxiv.org/abs/2106.02285