

硕 士 研 究 生 读 书 报 告



题目 网格卷积神经网络：一种带边的网络

作者姓名 曾志通

作者学号 21951423

指导教师 李启雷

学科专业 软件工程

所在学院 软件学院

提交日期 二○一九年十二月

MeshCNN: A Network with an Edge

A Dissertation Submitted to

Zhejiang University

in partial fulfillment of the requirements for

the degree of

Master of Engineering

Major Subject: Software Engineering

Advisor: Qilei Li

By

Zhitong Zeng

Zhejiang University, P.R. China

2019

摘要

多边形网格为三维形体提供了一种有效的表达方法。它们明确地捕获形体表面和拓扑关系，并利用非均匀性来表示大的平面区域以及尖锐的、复杂的特征。然而，这种非均匀性和不规则性阻碍了使用结合了卷积和池化操作的神经网络进行网格分析的工作。在本文中，我们利用网格的独特性质，使用网格卷积神经网络（MeshCNN）来对三维形体进行直接的分析，这是一个专门为三角网格设计的卷积神经网络。与经典的卷积神经网络（CNN）类似，MeshCNN利用其固有的测地线连接，将用来操作网格的边的特有的卷积层和池化层结合起来。卷积被用在边及该边关联的两个三角形的四条边上，池化则通过一种保留表面拓扑关系的边坍缩操作来实现，从而为后续的卷积操作产生新的网格连接。MeshCNN学习哪些边需要被坍缩，从而形成一个任务驱动的过程，在该过程中，整个网络在舍弃冗余特征的同时暴露和扩展重要的特征。我们展示了这一任务驱动的池化过程在各种应用于三维网格的学习任务中的有效性。

**关键词**： 网格分析，形体分析，形体建模，神经网络

Abstract

Polygonal meshes provide an efficient representation for 3D shapes. They explicitly capture both shape surface and topology, and leverage non-uniformity to represent large flat regions as well as sharp, intricate features. This non-uniformity and irregularity, however, inhibits mesh analysis efforts using neural networks that combine convolution and pooling operations. In this paper, we utilize the unique properties of the mesh for a direct analysis of 3D shapes using MeshCNN, a convolutional neural network designed specifically for triangular meshes. Analogous to classic CNNs, MeshCNN combines specialized convolution and pooling layers that operate on the mesh edges, by leveraging their intrinsic geodesic connections. Convolutions are applied on edges and the four edges of their incident triangles, and pooling is applied via an edge collapse operation that retains surface topology, thereby, generating new mesh connectivity for the subsequent convolutions. MeshCNN learns which edges to collapse, thus forming a task-driven process where the network exposes and expands the important features while discarding the redundant ones. We demonstrate the effectiveness of our task-driven pooling on various learning tasks applied to 3D meshes.

**Keywords:** Mesh Analysis, shape analysis, shape modeling, neural networks

1引言

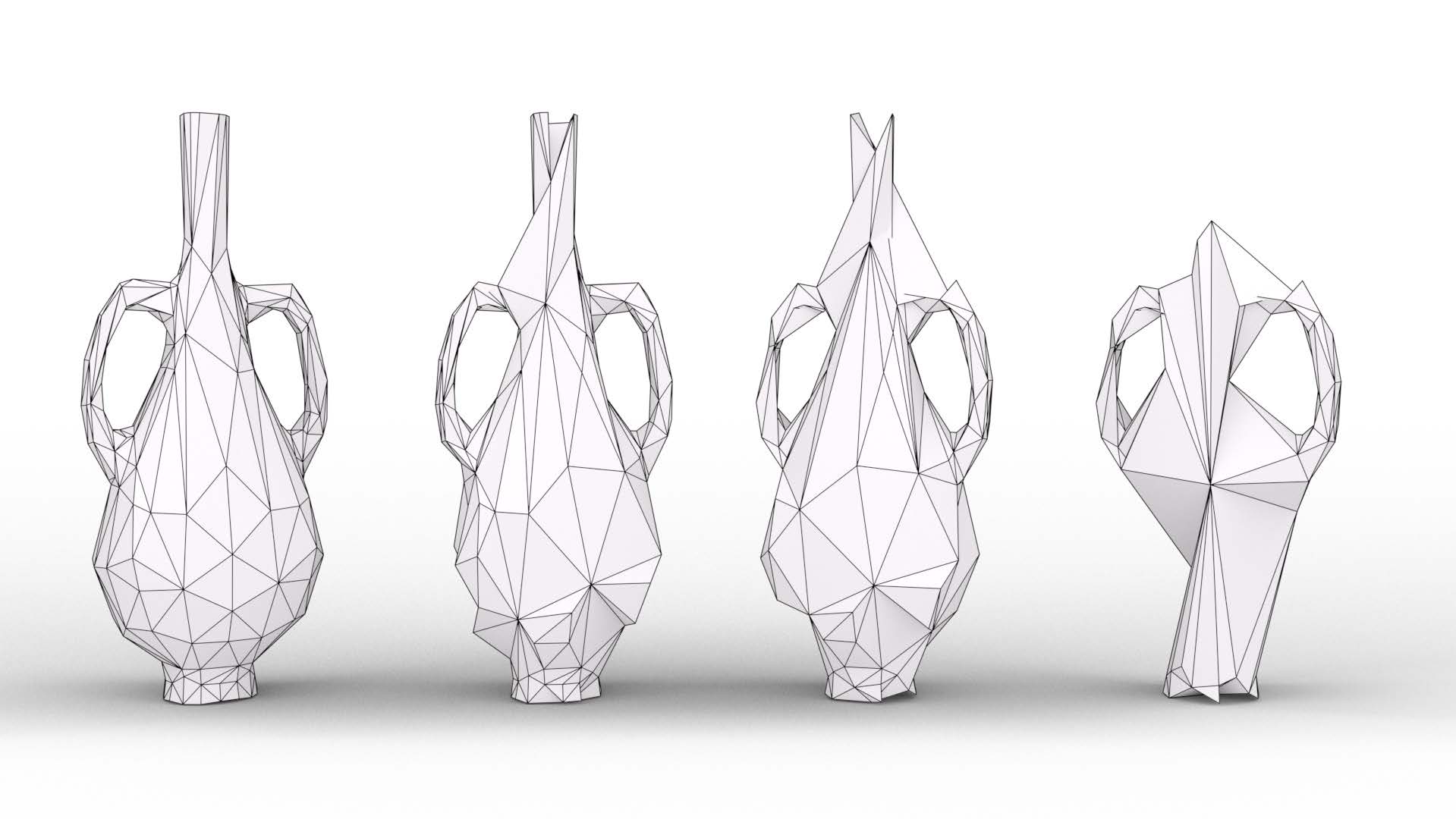
三维形体是计算机图形学领域的前沿和中心，也是计算机视觉和计算几何等相关领域的主要议题。我们周围的形体，尤其是那些描述自然实体的形体，通常是由连续的表面组成的。

为了便于计算和数据处理，各种三维形体的离散近似法被提出，并用于在一系列应用中表示形体。多边形网格表示法（polygonal mesh representation），简称网格（mesh），是许多人的最爱，它通过三维空间中的一组二维多边形来近似表面[Botsch et al. 2010]。网格为形体提供了一种有效的、非均匀的表示法。一方面，只需要少量的多边形就可以捕获大而简单的平面。另一方面，该表示法的灵活性能在需要的地方支持更高的分辨率，允许对凸起的形体特征进行忠实的重建和描绘，这些特征通常在几何上错综复杂。网格的另一个有利特性是其先天地提供了连接信息，这形成了对潜在表面的一个综合表达。

与另一个流行的选择：点云表示法相比，这些优势是显而易见的。尽管点云表示法简单且与普通的数据获取技术（扫描）直接相关，但是当对更高的质量和尖锐形体特征的保真性有需求时，它就显得乏力。

近年来，在图像上使用卷积神经网络已经在诸如分类和语义分割等各种任务上展现出了卓越的效果[2013; 2014; 2018]。它们成功的秘诀在于将卷积层、非线性层和池化层结合起来，从而得到一个对于无关的输入变量保持不变性(或鲁棒性)的框架[LeCun 2012; Krizhevsky et al. 2012]。然而，由于图像是表示在一个离散数值的规则栅格上，将卷积神经网络拓展到不规则的结构上并不容易。

最初的方法使用规则的表示法来避免将卷积神经网络应用到不规则数据上：将三维形体映射到多个二维投影上[Su et al. 2015]或者使用三维体素栅格[Wu et al. 2015]。虽然这些方法获益于直接使用易于理解的图像卷积神经网络操作，但是其间接表示需要大量的内存和冗余的卷积神经网络计算（例如，在不会被使用到的体素上所做的卷积操作）。



