保特征的单幅图像三维网格重建

邢燕[1,2], 马俊[1)*], 檀结庆[1)]

[1)] (合肥工业大学数学学院 合肥 230000)
[2)] (情感计算与先进智能机器安徽省重点实验室 合肥 230000)
(majun@mail.hfut.edu.cn)

摘 要: 针对图像重建三维物体方法中存在无法保持物体尖锐特征的问题, 基于深度神经网络, 对输入单幅图像提出一种有效的保特征三维网格生成方法. 对单幅输入图像使用 VGG-16 提取图像特征, 并特别设计了图像边缘检测层获取物体的尖锐特征; 将三维网格(初始为椭球)的顶点投影到特征图和边缘检测图上, 以获得顶点局部特征, 并判断其是否为尖锐特征点; 然后, 将局部特征和顶点位置串联输入到改进的图卷积神经网络(graph convolutional neural network, GCNN), 对于非尖锐特征点采用普通 GCNN, 对于检测到的尖锐特征点采用 0 邻域图卷积神经网络(0-neighborhood GCNN, 0N-GCNN), 以期其尽量不被邻域顶点过度光滑; GCNN 的输出预测了顶点的新位置和三维特征; 最后, 对网格的顶点及特征用 Loop 细分上采样. 执行 3 次上述变形(二维特征投影、尖锐特征检测、GCNN 变形、上采样)后, 初始椭球最终变形为输入图像中物体模样. 实验使用 ShapeNet 数据集, 在 PyTorch 框架下实现, 从定性和定量两方面与现有方法进行了比较. 实验结果表明, 在 Chamfer 距离和 *F*-score 两类定量指标上均优于大部分现有方法, 而 Chamfer 距离和 *F*-score(2τ)的均值表现为最优. 视觉比较也表明, 文中方法可有效地提升特征保持性能.

关键词: 图卷积; 网格重建; 特征检测; 深度学习
中图法分类号: TP391.41 **DOI:** 10.3724/SP.J.1089.2023.19375

Feature Preserving Mesh Reconstruction from a Single Image

Xing Yan[1,2], Ma Jun[1)*], and Tan Jieqing[1)]

[1)] (*School of Mathematics, Hefei University of Technology, Hefei* 230000)
[2)] (*Anhui Province Key Laboratory of Affective Computing & Advanced Intelligent Machine, Hefei* 230000)

Abstract: The reconstruction of 3D objects from image with the problem of failing to maintain sharp features of objects. Based on deep neural network, an effective feature preserving 3D mesh generation method is proposed for a single input image in this paper. Firstly, image features are extracted using VGG-16 for the input images, and the image edge detection layer is specially designed to obtain the sharp features. Secondly, the vertices of the mesh (initially ellipsoid) are projected onto the feature map and edge detection map to obtain the local features of the vertices, and judge whether they are sharp feature points. Thirdly, the local features and positions of the vertices are concatenated and input into the improved graph convolution neural network (GCNN). For the non-sharp feature points, the ordinary GCNN is used, and for the detected sharp feature points, the 0-neighborhood graph convolution neural network (0N-GCNN) is used to avoid being over-smoothed by the neighboring vertices as much as possible. The output of GCNN predicts the new position and features of the vertices. Finally, the vertices and features of the mesh are up sampled by Loop sub-

收稿日期: 2021-09-23; 修回日期: 2022-01-08. **基金项目**: 国家自然科学基金(62172135); 中央高校基本科研业务费专项资金(PA2020GDSK0060). **邢燕**(1977—), 女, 博士, 副教授, 硕士生导师, CCF 会员, 主要研究方向为计算机辅助几何设计与计算机图形学、图像处理、深度学习; **马俊**(1997—), 女, 硕士研究生, 论文通信作者, 主要研究方向为三维建模; **檀结庆**(1962—), 男, 博士, 博士生导师, 主要研究方向为非线性科学计算.

division. After going through above deformation process (2D feature projection, sharp feature detection, deformation by GCNN, upsampling) three times, the initial ellipsoid is finally transformed into the shape in the input image. The experiments are implemented on ShapeNet dataset based on PyTorch framework. The proposed method is compared with the existing methods quantitatively and qualitatively. The experimental results show that this method is superior to most existing methods in both Chamfer distance and *F*-score, and the mean values of Chamfer distance and *F*-score(2τ) are the best. Visual comparison also shows that this method effectively improves the feature preservation performance.