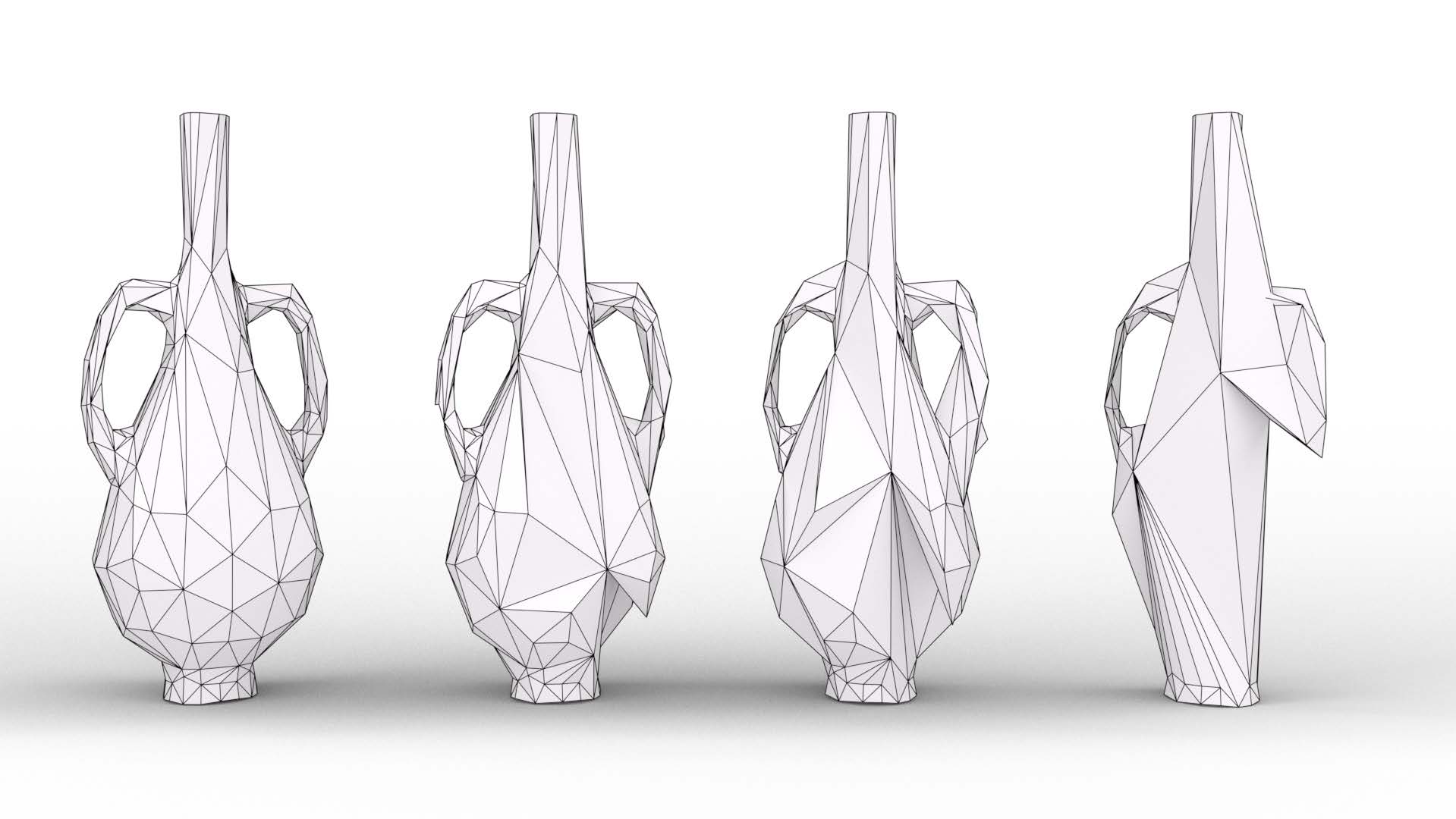


图1. 网格池化作用在不规则的结构上并在空间上适应于该任务。不像几何简化（去除引起最小失真的边），网格池化指派网络中需要坍缩的边。上面一行：训练MeshCNN来分类一个花瓶是否有柄；下面一行：训练MeshCNN以分类花瓶是否有瓶颈（顶部）。

更加高效的方法是直接将卷积神经网络应用到不规则的、稀疏的点云表示法上[Qi et al. 2017a]。尽管这些方法获益于其紧凑的输入表示，但是它们从本质上忽略了局部表面。不仅如此，其邻域和连接性的概念难以定义，使得卷积和池化操作的应用不容易。这种含糊性导致了一波旨在克服这一挑战的工作[Monti et al. 2017; Wang et al. 2018a; Li et al. 2018; Yi et al. 2017]。

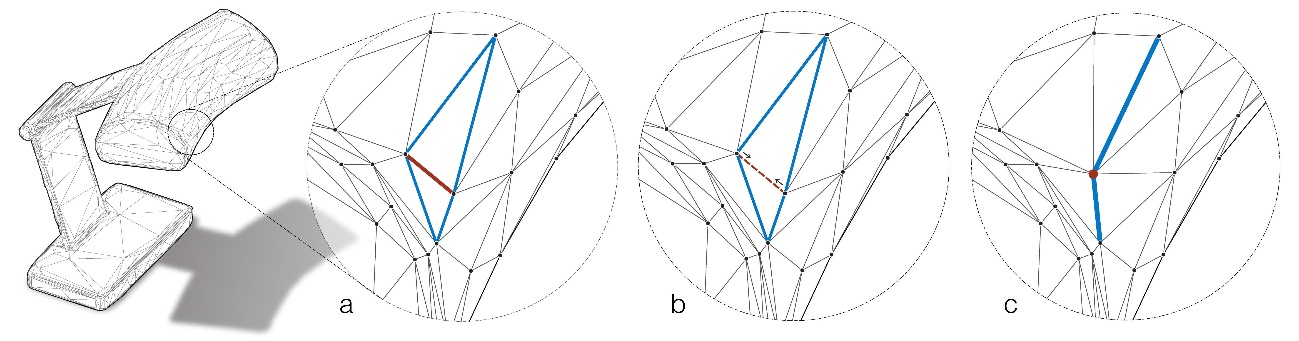


图2. (a) 特征是通过对由一条边的两个相邻三角形的四条边组成的邻域应用卷积而在边上被计算。图中四条蓝色的边与红色的边相邻。池化操作见(b)和(c)。在(b)中，红色的边被缩成一个点，其四条（蓝色的）邻边合并成(c)中的两条（蓝色的）边。注意到在一次单一的边坍缩步骤中，五条边被转换成两条。

为了挖掘原生的网格表示的先天潜力，我们提出了网格卷积神经网络（MeshCNN）：一个类似著名的CNN的神经网络，但是专为网格而设计。MeshCNN直接在不规则的三角网格上进行操作，执行为适应网格特性而设计的卷积和池化操作。在MeshCNN中，一个网格的边就像一张图像中的像素，因为它们是基础构建模块，所有操作都应用在它们之上。因为每条边都伴随着两个面（三角形），我们选择在边上进行操作，定义了自然的固定大小的四边卷积邻域（见图2）。我们利用一致的面法向顺序来应用一种对称的卷积运算，其学习到的边的特征对于旋转、缩放、平移变换都具有不变性。

MeshCNN的一个关键特征是其独有的池化操作，网格池化（mesh pooling），其在不规则的结构上进行操作，并在空间上与任务相适应。在卷积神经网络中，池化对网络中的特征数目进行下采样，从而学习去除信息量较少的特征。因为特征在边上，一种直觉上的下采样方法是使用著名的网格简化技术——边坍缩[Hoppe 1997]。卷积的边坍缩去除那些导致最小几何失真的边，但网格池化却与之不同，它以面向任务的方式指派网络中的哪条边需要被坍缩。其删除的边是那些特征对所用目标贡献最小的边（见图1和8中的示例）。

为了增加灵活性和支持各种可行的数据，每个池化层将网格简化成预先定义的常量数目的边。此外，虽然MeshCNN产生任意指定数目的输出边，但它并不知道输入网格的大小，而且有能力处理不同的三角剖分。如图1所示，我们展示了中间阶段的可计算的池化步骤，其最终输出同样具有语义上的可解释性。为了说明我们这一方法的能力，我们针对通常的数据集和高度非均匀的网格在形体分类和分割任务上进行了各种实验，在与最先进的方法的比对中展现出了优越的结果。

2 相关工作

在我们的工作中提出或使用许多操作都是基于经典的网格处理技术[Hoppe 1999; Rusinkiewicz and Levoy 2000; Botsch et al. 2010; Kalogerakis et al. 2010]，或者更加明确地说是网格简化技术[Hoppe et al. 1993; Garland and Heckbert 1997; Hoppe 1997]。特别地，我们为我们任务驱动的池化技术使用了边坍缩（edge-collapse）技术[Hoppe 1997]。虽然经典的网格简化技术旨在在引起最小几何失真的情况下减少网格元素的数目[Tarini et al. 2010; Gao et al. 2017]，但在本项工作中我们使用该技术是为了在神经网络的上下文中减少特征映射的分辨率。接下来，我们将根据输入表示方式的类型为组织结构，重温使用神经网络进行三维数据分析的工作。

**2.1 多视点二维投影**

通过多个视点的二维投影来表示三维形体使得利用来自二维域的现存技术和架构称为可能。通过标准卷积神经网络模型，这些被渲染的图像被当做后处理的输入。[Su et al. 2015]首次将多视点卷积神经网络应用于形体分类任务，然而这些方法（未处理的）无法实现语义分割。后来，[Kalogerakis et al. 2017]提出一个更加综合的用于形体分割的多视点框架：为每个视点生成图像水平的分割图，然后使用端到端训练的条件随机场（CRF）来求解标签的一致性。[Qi et al. 2016]同时探索了基于视点和体素的方法，并观察到前者相比于当时一些可行方法所具备的优越性。

**2.2 体素**

将三维形体转变成二元体素形式提供了一种基于栅格的表示，类似于一张图像的二维栅格。同样地，应用在二维栅格上的操作可以很直接地被扩展到三维栅格中，从而使得普通的基于图像的方法可以被很自然地迁移到形体域。[Wu et al. 2015]是这一概念的先驱，其提出一种为分类和修复任务处理体素化形体的卷积神经网络。接着，[Brock et al. 2016]使用一种基于体素的变分自编码器（VAE）处理形体重建问题，与此同时，[Tchapmi et al. 2017]使用一种体积化的神经网络将三线性插值（trilinear interpolation）与条件随机场（CRF）结合起来以提高语义形体分割能力。[Hanocka et al. 2018]使用体积化的形体表示法去训练一个网络，从而为形体配准问题回归出基于栅格的变形域，并将估计出来的变形作用于原来的网格。

尽管体积表示法具有诱人的简洁性，但是其在计算上的要求很高，需要消耗大量的内存。为了缓解这一问题，已经有几种加速策略被提出，他们利用体积块中形体占比的稀疏性来进行表示法的简化[Li et al. 2016; Riegler et al. 2017; Wang et al. 2017; Graham et al. 2017]。

**2.3 图**

允许非规则性的基于栅格的表示法的一种常见泛化是图结构。为了支持基于图的数据分析，大量关注点都集中在神经网络在流行任务中的应用上，这些任务都涉及到以图的形式表示数据，主要包括社交网络、通信中的传感器网络和遗传学数据。一种方法主张处理图表示法的拉普拉斯变换[Bruna et al. 2014; Henaff et al. 2015; Defferrard et al. 2016; Kostrikov et al. 2018]，因此操作都在谱域（spectral domain）进行。另一种方法通过提取局部连接区域并将其转换为一种供神经网络处理的规范形式，从而直接对图进行处理[Niepert et al. 2016]。[Atwood et al. 2016]提出了扩散卷积（diffusion-convolution），该扩散操作被应用在每个结点上以决定其局部邻域。[Monti et al. 2017]利用图的空间域（spatial domain）将表面参数化为局部的块面（patches）。[Xu et al. 2017]在表面小块（surface patches）上使用定向卷积（directional convolutions）以完成语义分割的任务。[Yi et al. 2017]在三维分割任务上使用谱域中的图卷积（graph convolutions）。[Kostrikov et al. 2018]使用一个拉普拉斯表面网络去构造一个三维形体的生成模型。[Such et al. 2017]在图中引入了顶点滤波的概念，但是没有包含特征聚合的池化操作。这些方法通常是在图的顶点上进行操作。

**2.4 流形**

[Masci et al. 2015]的开创性工作在网格上引入局部特征的深度学习（类似[Kokkinos et al. 2012]中的固有的网格描述子），并且展示了如何使用学到的特征进行识别和检索。特别地，他们展示了如何使卷积运算成为网格的固有部分。

通常情况下，一个流形形体的局部面片是近似于欧式的（Euclidean）。这种特性可被用来使用标准卷积神经网络进行流形形体分析，通过将三维流形参数化到二维[Henaff et al. 2015; Boscaini et al. 2016; Sinha et al. 2016; Maron et al. 2017; Ezuz et al. 2017]。[Boscaini et al. 2015]使用顶点频率分析（vertex-frequency analysis）去学习一个局部本征的三维形体描述子。另一个方法使用复曲面（toric topology）的拓扑去定义形体图上的卷积[Haim et al. 2018; Maron et al. 2017]。[Poulenard et al. 2018]定义了一种允许在网络的各层间传播测地信息的新的卷积层。

[Verma et al. 2018]提出了一种图神经网络（graph neural network），该网络在进行卷积操作时，其每个结点的邻域不是预先定义好的，而是根据特征动态地计算出来的。[Tatarchenko et al. 2018]提出了正切卷积（tangent convolution），其中每个点的一个小邻域被用来重建用于施加卷积的局部特征。与先前的工作不同，他们通过在一个规则的三维栅格上进行下采样来包含池化操作。一些生成式模型也已经被提出。[Litany et al. 2018]引进了一种可以用来做形体补全的自编码器。[Ranjan et al. 2018]展示了如何用网格自编码器生成三维人脸。

[Bronstein et al. 2017]给出了一份关于几何深度学习的综述。与前面的方法相比，我们的方法的独特性在于我们的网络操作是专门为适应网格结构而设计的。特别地，我们学习了一个独特的基于目标任务的池化操作去选择哪些区域需要被简化。

据我们所知，这是首个提出(i)一个在网格的边上的卷积操作和(ii)一个面向任务的学习型的网格池化操作的工作。在[Ranjan et al. 2018]中，一个用于网格自编码器的固定的池化操作已经被提出。而可被习得的池化操作已经在图卷积神经网络的上下文中被提出[Ying et al. 2018; Cangea et al. 2018]。然而这些操作并没有考虑到独特的三角网格特性。

一种提取边的特征的卷积已经在对偶图卷积模型[Monti et al. 2018]中被提出，它是在图注意力网络（graph attention networks）[Velickovic et al. 2018]上进行扩展。然而，他们工作中的关注点和优化机制与我们的截然不同；在这项工作中我们利用网格独特的结构和属性定义了它的操作，这使得我们可以定义一种对称性的卷积以获得不变的网络特征。

**2.5 点云**

点云可以说是所有表示法中最简单的一种了，它为潜在的三维形体提供了一种不加修饰的近似。与数据获取的密切关系及从其他表示法转换过来的简便性，使得点云成为了数据分析中的一种经典选项。因此，近来的工作多关注于探索使用神经网络进行点云形体分析的技术。点云网络（PointNet）[Qi et al. 2017a]提出使用跟随着全局最大池化的1x1卷积来实现次序不变性。在随后的工作中PointNet++[Qi et al. 2017b]中，其对点进行了划分以更好地捕捉局部结构。[Wang et al. 2018b]考虑了局部的点邻域信息，进行由基于特征空间中距离的点间相似性计算所驱动的动态更新。而大多数基于点云的方法都关注于全局或中层的特性，[Guerrero et al. 2018]提出一种用于估计局部形体属性（例如，来自原始点云的法向和曲率）的网络，[Williams et al. 2018]则从点云中学习表面重建的几何先验。[Atzmon et al 2018]通过将点云函数映射到体素函数，在点云上定义了一种高效的卷积操作。这使得该方法具有关于点集顺序的不变性，且对数据中的一些变形具有鲁棒性。最近，[Li et al. 2018]提出了PointCNN，将卷积的概念从一个局部栅格拓展到欧几里得空间邻域上的点集的-卷积（-convolution）。

与先前的工作不同，在本工作中，我们靠网格的边提供非均匀的、测地的邻域信息，并与卷积邻域达到数目上的一致。以一种符合形体几何与拓扑的方式，利用诸如边坍缩的网格抽取（mesh decimation）技术，具有不变性的特征计算在边上被执行。

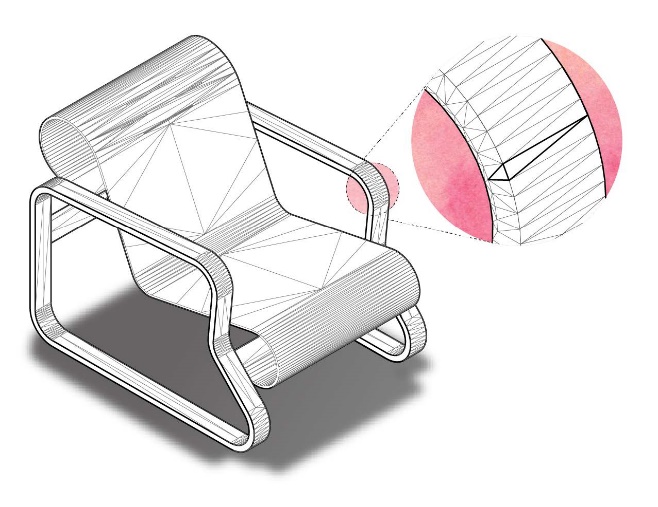
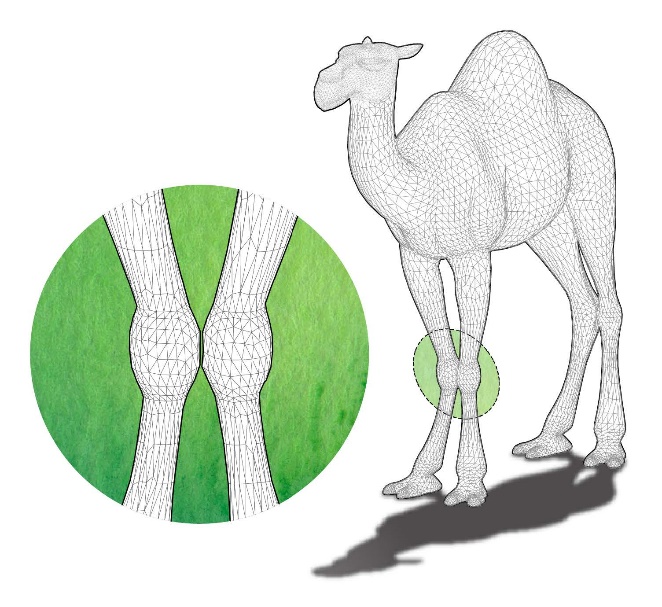


图3. 多边形网格的优点。左图：对形体结构的精确描绘。与点云不同，通过测地上的分离性，该网格能很容易地区分出骆驼关节点（放大）上的不同实体，尽管他们在欧几里得空间上十分接近。右图：适应性的非均匀表示。大的平坦区域能用少量大多边形来表示，而细节区域则用大量小多边形来表示。

3 概述：在网格上使用卷积神经网络

非均匀的多边形网格是计算机图形学中最基础、最常用的三维数据表示方法；大的平坦区域使用小规模的大多边形，而精细的区域则使用大规模的多边形。一个网格显式地表示了一个表面的拓扑：忠实地描述复杂的结构，同时消除邻近表面的歧义性（见图3）。