Key words: graph convolution; mesh reconstruction; feature detection; deep learning

从图像中重建物体一直是计算机视觉领域中一项重要工作, 而从单幅图像中恢复真实物体的三维网格形状更是极具挑战性的任务. 通常做法是借助一些硬件(如激光扫描仪)对物体进行多次扫描, 再重建物体. 但是这种方式需要的硬件设备价格昂贵, 如激光扫描仪, 而且有些物体也不方便进行扫描操作. 于是, 人们开始考虑研究直接从图像得到物体的三维表示的方法. 由于自然界中的各种物体的图像是容易获得的, 因此该方法是便捷的. 早期大多数的工作是从图像重建出物体的体素表示[1-2], 即用三维空间中一些排列规则的小立方体体素来建模物体, 其类似于二维图像中的像素, 是二维卷积神经网络(convolutional neural network, CNN)的一种最直接的推广形式. 然而, 这种表示较消耗内存, 在相同的硬件设备(内存和处理器)条件下, 只能获得较低的分辨率. 随后, 有学者提出从单幅图像重构物体的三维点云表示[3-5], 点云表示相比体素表示更轻量级. 但点和点之间没有连接关系, 不能直接绘制出物体表面. 网格作为另一种轻量级的三维物体表示方法, 既克服了体素内存消耗大的缺点, 又能够比点云更好地建模物体的表面细节, 故受到了研究者们的青睐, 人们更倾向于选择使用网格建模物体. 还有从图像重建体素再转化为网格的方法[6-7], 或者从图像重建点云再转化为网格的方法[8-9], 它们均不能直接从图像得到三维网格. 而从单幅图像重构物体网格的方法[10], 据悉是首次利用三维监督从单幅图像获得三维网格的工作, 但是其仍然存在一些问题, 如图像中物体边缘的尖锐特征保持得不好. 故针对单幅图像三维网格重建中的特征保持问题, 提出了 0 邻域图卷积神经网络(0-neighborhood graph convolution neural network, 0N-GCNN).

1　相关工作

已有的工作, 如即时定位与地图构建(simultaneous localization and mapping, SLAM)[11]和运动恢复结构(structure from motion, SFM)[12], 是从采集的数据恢复物体的体素和点云表示. SFM[12]是一种传统的三维重建算法, 其能够采集三维数据再离线地还原出其点云形式. SLAM[11]能够并发地采集数据, 并建造增量式地图. 其表示为点云、体素、八叉树等形式. 网格表示法能够高分辨率地建模物体表面细节, 故本文采用网格进行物体建模.

从彩色图像中恢复物体的三维网格是一项较有意义且富有挑战性的工作, 众多学者对其不断探索. 目前已有部分基于单幅图像的三维网格重建工作[10,13-20]中, 文献[10]是最早一批提出从单幅图像复原出物体的网格形式的论文之一, 它通过训练一个神经网络模型, 并输入任意一幅图像得到物体的三维网格, 整个过程是端到端的, 不需要人工后期处理. 这项工作克服了以往三维重建工作需要昂贵设备、耗时且不容易操作等局限. 当然, 这项工作并不完美, 例如, 它不能较好地保持物体特征, 对于物体的边、角以及较细的部位复原效果差. 一些学者提出从多幅图像中恢复物体的三维形状的方法, 即对同一个物体在不同角度下的照片进行三维重建[21-23]; 以及从轮廓线中重建物体三维网格的方法[24]. 还有一些学者提出从 RGB-D 图像或者是全息图中恢复物体的三维形状[25]. 但是, RGB-D 图像需要使用深度摄像机, 其在现今生活中尚未广泛使用. 也有学者提出室内场景的恢复[6,26-27], 其不再局限于单个物体, 是针对复杂场景的三维重建.

经典的二维 CNN 在图像特征提取、图像识别、图像复原等方面大获成功, 以期能够把二维 CNN 扩展到三维重建上. 但是网格表示的三维物体不同于规则排列的二维图像像素阵列, 不能设计尺寸相同的卷积核, 可以采用 GCNN 把网格看作“图”进行三维重建.

二维 CNN 通常以图像的像素为单位, 通过中

心点像素和相邻点像素的加权和获取图像特征图. GCNN 是在更一般的图结构上使用卷积运算[28], 即

$$\boldsymbol{f}_p' = w_0 \boldsymbol{f}_p + \sum_{q \in N(p)} w_1 \boldsymbol{f}_q \tag{1}$$

其中, p 表示网格上的顶点; $\boldsymbol{f}_p$ 表示 p 的特征, q 表示顶点 p 邻域内的点; w_0 和 w_1 表示权重; $\boldsymbol{f}_p'$ 表示对顶点 p 卷积后的结果. 从式(1)可以看出, 卷积后顶点 p 的特征是和其邻接点特征的加权求和. 网格的顶点是图中的结点, 网格中的边表示图中结点之间的连接关系, 故三维网格就是图, 可以通过 GCNN 对三维网格进行卷积操作. 文献[10]是一种便捷的从图像获得物体三维网格的方法, 但其在特征保持方面存在不足. 为此, 本文在其基础上提出了改进方法, 以增强模型的特征保持能力.

2 本文方法

输入一幅输入图像, 通过 GCNN 对初始的椭球模板不断变形, 最终重建出物体的三维形状. 具体的网络架构如图 1 所示.

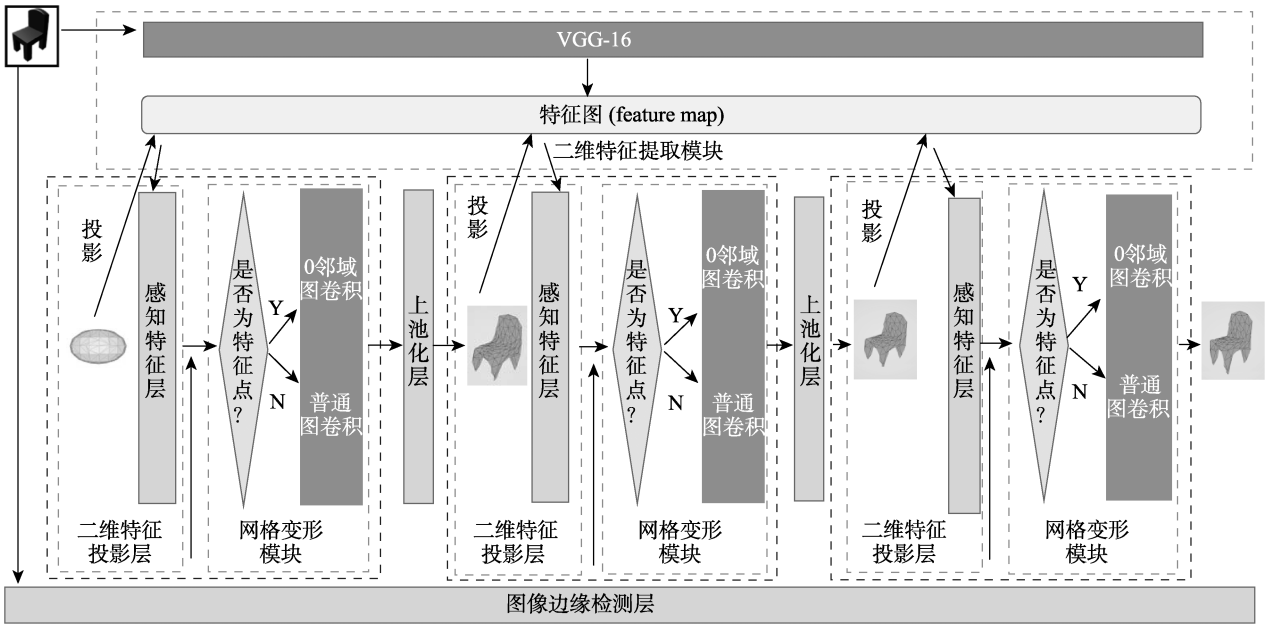


图 1 本文方法架构

本文提出的网络结构由二维特征提取模块(VGG-16)、图像边缘检测层、二维特征投影层、网格变形模块和上池化层等组成.

本文算法步骤如下:

Step1. 读入图像, 利用 VGG-16 获得图像特征, 并使用 Canny 算法获取图像边缘位置.

Step2. 把网格投影到二维特征图上, 利用双线性插值得到网格局部特征.

Step2.1. 将投影后得到的位置和边缘检测结果对比, 判断是否为尖锐特征.

Step2.2. 对尖锐特征点和非特征点分别使用 0N-GCNN 和普通 GCNN 进行变形.

Step3. 变形后的输出(更新后的顶点位置和三维特征)传入上池化层, 得到加密后的顶点集及其三维特征.

Step4. 将加密后的顶点输入二维特征投影模块得到投影的局部特征. 投影后的局部特征和加密的顶点坐标以及加密的三维顶点特征串联起来输给下一层网格变形模块.

网络设置 2 次上池化和 3 次变形得到最终的网格, 其过程如图 1 所示.

2.1 二维特征提取模块和二维特征投影层

VGG-16 可较有效地提取图像特征, 使用 VGG-16 网络提取图像特征, 获得不同尺度的二维特征图. 然后, 将待变形网格投影到不同尺度的二维特征图上, 用双线性插值获得网格顶点的非整型投影坐标处的特征, 把不同尺度的特征级联在一起得到网格顶点的局部特征. 这里不同尺度的特征分别取自 VGG-16 的 conv2_3, conv3_3, conv4_3 和 conv5_4.

2.2 图像边缘检测层

边缘检测广泛应用于图像处理和计算机视觉中, 通常用于特征提取和特征检测. 通过边缘检测得到图像中亮度明显变化的点或者不连续的区域, 最终得到一个二值图像, 以表示原图像的边缘. 目前, 边缘检测中经常使用的算法有 Sobel 算法、

Laplace 算法和 Canny 算法等. 其中, Canny 算法是目前使用最多也是相对比较完善的一种方法. 本文采用的是 Canny 算法, 首先使用边缘差分算子计算水平和垂直方向的差分 $\boldsymbol{G}_x$ 和 $\boldsymbol{G}_y$, 梯度计算公式为 $\boldsymbol{G}=\sqrt{\boldsymbol{G}_x^2+\boldsymbol{G}_y^2}$. 然后, 对梯度幅值进行非极大值抑制, 寻找像素点梯度的局部最大值; 最后, 使用双阈值算法检测和连接边缘.