为了实现直接将CNN范式应用到三角网格上的目标，我们需要有类似CNN标准构建模块的定义与实现：卷积和池化层。图像是由离散数值的一个规则栅格所表示，网格分析正好相反，它的关键挑战在于固有的不规则性和非均匀性。在这项工作中，我们的目标是利用这些具有挑战性的、独特的属性，而不是避开它们。因此，我们在设计网络时故意将卷积和池化操作直接应用到网格的构造上，而非转化为规则的、均匀的表示法。

**3.1 具有不变性的卷积**

在我们的设定中，所有形体都被假设有可能带有边界边的流形网格。这一假设保证了每条边最多关联了两个（三角）面片，因此其与二或四条其他边相邻接。一个面的顶点按逆时针方向排序，则每条边的四条邻边被定义了两种可能的顺序。以图4为例，*e*的一圈邻域（1-ring neighbors）可被排列成(*a*, *b*, *c*, *d*)或者(*c*, *d*, *a*, *b*)，这取决于哪一个面定义了首个邻边。这使得卷积的感受野变得含糊不清，阻碍了不变特征的构造。

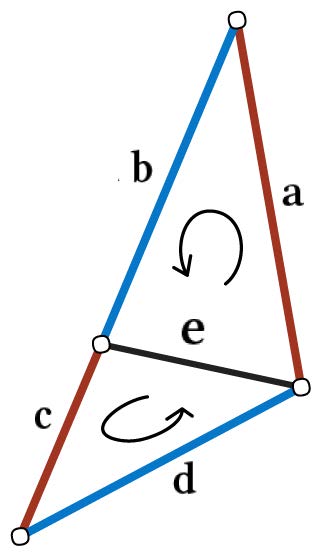
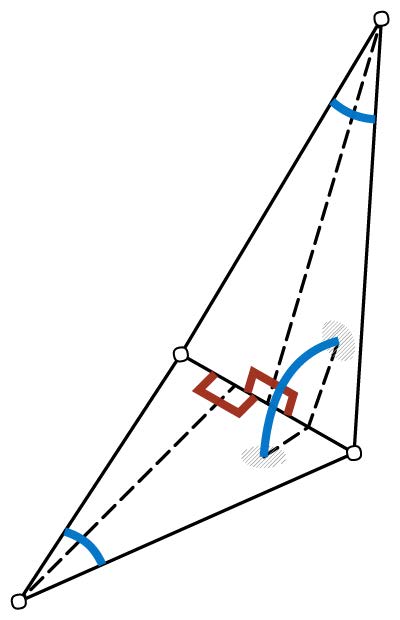
 ****

图4

在我们的网络中，我们采取了两个措施来解决这一问题并保证相似性变换（旋转、平移、缩放）的不变性。首先，我们仔细设计了一条边的输入描述子，使其只包含相对的、关于相似性变换具有固有不变性的几何特征。其次，我们将1-ring的四条边聚成两对含有歧义性的边（如，*a*和*c*，*b*和*d*），并通过在各对边上应用简单的对称性函数来生成新的特征（如，sum(*a*,*b*)）。卷积应用在新的对称的特征上，因此消除了顺序上的歧义性。

**3.2 输入特征**

对于每条边，输入的边特征是一个5维向量：二面角（dihedral angle）、邻接三角形的两个内角（inner angles）以及两个边长比率（edge-length ratios）。这个边长比是该边与邻接三角面片上垂线（如图中虚线所示）的比值。我们队这两个基于面片的特征（内角和边长比）进行了排序，从而解决了顺序上的歧义性并保证了不变性。可以观察到所有这些特征都是相对的，这使得它们关于平移、旋转和均匀缩放（uniform scale）变换都具有不变性。

**3.3 全局排序**

边的全局顺序是指一个特定形体的边的数据（输入特征）进入网络的顺序。这一排序在卷积阶段是没有影响的，因为卷积只在局部邻域内进行。更进一步说，完全卷积化的任务（如，分割）都不受其影响。对于那些需要全局特征聚合的任务（如，分类），我们遵循[Qi et al. 2017a]在PointNet中所提出的惯例，使用一个全局平均池化层连接网络的卷积部分和全连接部分。这一层使得初始排序变得无关紧要，从而保证了转换的不变性。

**3.4 池化**

如图2中(b)和(c)所示，网格池化由一个边坍缩过程完成。在(b)中，虚线那条边被探索成一个点，随后四条邻接的（蓝色的）边合并成(c)中两条（蓝色的）边。注意到在这个边坍缩操作中，五条边变成了两条。这一操作是根据（最小范式）边特征的优先顺序来进行的，因此允许网络选择哪部分网格将要被简化，哪部分将要被保留。这造就了一个任务驱动的过程，网络学会根据其任务来确定对象部分的重要性（见图1）。

我们这一简化过程的显著优点是，在到达最终的全连接层之前，它提供了关于池化层输出维度的灵活性。池化同时有利于初始网格三角剖分的鲁棒性。尽管它没有提供三角剖分的等价性，但在实践中，通过不断地坍缩边和简化网格，我们观察到不同的初始网格收敛到类似的表示。

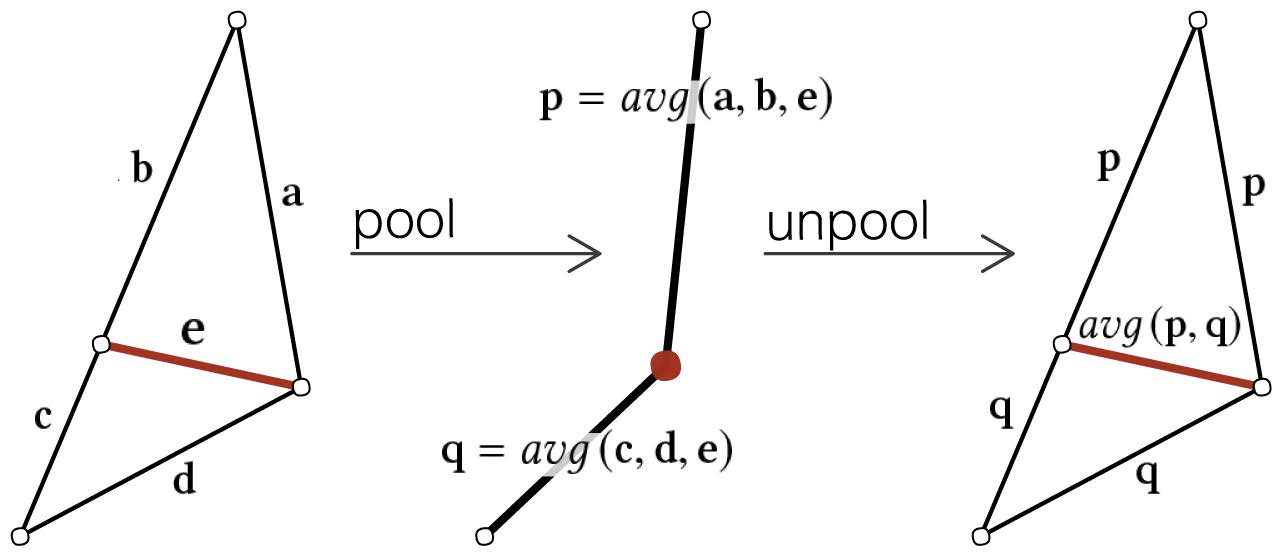


图5. 网格池化与反池化过程中的特征聚合

4. 方法

一个基于栅格（如，图像）的表示方法在一个单一矩阵中方便地提供了空间邻域（连通性）和特征。然而，因为不规则的网格并不遵守这一形式，我们必须将特征与连通性分开定义。我们在网格的标准构造中实现这一点。

一个网格由一对(*V*, *F*)所定义，其中*V* = {*v*1, *v*2, …}是中顶点位置的集合，而*F*则定义了连通性（三角网格的顶点三元组）。给定(*V*, *F*)，网格的连通性也可以用边集*E*来定义，其是顶点对的集合。

所有这些网格元素*V*、*F*和*E*都可以与各种特征相关联（如，法向和颜色）。在本项工作中，*E*也保存了一组特征。这些边的特征一开始是一组具有相似不变性的几何特征（相当于图像中的RGB值），随着它们在网络中的层间传播而形成更高级的抽象。

在我们的设定中，网格为网络提供了两种属性：卷积邻域的连通性和初始的几何输入特征。一旦输入特征被提取，网格顶点就没有意义了。边坍缩操作产生的新顶点位置对分类和分割任务没有影响，它们被计算出来仅用于可视化的目的。

接下来，我们将扩展并提供关于我们的网格卷积、网格池化和网格反池化操作的细节。

**4.1 网格卷积**

我们为边定义一种卷积操作，其在空间上通过四条相邻的边来定义（如图3）。回想一下，卷积是一个核（kernel）*k*与一个邻域的点积，因此一条边特征*e*和它邻边的卷积被定义为：

其中是*e*的卷积邻域中的第*j*个特征。如图4所示，*e*的四邻域，即是或，此时滤波器的每个值最多将在两条可能的边上做运算（如，对于，作用在*a*或*c*上）。为了保证卷积操作关于输入数据顺序的不变性，我们对有歧义的边对施加一组简单的对称性函数。这将产生一组确保具有不变性的新的卷积邻域。在我们的设定中，一个边*e*的感受野（receptive field）被定义为

这导致了一个对网格元素的初始顺序不敏感的卷积操作，因此不管顺序如何它都会产生相同的输出。回想一下，一个多通道张量与一个核的卷积可以由通用矩阵乘（GEMM）来实现：通过将图像扩展（或展开）成一列矩阵（即，*im2col*[Jia 2014]）。等价地，我们也构建了一个展开矩阵去高效地执行卷积操作。