边缘检测层将利用上述算法得到输入图像的二值边缘图像. 再将二维特征投影层得到的网格投影后的二维坐标和边缘图像比较. 如果投影后的二维坐标与图像中检测到的边缘位置重合, 则认为该点是尖锐特征点.

2.3　改进的 GCNN

传统的 GCNN 对当前顶点与邻域内的顶点特征进行加权求和, 邻接点的信息参与到当前顶点的信息计算, 可以得到顶点局部信息. 但这种做法对所有顶点并不总是好的(如边界点), 边界点的特征可能会被过度光滑, 因此使用改进的 GCNN 处理检测到的尖锐特征.

通常, GCNN 的特征矩阵表示为

$$\boldsymbol{H}'=\sigma\left(\boldsymbol{MHW}+\boldsymbol{b}\right).$$

其中, $\boldsymbol{H}\in\mathbb{R}^{n\times m}$ 表示特征矩阵; $\boldsymbol{M}\in\mathbb{R}^{n\times n}$ 表示邻接关系的矩阵, $\boldsymbol{M}=\boldsymbol{I}+\tilde{\boldsymbol{L}}\in\mathbb{R}^{n\times n}$; $\boldsymbol{W}\in\mathbb{R}^{m\times m'}$ 表示要学习的权重矩阵; n 表示网格顶点个数; m 和 m' 分别表示卷积前后的特征维度; $\boldsymbol{b}\in\mathbb{R}^{n\times m'}$ 表示偏差在所有行上的广播; 激活函数 σ 对所有行作用; $\boldsymbol{I}$ 表示单位矩阵, $\tilde{\boldsymbol{L}}$ 表示放缩归一化后的拉普拉斯矩阵, 即 $\tilde{\boldsymbol{L}}=\dfrac{2}{\lambda_{\max}(\boldsymbol{L})}\boldsymbol{L}-\boldsymbol{I}$. 其中, $\boldsymbol{L}=\boldsymbol{I}-\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}$ 是对称归一化的拉普拉斯矩阵; $\boldsymbol{A}$ 表示邻接矩阵; $\boldsymbol{D}$ 表示度矩阵; $\lambda_{\max}(\boldsymbol{L})$ 表示 $\boldsymbol{L}$ 的最大特征值. 拉普拉斯矩阵 $\boldsymbol{L}$ 放缩归一化后的特征值位于 $[-1,1]$, 保证拉普拉斯变换是一个压缩映射.

改进的 GCNN 的特征矩阵表示为

$$\boldsymbol{H}'=\sigma\left(\boldsymbol{M}'\boldsymbol{HW}+\boldsymbol{b}\right)\tag{2}$$

其中, $\boldsymbol{M}'_{j,\cdot}=\begin{cases}\boldsymbol{e}_j, & \text{if } j\in S\\ \boldsymbol{M}_{j,\cdot}, & \text{otherwise}\end{cases}$. $\boldsymbol{M}_{j,\cdot}$ 表示矩阵 $\boldsymbol{M}$ 的第 j 行; $\boldsymbol{e}_j$ 表示第 j 个元素是 1 的单位向量; S 表示检测到的特征点下标集合.

由式(2)可知, 当检测到特征点之后, 把矩阵 $\boldsymbol{M}$ 中特征点对应的行置为单位向量, 即除特征点对应索引位置是 1, 其余均为 0. 非特征点所在行不发生改变, 得到矩阵 $\boldsymbol{M}'$, 则在用式(2)进行 GCNN 运算时, 尖锐特征点的特征不再受邻接点的特征影响. 修改后的 GCNN 公式可避免尖锐特征受邻域特征影响而被过度光滑.

2.4　网格变形模块和上池化层

图 2 所示网格变形模块是由 GCNN 组成的. 本文设计了 3 个网格变形模块对初始椭球网格不断变形. 网格的顶点坐标 $\boldsymbol{C}_i$, 顶点特征 $\boldsymbol{F}_i$ 和局部特征 $\boldsymbol{f}$ 级联后一起传给变形块做 GCNN 运算. 在第 1 个变形块的输入中没有顶点特征 $\boldsymbol{F}_i$, 其原因是顶点特征是 GCNN 之后得到的.

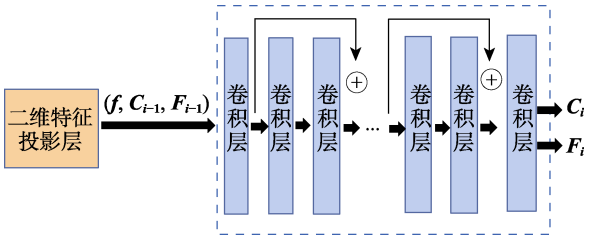


图 2　网格变形模块

如图 1 所示, 网格一共经历了 3 次变形, 每个变形块中有 14 个 GCNN 层, 利用如图所示的“快捷”连接形成了 6 个残差块. 使用残差网络是因为其更易于优化, 而且适合搭建更深的网络结构, 实验结果的精确度也更高. 更多残差网络的内容可以参考文献[29].

本文方法采用由粗到精的变形方式, 在变形块之间对网格进行加密, 称为上池化层(见图 1). 加密过程是在网格所有三角形的 3 条边中点处增加 1 个新顶点, 然后依次连接这 3 个中点, 则原来每个三角形一分为四, 每个新顶点的位置和特征用其所在边 2 个端点的位置和特征的平均来计算.

3　损　失

在神经网络反向传播时, 要借助损失函数更新网络参数, 设计恰当的损失函数能够让预测网格更好地逼近目标网格. 使用的损失函数包含了 4 项损失, 总损失是 4 项损失的加权和

$$l_{\text{all}}=l_{\text{c}}+\lambda_1 l_{\text{e}}+\lambda_2 l_{\text{lap}}+\lambda_3 l_{\text{n}}\tag{3}$$

其中, 系数 λ_1, λ_2, λ_3 用来权衡 4 项损失的比重. Chamfer 距离损失作为最重要的保真项, 度量了预测网格和真实网格的差距, 确保预测网格的顶点向正确的位置变形; 而边长损失、拉普拉斯损失以及法向损失作为正则化损失, 分别起到防止三角

网格中边长过长和出现飞点、发生自交以及保证网格表面光滑的作用.

第 1 项 Chamfer 距离损失为

$$l_{\mathrm{c}}=\sum_{p\in P}\min_{q\in Q}\|p-q\|_2^2+\sum_{q\in Q}\min_{p\in P}\|p-q\|_2^2 \tag{4}$$

第 2 项边长损失引导网络生成边长尽量均匀的网格, 计算公式为

$$l_{\mathrm{e}}=\sum_{p\in P}\sum_{k\in N(p)}\|p-k\|_2^2 .$$

第 3 项拉普拉斯损失计算公式为

$$l_{\mathrm{lap}}=\sum_{p\in P}\left\|\delta_p'-\delta_p\right\|_2^2 .$$

其中, δ_p 和 δ_p' 分别表示变形前后的拉普拉斯坐标, $\delta_p=p-\dfrac{1}{|N(p)|}\sum_{k\in N(p)}k$.

第 4 项法向损失用来获得光滑的表面, 计算公式为 $l_{\mathrm{n}}=\sum_{p\in P}\sum_{k\in N(p)}\left\|\langle p-k,n_q\rangle\right\|_2^2$. 其中, P 表示预测网格的点集; Q 表示对应的真实网格的点集; $N(p)$ 表示顶点 p 的邻域.

4 实验及结果分析

本节介绍实验设置以及实验结果. 大量的实验表明, 本文方法与已有方法相比在特征保持上有了较为明显的提升.

(1) 数据集. 本文采用的 ShapeNet 数据集来自文献[1], 目前大多数的三维建模工作使用的数据集均是此数据集. 其包含 13 个类别, 总共有 5 万个三维 CAD 模型. 数据集中还包含模型的渲染图像、相机的内置和外置参数等. 受实验设备的限制, 训练需要大量的时间, 分别在其中的 6 个类别上做了实验. 这 6 个类别在现实生活中是常见的, 分别是飞机、长凳、台灯、椅子、汽车和枪支.

(2) 参数设置. 实验中总共跑了 50 个 epoch, 前 40 个 epoch 的学习率是 3×10^{-5}, 后 10 个 epoch 的学习率降为 1×10^{-5}, 使用的是 Adam 优化器, 权重递减率是 1×10^{-5}. 初始椭球体共有 156 个顶点, 462 条边, 最终生成的网格共有 2 466 个顶点.

网络的 3 个网格变形块中均计算了式(3)的 4 项损失, 超参数 λ_1, λ_2, λ_3 的设置中, λ_1 均取 0.1, λ_2 在 3 个网格变形块中分别取 0.03, 0.3, 0.3, λ_3 取 1.6×10^{-4}. 在网格初次变形时允许网格发生大的变形, 而在后面 2 个变形块中加大拉普拉斯项的权重用以防止自交, 保持局部细节.

(3) 基线. 本文方法使用深度神经网络从单幅图像重建物体网格, 故与基于深度学习的单幅图像建模物体的方法[1,3,10,13-15,19-20,30]作为基线进行比较. 文献[1]建模的是体素形式, 文献[3]建模的是点云形式, 文献[10,13]是基于 GCNN 和三维监督的从单幅图像重建网格的方法, 文献[14-15,19]是可以建模非 0 亏格物体的方法, 文献[20]是基于残差网重建物体的方法, 文献[30]是无监督网格生成模型.

(4) 评估标准. 本文使用 F-score 和 Chamfer 距离作为评估的数值指标. F-score 的 F 是在给定阈值下精确率 p 和召回率 r 的调和平均, 即

$$F=\frac{2pr}{p+r}.$$

F 越大, 说明结果越好.

Chamfer 距离 l_{c} 越小说明建模的网格和真实网格越接近, 预测的结果越好.