在实践中，我们能通过将所有边缘特征聚合成的特征张量以使用高度优化的批操作（如，*conv2D*）,其中是边的数目，是特征通道的数目，*5*是边和卷积邻域的数目（等式2）。这一矩阵使用标准的*GEMM*与卷积的权重矩阵相乘。

在卷积运算之后，一个新的批特征张量（batched-feature-tensor）将被生成，其中新的特征数等于卷积核的数目（和图像中的一样）。注意到每个池化阶段后，新的连通性会为下一层卷积定义新的卷积邻域。

**4.2 网格池化**

在不规则数据上扩展卷积池化之前，我们先通过明确三个核心操作来推广池化的概念：

1. 对给定的邻接关系定义池化区域
2. 合并每个池化区域的特征
3. 为被合并的特征重定义邻接关系

对于诸如图像等规则数据上的卷积，邻接关系是固有的，因此池化区域直接取决于所选卷积核的大小。由于每个区域的特征都以一种能产生新的均匀间隔栅格的方式被合并（如，通过*avg*或*max*），新的邻接关系再一次被内在地定义。在解决了上述三个通用的池化操作之后，传统的池化显然是该泛化过程的特例。

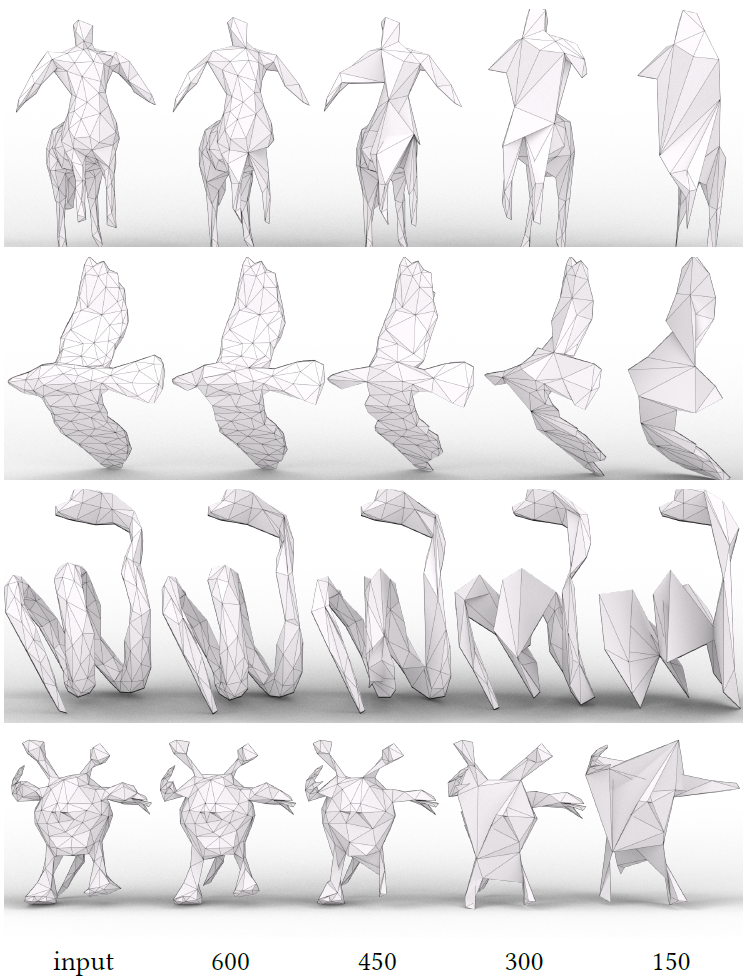


图6. SHREC 11分类数据集中间阶段被池化了的网格。输入网格都先被简化成大概750条边（500个面片），随后被池化成600、450、300和150条边。

网格池化是泛化池化的另一个特例，其邻接关系由拓扑所决定。图像拥有自然的缩减因数，例如对尺寸4使用池化。与之不同，我们将网格池化定义成一系列边坍缩操作，它们将五条变转换成两条。因此通过增加一个定义池化后网格中目标边数目的超参数，我们可以在每次池化操作后控制网格的分辨率（*resolution*）。在运行时，提取网格的邻接信息需要查询不断更新的特殊数据结构（细节详见[Berg et al. 2008]）。

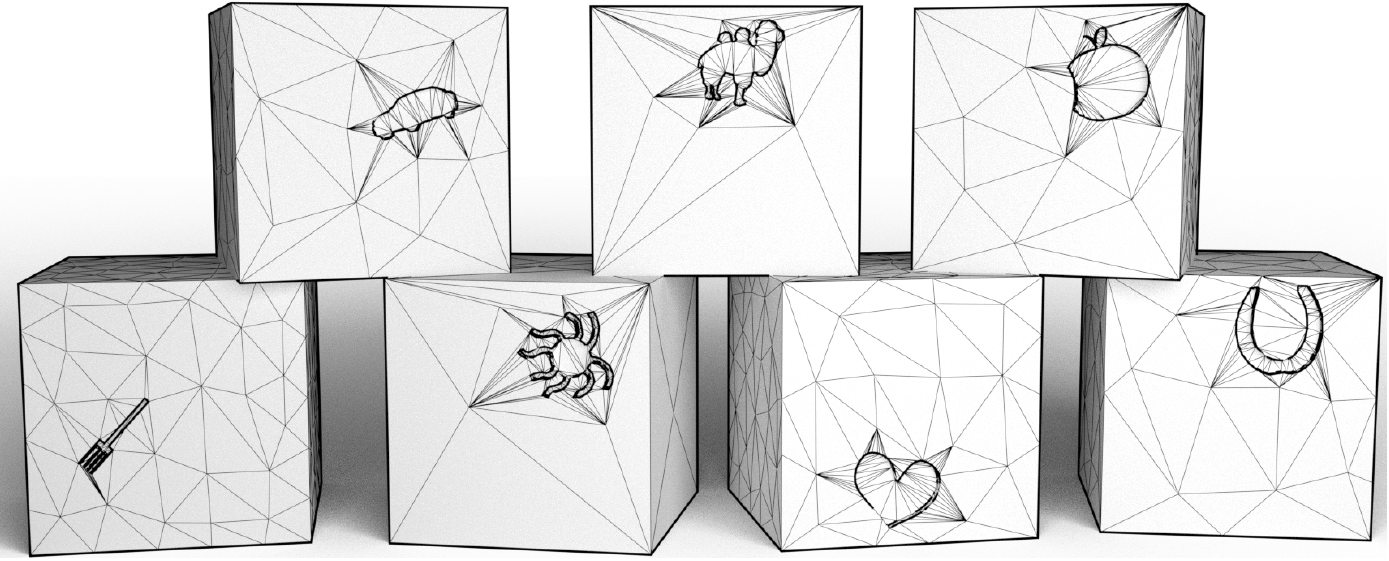


图7. 雕刻了的立方体分类数据集。我们通过从MPEG-7[Latecki and Lakamper 2000]中取出贴图来生成23种不同的类，并将它们放置到一个随机位置的一个随机面片上。

我们通过边特征的大小来对边坍缩的顺序进行优先排序（使用一个优先队列），这允许网络选择网格的哪部分与解决任务相关。这使得网络能非均匀地坍缩那些对loss最不重要的区域。回顾到坍缩一条与两个面相邻接的边将删掉三条边（如图2所示），因为每个面都变成了一条边。每个面包含了三条边：最小的边与其两条邻边（见图2中红色的最小边与其蓝色的邻边）。通过对所有特征通道取均值，每个面上三条边的特征都被合并到一条新的边的特征当中。

边坍缩的优先顺序依据的是边特征的强度，其通过边特征的-norm来计算。特征聚合的方式如图5所示，此处有两个合并操作，分别针对最小边特征*e*的两个邻接三角进行，最终生成两个新的特征向量（用*p*和*q*来表示）。通道*i*中两个三角形新的边特征定义如下：

这半边数据结构在边坍缩后被更新，用于后续的边坍缩操作。

最后，注意到并非每条边都可以被坍缩。在我们的设定中，一条边坍缩后产生一个非流形的面是不被允许的，因为它违背了四个卷积邻域的假设。因此，如果一条边在其1-ring的交集处有三个顶点，或者它有两个边界顶点，这条边的坍缩将被认为是无效的。

**4.3 网格反池化**

反池化是池化操作的（部分的）反转。池化层降低特征激活值的分辨率（编码或压缩信息），而反池化层则增加特征激活值的分辨率（解码或反压缩）。池化操作记录了来自合并操作（如，最大位置）的历史，并用它们扩展特征激活值。因此，反池化并没有可学习的参数，它通常与卷积结合在一起以恢复池化操作中丢失的原始分辨率。与卷积的结合有效地使反池化成为一种可以被学习的操作。

每个网格反池化层都与一个网格池化层成对存在，以对网格的拓扑和边的特征进行上采样。这一反池化层通过存储池化操作前的连通性来恢复上采样的拓扑（在网格池化之前的）。注意到对连通性进行上采样是一个可逆的操作（就像在图像中的一样）。对于反池化边的特征计算，我们维护了一个图（graph），其存储了从原始边（池化之前）到新边（池化之后）的邻接关系。每个反池化的边特征是池化后边缘特征的带权组合。平均反池化的示例如图5所示。

5. 实验

MeshCNN是将卷积神经网络直接应用到三角网格上的一种通用方法，其具有广泛的应用前景。使用在第4节中所述的MeshCNN的构建模块，我们能为以解决不同任务为目的构造不同的网络配置。像传统的卷积神经网络一样，这些构建模块提供了一种“即插即用”（*plug-and-play*）的框架。出于计算效率的考量，在池化操作中，我们只为每个池化进行一次特征聚合。当边的排序和坍缩被序列化执行时，这种松弛性允许其在GPU上进行特征聚合操作，这将提高计算效率。