4.1 消融实验

为了验证特征检测和改进 GCNN 的有效性, 本文进行了消融实验, 与删除了特征检测和改进 GCNN 的版本进行对比, Chamfer 距离结果如表 1 所示. 从表 1 可以看出在 6 个类别上改进的平均 Chamfer 距离更小; 同时还和消融后的版本做了视觉效果的比较, 如图 3 所示.

表 1 消融实验 Chamfer 距离对比

类别	消融版本	本文
飞机	**0.393**	0.399
长凳	**0.703**	0.762
台灯	1.214	**1.141**
椅子	0.625	**0.619**
汽车	0.313	**0.275**
枪支	0.449	**0.429**
均值	0.616	**0.604**

注. 粗体表示最优的结果.

a. 输入图像 b. 消融版本 c. 本文

图 3 消融实验对比图

4.2　定量比较

本文方法与基线方法的定量比较结果如表 2 和表 3 所示. 表 2 是在 2 种不同阈值 τ 和 2τ 下计算的 F，其中 $\tau=10^{-4}$. 由表 2 可知，在阈值为 τ 时，文献[13]方法表现最好，这是因为其采用了非均匀网格和保特征算法，并为计算全局三维特征在网格和体素表示之间多次变换，在这样复杂的处理后，文献[13]得到了 F 指标最优的结果. 在阈值为 τ 时，本文方法除了椅子类指标略低于 P2M[10]方法，其他类别和均值均是次优；在阈值为 2τ 时，除了椅子和长凳类 P2M[10]方法的结果最好，本文方法在飞机、台灯、汽车、枪支等其他类别和均值上数值指标均是最好的.

表 3 记录了预测网格和真实网格间的 Chamfer 距离. 在计算 Chamfer 距离时，除了文献[18]方法采用预测网格上采样 30 000 个点，真实网格上采样 2500 个点(原文处理方法)之外，其他方法均是在预测网格和真实网格上各采样 10000 个点. Chamfer 距离值均乘以 10^3，由表 3 可知，本文方法的 Chamfer 距离指标在大部分类别和均值上优于对比方法.

4.3　视觉比较

本节提供视觉建模结果用于直观比较. 从图 4 可以看出，在保持尖锐特征上，本文方法稍胜一筹.

表 2　ShapeNet 上 6 个不同类别的 *F*-score (阈值 τ 和 2τ)

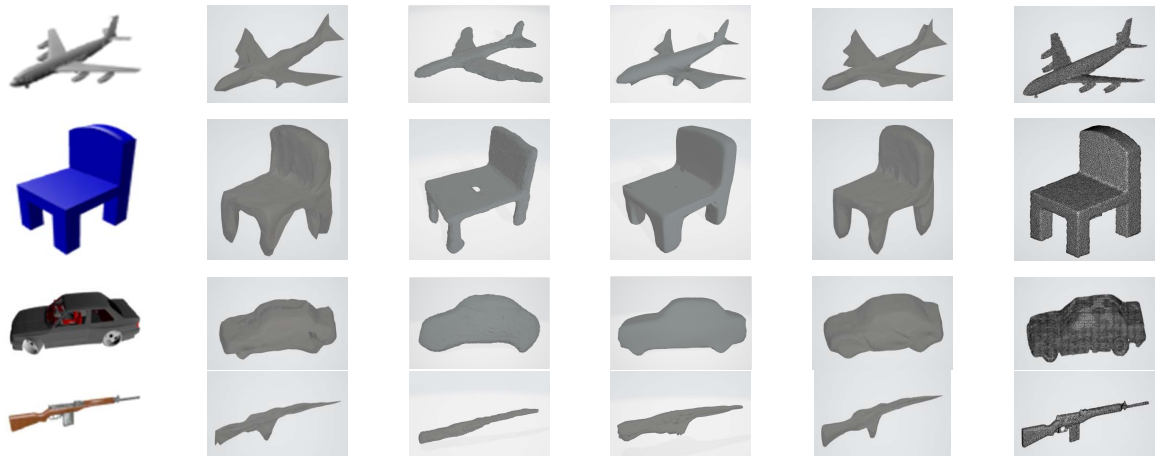
类别	3D-R2N2[1]		PSG[3]		N3MR[30]		P2M[10]		GEOMetrics[13]		本文	
	τ	2τ	τ	2τ	τ	2τ	τ	2τ	τ	2τ	τ	2τ
飞机	41.460	63.230	68.200	81.220	62.100	77.150	71.120	81.380	**89.000**		***74.760***	**83.910**
长凳	34.090	48.890	49.290	69.170	35.840	49.580	57.570	**71.860**	**72.110**		***57.660***	71.540
台灯	32.350	44.370	41.400	58.840	27.970	39.410	48.150	61.500	**58.650**		***50.510***	**64.670**
椅子	40.220	55.200	41.600	63.700	30.250	44.590	***54.380***	**70.420**	**56.610**		53.740	70.030
汽车	37.800	54.840	50.700	77.790	36.660	53.930	67.860	84.150	**74.640**		***69.120***	**84.640**
枪支	28.340	46.870	69.960	82.650	73.200	63.280	73.200	83.470	**88.360**		***76.880***	**85.480**
均值	35.710	52.230	53.530	72.230	44.340	54.660	62.050	75.460	**73.230**		63.780	**76.710**

注. 粗体表示最优的结果；黑斜体表示次优的结果.

表 3　ShapeNet 上 6 个不同类别的 Chamfer 距离

类别	3D-R2N2[1]	PSG[3]	N3MR[30]	AtlasNet[15]	Skeleton[19]	TMN[14]	ResMeshNet[20]	P2M[10]	本文
飞机	10.434	3.824	3.550	1.529	1.364	1.390	1.490	1.890	**0.415**
长凳	10.511	3.504	10.865	2.264	1.639	2.172	2.270	1.774	**0.757**
台灯				3.999	**0.717**		9.460	3.561	1.177
椅子	4.723	2.553	15.891	1.342	1.002	3.064	3.190	1.923	**0.772**
汽车				1.320	3.639		2.560	1.408	**0.284**
枪支	10.176	1.473	3.230	2.276	1.784	1.142	1.390	1.793	**0.442**
均值	8.961	2.839	8.384	2.122	1.691	6.626	3.393	2.058	**0.641**

注. 粗体表示最优的结果.



a. 输入图像　b. 文献[10]　c. 文献[19]　d. 文献[15]　e. 本文　f. 真实网格

图 4　不同方法视觉效果的对比

5 结 语

从单幅图像建模得到物体的三维网格是一项具有重要意义的工作. 目前, 有许多学者对其进行不断的探索和完善, 相信其在实际生活中具有广泛的应用前景. 本文利用 CNN 从图像中提取特征, 再借助改进的 GCNN 对整个网格变形的同时尽量保持物体的特征. 从测试的数值结果以及视觉效果可以看出, 改进方法是有效果的. 但是不可否认的是本文仍存在部分不足. 例如, 无法改变拓扑结构, 如果被建模的物体存在洞, 那么不能精确复原. 由于采用的初始网格是一个水密网格, 并且在变形过程中一直保持网格的拓扑结构不变. 未来将研究任意拓扑的图像三维重建, 研究基于多视角图像的网格重建也是一项值得考虑的工作.

参考文献(References):

[1] Choy C B, Xu D F, Gwak J Y, *et al*. 3D-R2N2: a unified approach for single and multi-view 3D object reconstruction[C] //Proceedings of the 14th European Conference on Computer Vision. Heidelberg: Springer-Verlag, 2016: 628-644

[2] Tatarchenko M, Dosovitskiy A, Brox T. Octree generating networks: efficient convolutional architectures for high-resolution 3D outputs[C] //Proceedings of the IEEE International Conference on Computer Vision. Los Alamitos: IEEE Computer Society Press, 2017: 2107-2115

[3] Fan H Q, Su H, Guibas L. A point set generation network for 3D object reconstruction from a single image[C] //Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2017: 2463-2471

[4] Achlioptas P, Diamanti O, Mitliagkas I, *et al*. Learning representations and generative models for 3D point clouds[C] //Proceedings of the 35th International Conference on Machine Learning. New York: PMLR, 2018: 40-49

[5] Lin C H, Kong C, Lucey S. Learning efficient point cloud generation for dense 3D object reconstruction[C] //Proceedings of the 32nd AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2018: 7114-7121

[6] Gkioxari G, Johnson J, Malik J. Mesh R-CNN[C] //Proceedings of the 17th IEEE/CVF International Conference on Computer Vision. Los Alamitos: IEEE Computer Society Press, 2019: 9784-9794

[7] Lorensen W E, Cline H E. Marching cubes: a high resolution 3D surface construction algorithm[C] //Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques. New York: Association for Computing Machinery, 1987: 163-169

[8] Hanocka R, Metzer G, Giryes R, *et al*. Point2Mesh: a self-prior for deformable meshes[J]. Association for Computing Machinery, 2020, 39(4): Article No.126

[9] Bernardini F, Mittleman J, Rushmeier H, *et al*. The ball-pivoting algorithm for surface reconstruction[J]. IEEE Transactions on Visualization and Computer Graphics, 1999, 5(4): 349-359

[10] Wang N Y, Zhang Y D, Li Z W, *et al*. Pixel2Mesh: generating 3D mesh models from single RGB images[C] //Proceedings of the 15th European Conference on Computer Vision. Heidelberg: Springer-Verlag, 2018: 55-71

[11] Leonard J J, Durrant-Whyte H F. Mobile robot localization by tracking geometric beacons[J]. IEEE Transactions on Robotics and Automation, 1991, 7(3): 376-382

[12] Schönberger J L, Frahm J M. Structure-from-motion revisited[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2016: 4104-4113