接下来，我们将展示MeshCNN在分类和分割任务上的表现。附录A将给出所使用的网络架构的细节。

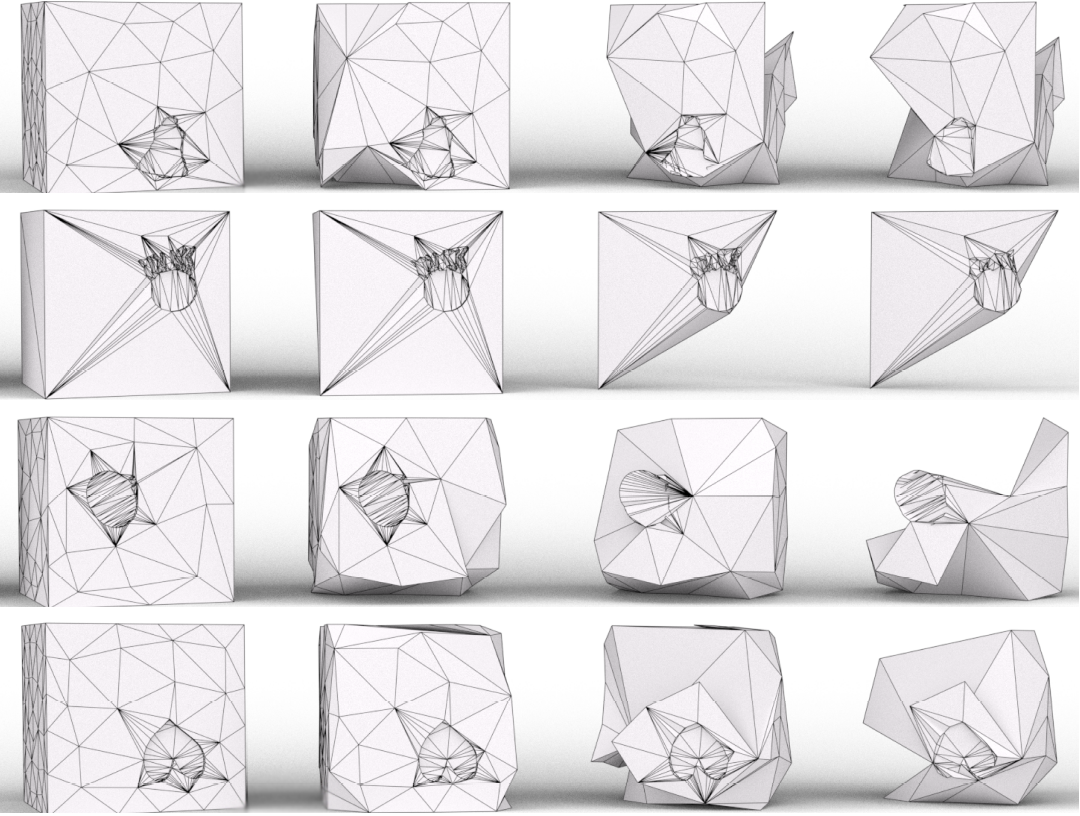


图8. 训练MeshCNN去预测图标雕刻的种类。可观察到该网络如何保留对分类任务来说重要的边，并取出冗余的边。

**5.1 数据处理**

在所有的数据集中，我们都将每一个网格简化成大致相同数目的边。正如早先所提到的，MeshCNN并不要求所有的样例有相同数目的边。然而，与卷积神经网络中对图像的初始resize操作类似，几何网格抽取（decimation）有助于减小输入分辨率及训练所需的网络容量。由于分类任务学习一个全局的形体表示，我们通常使用一个较低的分辨率（750条边）；分割任务则与之相反（2250条边）。

**5.1.1 数据增强**

对于本网络，有几种现存的用于生成更多数据样例的数据增强方式。因为我们的输入特征是相似性无关的，所以应用旋转、平移和各向同性缩放（在*x*、*y*和*z*上等尺度放缩）变换都无法生成新的输入特征。但是，我们能通过在定点位置的*x*、*y*和*z*上应用各向异性的缩放（anisotropic scaling）——（每个值都从一个和的正态分布中随机采样）来生成新特征，它能改变网络中的输入特征。我们还将网格表面上的点平移（shift）到不同的位置。此外，我们通过执行随机的边翻转来增强每个物体的网格划分。由于输入分辨率的灵活性，我们能在训练之前坍缩一小部分随机的边。

**5.2 网格分类**

**5.2.1 SHREC**

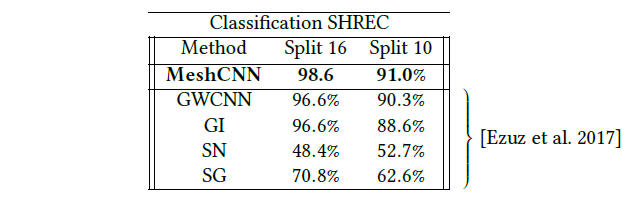


表1. SHREC 30类分类任务（与[2017]中的结果作对比）。Split 16和Split 10都是训练集划分，至多训练200个epoches。

我们对来自SHREC数据集[Lian et al. 2011]的30类数据进行了分类，其每类有20个样例。我们按照[Ezuz et al. 2017]中的设定，将每类训练集的数目划分为16和10，在200个epochs后停止训练。因为我们没有使用[2017]里的精确划分，我们的结果是从3个随机生成的16和10划分数据集中取平均。表1对结果进行了展示。作为比较，我们直接从[2017]中取评估值，其对下述方法进行了对比：SG [Bronstein et al. 2011] (bag-of-features representation，特征袋表示法)、 SN [Wu et al. 2015] (volumetric CNN，体积卷积神经网络)、GI [Sinha et al. 2016] (CNN on fixed geometry images，固定几何图像上的卷积神经网络) 以及GWCNN[2017]（可学习的几何图像）。如图6所示，我们方法的优势显而易见。我们对数据集中一些网格池化简化的示例进行了可视化。我们观察到网格池化以一致的语义方式进行（如图11所示）。

**5.2.2 立方体雕刻（Cube engraving）**

为了展示MeshCNN的独特能力，我们对一组带有浅图标（icon）雕刻的立方体进行建模（如图7所示）。我们使用来自MPEG-7二进制形体[Latecki and Lakamper 2000]数据集的23类，每类有大概20个图标。我们为每类留出3个图标作为测试集，剩下的用于训练。对于每个图标，我们随机采样了10个不同的位置（位置、旋转和立方体面）用来嵌入图标。每个立方体有大约500个面片，这意味着平坦区域的细节形体有更少的三角形，而平坦区域中较不精细的形体有较多的三角形。这个数据集总共包含4600个形体，其训练集/测试集的划分为3910/690.我们计划在出版发布这一数据集及其数据合成代码。

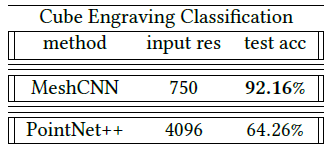


表2. 雕刻立方体的分类结果（如图7所示）。

我们训练MeshCNN去分类这些立方体。在表2中我们展示了定量的结果。为了可视化网格池化对分类任务的影响，我们提取了每次网格池化操作后的中间结果（如图8所示）。我们可以观察到MeshCNN是如何学会削减那些与分类任务无关的边（平的立方体表面），同时保留那些在雕刻图标内或周围的边。

我们同时在该数据集上训练了基于点云的方法，结果也展示在表2中。虽然这个例子可能会被认为是有意为之，但它是用来强调MeshCNN在包含大量几何分辨率差异的三维形体上所表现出的优势。

**5.3 网格分割**

MeshCNN的另一个应用是一致性形体分割，它是形体分析与合成中许多应用的重要构件。在COSEG [Wang et al. 2012]和人体分割[Maron et al. 2017]数据集上，对于每条边，我们使用监督学习的方法训练MeshCNN去预测其属于特定部分的概率。由于这两个数据集都提供了每个面片分割的ground truth，我们基于原始分辨率上的标签，在简化后的网格上生成edge-level的语义标签。

最直接的MeshCNN语义分割设置是使用一系列网格卷积层（以及正则化和非线性激活单元）。然而，合并的网格池化是的MeshCNN有能力学得语义驱动的边坍缩。考虑到网格池化减少了输入网格的分辨率，使其不再与ground truth的edge-level标签保持一致。为此，我们采用网格反池化层来进行上采样，将分辨率恢复至初始输入的大小。

**5.3.1 COSEG**

我们评估了MeshCNN在COSEG数据集上执行分割任务的表现。该数据集包含三大类：外星人、花瓶和椅子，每类分别含有200、300和400个模型。我们对每个形体类划分成85%/15%的训练集/测试集比率。我们与PointNet、PointNet++和PointCNN做了比较，表3展示了所有模型的最佳准确率。在这个数据集上，我们的技术达到了比所有其他方法都更好的结果。

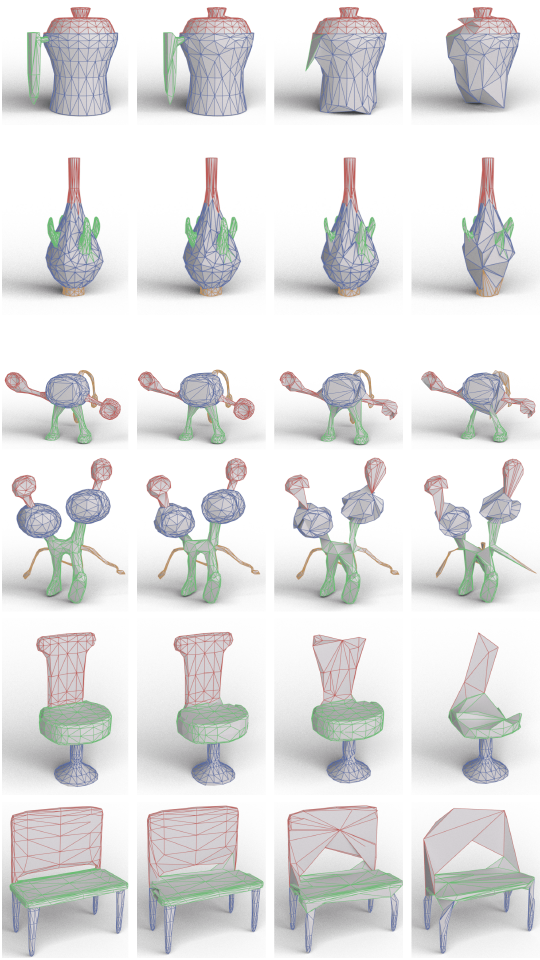


图9. 在测试形体上的语义分割结果。最左边展示的是每条边的分割预测值，后面展示的是中间每个池化层后的简化网格。为了可视化，中间过程的网格上的边被涂成了最终分割预测值的颜色。MeshCNN使用网格池化去学习从相同的语义上坍缩边，并通过网格反池化层展回成初始输入网格的分辨率（如最上面一行所示，整个花瓶的把手已经被坍缩成一条边）。

我们相信这是由于我们的网络视为网格结构专门定制的，这使得它比其他策略更具优势。为了进一步说明这一点，我们还报告了随机池化（要坍缩的边是随机挑选的）情况下的结果，并表明这种变化降低了网络的性能。此外，带有池化和反池化层的MeshCNN语义分割网络在一个给定测试集上的最终语义预测结果如图9所示。这也展示了所用的池化操作是如何适应于目标问题。

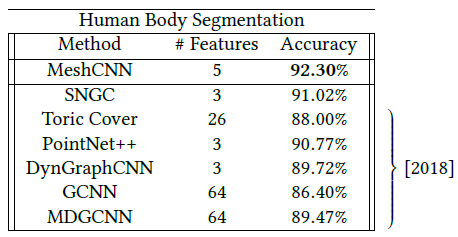


表4. 人体分割结果（从[Poulenard and Ovsjanikov 2018]中引用的对比）。最后一个对比引自[Haim et al. 2018]。

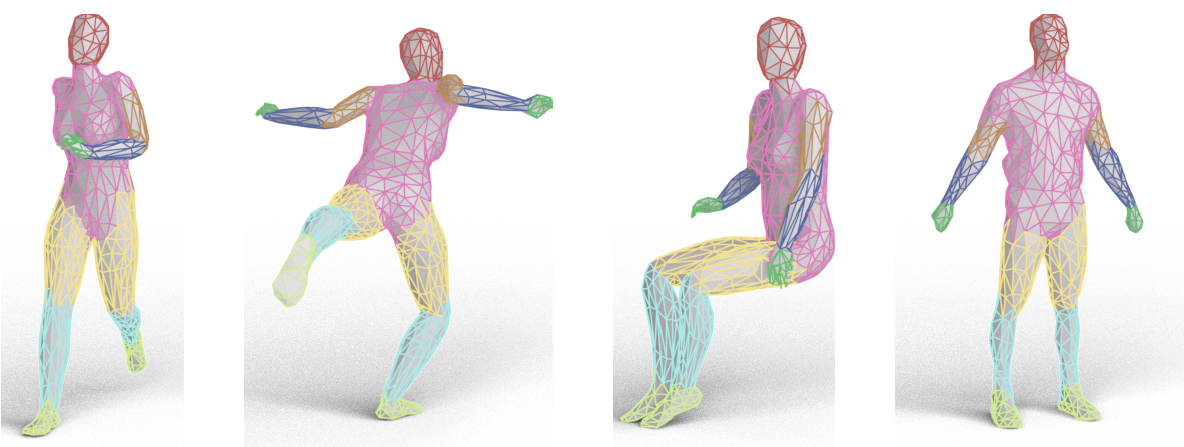


图10. 在数据集[Maron et al. 2017]上的人体分割结果。

**5.3.2 人体分割**

我们在由[Maron et al. 2017]提出的人体分割数据集上评估我么你的方法。该数据集包含来自SCAPE [Anguelov et al. 2005]、FAUST [Bogo et al. 2014]、MIT [Vlasic et al. 2008]和Adobe Fuse [Adobe 2016]的370个训练模型，和来自SHREC07 [Giorgi et al. 2007]（人体）数据集的18个测试模型。根据[Kalogerakis et al. 2010]中的标签，这些模型被手工分割成8个标签。近来，[Poulenard and Ovsjanikov 2018]汇报了他们的方法在这个数据集上与GCNN [Masci et al. 2015]、PointNet++ [Qi et al. 2017b]、Dynamic Graph CNN [Wang et al. 2018b]及Toric Cover [Maron et al. 2017] 作对比所得的结果。我们直接取[Poulenard and Ovsjanikov 2018]的结果并将其列在表4中。我们在表中增加了[Haim et al. 2018]最近的结果，其汇报了在本数据集上最领先的结果。在这个例子中，MeshCNN同样比其他方法（一些是基于图/流形的，另一些是基于点云的）更具优势，我们相信这是MeshCNN对网格结构和当前任务的适应性的结果。图10展示了MeshCNN的一些定性结果。

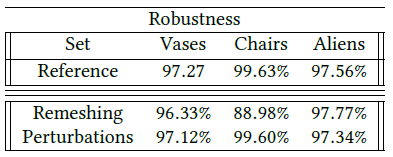


表5. 在两个鲁棒性数据集上的定量结果。

**5.4 附加评估**

**5.4.1 计算时间**

使用一块GTX 1080 TI图形显卡在2250/750边集上做分割和分类的训练，我们未经优化的PyTorch [Paszke et al. 2017]实现在每个样例上平均花费0.21/0.13秒。

**5.4.2 网格细分的鲁棒性**

我们使用COSEG分割数据集进行数个定性和定量实验，以此来检验我们的方法在不同三角剖分上的鲁棒性。为此，我们生成了数据集的两个修改版本。第一个是利用一个网格重建（remeshing procedure）过程所获得的（使用Blender和MeshLab），第二个是通过向1-ring中的一个随机顶点做偏移以对30%的顶点位置做随机扰动来实现。该表现上的细微差距（见表5）显示了其在网格细分变化上的弹性。图12中展示了定性结果。

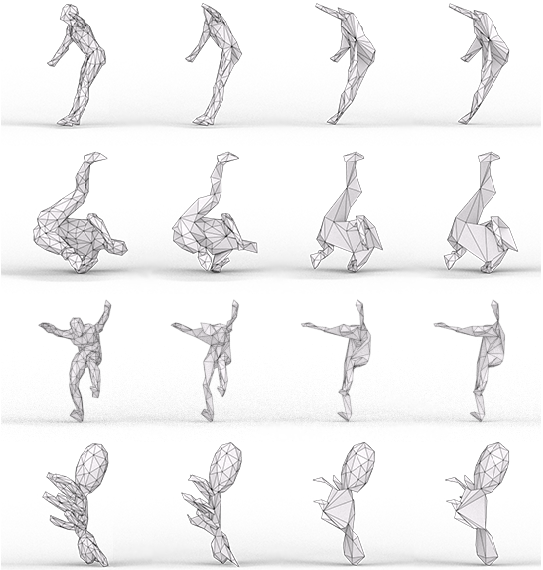


图11. 自适应池化具有在类中暴露出语义的潜力。我们观察到与形体分类任务（SHREC）中在同类上做语义池化的一致性。例如，人的头以类似的方式被池化，而另一类（底部）中相似的形体属性却以不同的方式被池化。