[13] Smith E J, Fujimoto S, Romero A, *et al*. GEOMetrics: exploiting geometric structure for graph-encoded objects[C] //Proceedings of the 36th International Conference on Machine Learning. New York: JMLR, 2019

[14] Pan J Y, Han X G, Chen W K, *et al*. Deep mesh reconstruction from single RGB images via topology modification networks[C] //Proceedings of the IEEE/CVF International Conference on Computer Vision. Los Alamitos: IEEE Computer Society Press, 2019: 9963-9972

[15] Groueix T, Fisher M, Kim V G, *et al*. AtlasNet: a papier-Mache approach to learning 3D surface generation[C] //Proceedings of the 31st IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2018: 216-224

[16] Zhou Y, Hu L W, Xing J, *et al*. HairNet: single-view hair reconstruction using convolutional neural networks[C]// Proceedings of the 15th European Conference on Computer Vision. Heidelberg: Springer-Verlag, 2018: 249-265

[17] Yan X C, Yang J M, Yumer E, *et al*. Perspective transformer nets: learning single-view 3D object reconstruction without 3D supervision[C] //Proceedings of the 30th International Conference on Neural Information Processing Systems. New York: ACM Press, 2016: 1704-1712

[18] Pontes J K, Kong C, Sridharan S, *et al*. Image2Mesh: a learning framework for single image 3D reconstruction[C]//Proceedings of the 14th Asian Conference on Computer Vision. Heidelberg: Springer-Verlag, 2018: 365-381

[19] Tang J P, Han X G, Pan J Y, *et al*. A skeleton-bridged deep learning approach for generating meshes of complex topologies from single RGB images[C] //Proceedings of the 32nd IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2019: 4536-4545

[20] Pan J Y, Li J, Han X G, *et al*. Residual MeshNet: learning to deform meshes for single-view 3D reconstruction[C] //Proceedings of the 6th International Conference on 3D Vision. Los Alamitos: IEEE Computer Society Press, 2018: 719-727

[21] Wen C, Zhang Y D, Li Z W, *et al*. Pixel2Mesh++: multi-view 3D mesh generation via deformation[C] //Proceedings of the 17th IEEE/CVF International Conference on Computer Vision. Los Alamitos: IEEE Computer Society Press, 2019: 1042-1051

[22] Huang Z, Li T Y, Chen W K, *et al*. Deep volumetric video from very sparse multi-view performance capture[C] //Proceedings of the 15th European Conference on Computer Vision. Heidelberg: Springer-Verlag, 2018: 351-369

[23] Sun Yankui, Tan Yuxi, Ding Chen, *et al*. Efficient surface reconstruction via Helmholtz reciprocity with two image Pairs[J]. Journal of Computer-Aided Design & Computer Graphics, 2009, 21(10): 1433-1437(in Chinese)
(孙延奎, 谭玉玺, 丁辰, 等. 利用 Helmholtz 互易原理由两对图像重建物体三维表面[J]. 计算机辅助设计与图形学学报, 2009, 21(10): 1433-1437)

[24] Chen Xuegong, Huang Wei, Ji Xing, *et al*. Solution to surface reconstruction from contours[J]. Computer Engineering and Applications, 2011, 47(14): 157-159+163(in Chinese)
(陈学工, 黄伟, 季兴, 等. 一种由轮廓线重建物体表面的方法[J]. 计算机工程与应用, 2011, 47(14): 157-159+163)

[25] Zhang Yan, Shen Jingling, Zhang Cunlin. An algorithm for reconstructing objects using multiple holograms[C] //Proceedings of the 2005 Annual Meeting of Holographic and Optical Information Processing Committee of Chinese Optical Society and 20th Anniversary of Construction. Beijing: Holographic and Optical Information Processing Committee of China Optical Society, 2005: 177-182(in Chinese)
(张岩, 沈京玲, 张存林. 利用多幅全息图重建物体的算法[C] //2005 年中国光学学会全息与光学信息处理专业委员会年会暨建会 20 周年纪念会论文集. 北京: 中国光学学会全息与光学信息处理专业委员会, 2005: 177-182)

[26] Nie Y Y, Han X G, Guo S H, *et al*. Total3DUnderstanding: joint layout, object pose and mesh reconstruction for indoor scenes from a single image[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2020: 52-61

[27] Liu Xin. GPU based large-scene 3D reconstruction and object modeling[D]. Beijing: University of the Chinese Academy of Sciences, 2012(in Chinese)
(刘鑫. 基于 GPU 的大场景三维重建和物体建模[D]. 北京: 中国科学院大学, 2012)

[28] Bronstein M M, Bruna J, LeCun Y, *et al*. Geometric deep learning: going beyond euclidean data[J]. IEEE Signal Processing Magazine, 2017, 34(4): 18-42

[29] He K M, Zhang X Y, Ren S Q, *et al*. Deep residual learning for image recognition[C] //Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2016: 770-778

[30] Kato H, Ushiku Y, Harada T. Neural 3D mesh renderer[C] //Proceedings of the 31th IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2018: 3907-3916