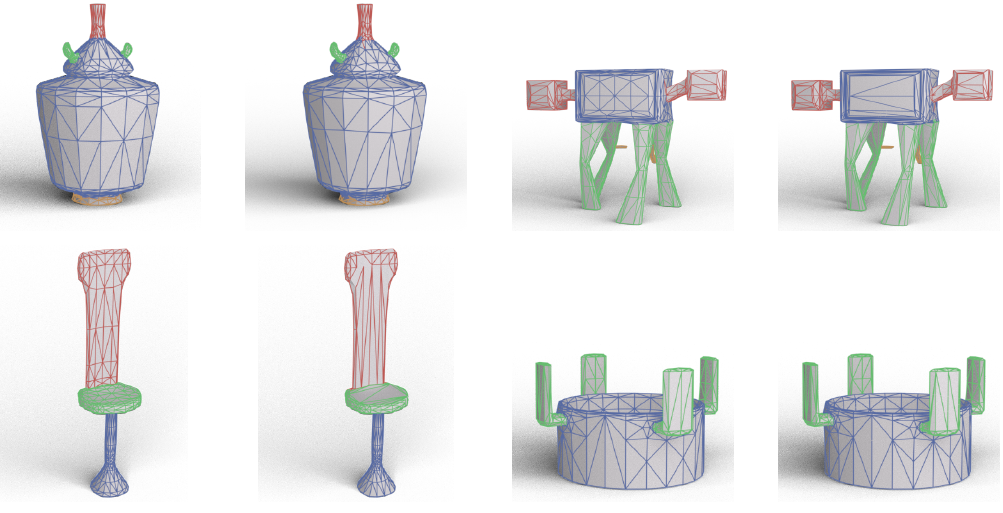


图12. 针对不同三角剖分的健壮性（语义分割）。左：测试形体，右：可选的三角剖分。

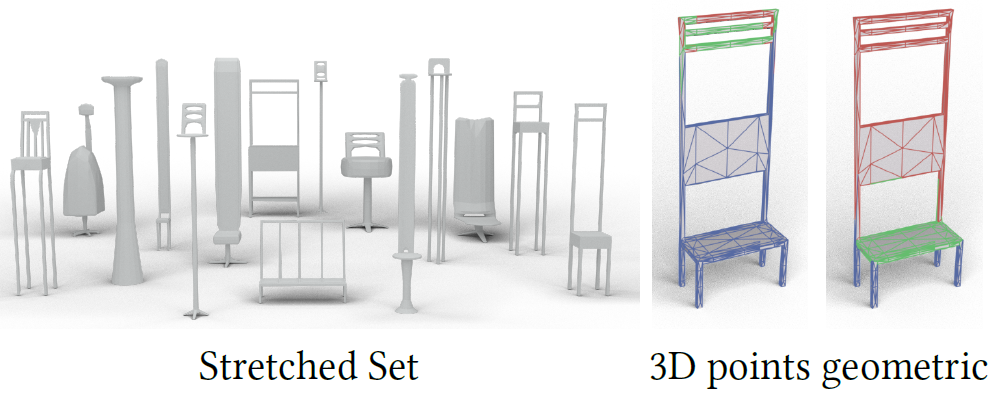


图13. 可微特征的鲁棒性。我们在（左侧）拉伸数据集（Stretched Set）上评估泛化性，使用带几何特征的MeshCNN，并与标准三维点方法（边的中点）作对比。虽然两者在标准测试集上都达到了很高的精度，但MeshCNN的几何特征(右)与三维点云的情况(中)相比可以更好地对拉伸集进行泛化。

**5.4.3 不变性特征**

使用相对特征的一个显著优点是，确保了MeshCNN关于旋转、平移和均匀缩放的不变性。常用的笛卡尔坐标先天就对刚体变换敏感。为了展示这一点，我们在语义分割任务上训练MeshCNN：(i)使用不变性的几何特征和(ii)使用边的中点（*x*, *y*, *z*）作为输入特征。为达到学习的泛化性，我们沿着垂直坐标做非均匀缩放（在训练时不使用这类数据增强）。与标准测试集的99.63%的结果相比，我们的相对几何特征达到98.44%的结果；而绝对坐标的结果退化到78.27%，与标准测试集的99.11%形成对比。需要注意的是，虽然我们的几何特征关于非均匀缩放具有不变性，但由于它们对定位不敏感，所以可以被更好的泛化。

6. 总结与展望

我们提出了MeshCNN，这是一种将神经网络直接应用于不规则三角形网格的通用方法。我们的工作的关键贡献是定义和应用了卷积和池化运算来适应不规则和非均匀的结构。这些操作有助于用最原生的方式对以网格表示的形体进行直接分析，并因此得益于与非均匀结构的表面流形表示法相关的独特属性。

**6.1 不变性卷积**

我们在作为网络运算基础构建的网格边上的选择是极其重要的，因为这一边集规定了一种简单的方法去为非规则结构上的卷积定义一个局部的、固定尺寸的邻域。利用三角网格独特的对称性，我们消除了邻域顺序的二元歧义，使得变换具有不变性。我们通过选择输入边缘特征来完成这一工作，这些特征都经过精心设计，只包含相对的几何属性，而不是绝对位置。因此，与常见的表示法（如，基于点云的）不同，顶点的笛卡尔坐标被忽略了，局部和非局部特征是位置无关的，这允许对形体特征进行更好的泛化，从而促进其对相似形变换的不变性。需要强调的是，我们使用顶点位置只是为了展示这些不断演化的网格，而这些位置对任务本身并无影响。

**6.2 空间适应性池化**

我们已经基于习得的边特征，通过边坍缩来实现一个池化操作，其基于学习到的边特征，从而实现了任务驱动的、网络损失函数导向的池化。在未来，我们想要添加一组专门的、独立的特征用于区分边坍缩的优先级，类似于注意力（attention）模型。对网络认为是重要的一系列特征进行可视化，提供了对网络学到什么的深刻见解。与使用绝对笛卡尔坐标相反，我们观察到我们的可微特征不仅提供了相似变化的不变性，而且还抑制了过拟合。通过跨越不同对象实现语义相似性池化，本网络的泛化性将得到进一步证明，这自然会导致更好的结果。研究这种强大的机制可能有助于更好地理解神经网络行为。

我们认为这种空间适应性的、不规则的、任务驱动的池化是一个重要的贡献，它也可能会影响到许多基于图像的卷积神经网络任务。例如，高分辨率图像分割通常会产生低分辨率的分割图并对其进行上采样，可能还应用了跳跃（skip）连接。MeshCNN中的池化操作在语义上简化了均匀特征的区域，而保留了复杂的特征；因此，在未来，我们有兴趣将类似的不规则池化应用到图像分割任务上以获得高分辨率的语义图，其大的均匀区域将通过小数目的三角面片来表示。

目前，我们的实现执行了序列化的边坍缩。这一操作也许可以通过使用一种边特征的并行排序技术[Bozidar and Dobravec 2015]而在GPU上实现并行，同时确保只有不相邻的边被同步坍缩

虽然我们的方法对不同的三角剖分具有健壮性（如我们实验中所展示的），但是MeshCNN与任何其他网络一样，依赖于好的训练数据才能做到成功泛化。从这种意义上来说，与图像中的对抗性噪声（adversarial noise）类似，MeshCNN容易受到对抗性网格重建攻击（adversarial remeshing attacks）的影响，这可能会损害网络的表现。关于这种对抗性攻击的健壮性是未来工作的一个有趣方向。

未来研究的另一个方向是生成式建模、网格上采样和对已有网格进行修改的属性合成。我们的想法是通过簿记（bookkeeping）边坍缩列表，以边坍缩操作的逆序来应用顶点分离技术（vertex-split）。这有点类似上采样层中所使用的的bookkeeping技术。因此，在合成新网格时，由网络决定要分离哪个顶点，例如，分割与具有高特征值的边相邻的顶点。

最后，我们确定了一个有前景的方向，就是将我们专为三角网格设计的策略扩展到一般的图（graph）上。基于边坍缩的池化和反池化能被以与我们的MeshCNN中类似的方式应用到通用的图上。至于卷积操作，我们必须考虑一种与一般图的不规则性相适应的选项。一种有趣的方法可能是使用注意力机制[Monti et al. 2018]来处理边。

参考文献

[1] Adobe. 2016. Adobe Fuse 3D Characters. http://www.mixamo.com.

[2] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian, Thrun, Jim Rodgers, and James Davis. 2005. SCAPE: Shape Completion and Animation of People. In ACM SIGGRAPH 2005 Papers(SIGGRAPH‘05).ACM, New York, NY, USA, 408-416. https://doi.org/10.1145/1186822.1073207

[3] James Atwood and Don Towsley. 2016. Diffusion-convolutional Neural Networks. In Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS’16). Curran Associates Inc., USA, 2001–2009. http://dl.acm.org/citation.cfm?id=3157096.3157320.

[4] Matan Atzmon, Haggai Maron, and Yaron Lipman. 2018. Point Convolutional Neural Networks by Extension Operators. ACM Trans. Graph. 37, 4 (July 2018), 71:1–71:12.

[5] Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. 2008. Computational Geometry: Algorithms and Applications (3rd ed. ed.). Springer-Verlag TELOS, Santa Clara, CA, USA.

[6] Federica Bogo, Javier Romero, Matthew Loper, and Michael J Black. 2014. FAUST: Dataset and evaluation for 3D mesh registration. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 3794–3801.

[7] Davide Boscaini, Jonathan Masci, Simone Melzi, Michael M Bronstein, Umberto Castellani, and Pierre Vandergheynst. 2015. Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks. In Computer Graphics Forum, Vol. 34. Wiley Online Library, 13–23.

[8] Davide Boscaini, Jonathan Masci, Emanuele Rodolà, and Michael Bronstein. 2016. Learning shape correspondence with anisotropic convolutional neural networks. In Advances in Neural Information Processing Systems. 3189–3197.

[9] Mario Botsch, Leif Kobbelt, Mark Pauly, Pierre Alliez, and Bruno Lévy. 2010. Polygon mesh processing. AK Peters/CRC Press.

[10] Darko Bozidar and Tomaz Dobravec. 2015. Comparison of parallel sorting algorithms. CoRR abs/1511.03404 (2015).

附录A 训练设置



表6. 在SHREC和Cube engraving数据集上的分类网络配置。网络的初始边数是750。它们都使用了Adam优化算法，学习率*lr*设为0.0002，同时使用了组归一化（group norm）（*g* = 16）。数据增强方式是5%的边翻转和20%的顶点滑动。

对于分类任务，我们使用与SHREC和Cube engraving数据集相同的网络架构。我们在表6中展示了网络设置和学习参数的细节。

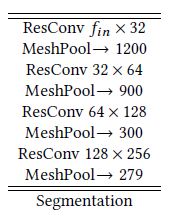


表7. 在COSEG和人体数据集上的分割网络配置。网络的初始边数是2250。这里指给出了网络的下半部分。上半部分与之相对称。

对于分割任务，针对COSEG和人体数据集，我们都是用了一种Unet[Ronneberger et al. 2015]类型的网络。表7提供了这一网络的细节。