基于卷积神经网络和投票机制的三维模型分类与检索

白 静[1)], 司庆龙[1)], 秦飞巍[2)]

[1)] (北方民族大学计算机科学与工程学院 银川 750021)
[2)] (杭州电子科技大学计算机学院 杭州 310018)
(baijing_nun@163.com)

摘 要: 针对现有基于深度学习的三维模型多视图分类算法利用最大池化、平均池化等像素级运算完成视图信息的融合, 可能造成模型有益信息淹没和混淆的问题, 提出一种基于卷积神经网络和投票机制的三维模型分类检索算法. 首先将三维模型转化为一组二维视图, 然后基于丰富的数字图像库 ImageNet 和成熟的图像深度学习模型 CaffeNet 完成二维视图的分类, 最后利用加权投票的方式完成三维模型的分类; 同时基于投票机制, 提出 4 种三维模型距离度量算法, 支持三维模型的检索. 将文中算法应用于刚性三维模型库 ModelNet10, ModelNet40, 非刚性三维模型库 SHREC10, SHREC11 和 SHREC15 中, 分类准确率分别为 93.18%, 93.07%, 99.5%, 99.5%和 99.4%, 检索性能突出; 并通过实验验证该算法的有效性.

关键词: 三维模型检索; 卷积神经网络; 投票机制; 深度学习; 非刚性三维模型
中图法分类号: TP391.41 **DOI:** 10.3724/SP.J.1089.2019.17160

3D Model Classification and Retrieval Based on CNN and Voting Scheme

Bai Jing[1)], Si Qinglong[1)], and Qin Feiwei[2)]

[1)] (*School of Computer Science and Engineering, North Minzu University, Yinchuan* 750021)
[2)] (*School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou* 310018)

Abstract: The existing deep learning algorithms for view-based 3D model classification use pixel-level operations, such as maximum pooling and average pooling, to fuse the views' information, which may lose or overwrite the useful information of 3D models. Aiming at the problem, a 3D model classification and retrieval algorithm based on convolutional neural network and voting scheme is proposed. Firstly, each 3D model is converted to a set of 2D views. Then, those 2D views are classified based on deep learning model CaffeNet with rich digital image library ImageNet. Finally, the 3D model is classified by weighted voting. Furthermore, based on the voting scheme, four distance measurement algorithms are proposed to retrieve 3D model. Experiments on the rigid 3D model libraries ModelNet10, ModelNet40, and the non-rigid 3D model libraries SHREC10, SHREC11 and SHREC15 demonstrate the effectiveness of the proposed algorithm. The classification accuracy rates for above five libraries are 93.18%, 93.07%, 99.5%, 99.5% and 99.4% respectively, and the retrieval performance is on par or better than state-of-the-art methods.

Key words: 3D model retrieval; convolutional neural network; voting scheme; deep learning; non-rigid 3D models

---

收稿日期: 2018-03-02; 修回日期: 2018-11-12. **基金项目**: 国家自然科学基金(61762003, 61502129); 宁夏自然科学基金(2018AAC03124); 宁夏高等学校一流学科建设(电子科学与技术: NXYLXK2017A07); 国家民族事务委员会"图像与智能信息处理创新团队"; 国家民族事务委员会中青年英才计划(2016GQR08); 浙江省自然科学基金(LQ16F020004). **白 静**(1982—), 女, 博士, 副教授, 硕士生导师, CCF 高级会员, 主要研究方向为 CAD&CG、机器学习; **司庆龙**(1986—), 男, 硕士研究生, 主要研究方向为机器学习; **秦飞巍**(1985—), 男, 博士, 副教授, CCF 会员, 主要研究方向为 CAD&CG、机器学习.

随着三维建模技术、成像技术及计算机视觉的不断发展, 三维模型的检索工作受到学者及业界的广泛关注. 传统的三维模型检索算法利用人工设计三维特征, 其表征的优劣取决于设计者对模型或检索需求的分析及理解, 而非模型或检索需求本身. 因此, 该类算法往往需要针对一个新的问题或新的数据集设计新的特征表示方式, 制约了三维模型检索技术的发展. 深度学习能让机器自动学习客观对象的多层抽象和表示, 从而理解各类复杂对象的内在含义, 其近年来在语音识别、计算机视觉等多类应用中取得了突破性进展, 也得到了三维模型检索领域学者的关注.

整体来看, 基于深度学习的三维模型分类算法可以分为 3 大类: 基于低层次特征的、基于体素模型的以及基于多视图表征的. 由于深度学习在图像领域应用相对成熟, 基于多视图表征的算法可以更好地利用图像领域优秀的深度学习模型, 获得不错的分类效果. 同时, 利用视图可以非常方便地建立三维模型同图像、草图等其他模态数据间的对应关系, 对三维模型多模态数据间的互检索具有积极的推进作用, 值得进一步研究与探索.

基于深度学习的三维模型多视图分类算法首先将三维模型转化为一系列二维图像, 并利用面向二维图像的深度学习算法获取视图特征, 再利用最大池化、平均池化等像素级运算完成视图信息的融合, 形成三维模型的最终表征及分类. 由于三维模型的多个视图是该物体不同视角下的几何投影, 具有相对独立性, 这种像素级的融合运算并无直接的物理或几何意义, 可能造成图像有益信息的淹没和混淆. 针对这一问题, 本文提出一种基于卷积神经网络(convolutional neural network, CNN)和加权投票的三维模型分类算法. 该算法首先利用多视图表征原始三维模型; 其次利用 CNN 完成基于视图的初步识别; 最后通过决策层的加权投票完成三维模型的最终分类, 以避免像素级的视图融合, 突出多数有效视角, 减小少数不佳视角干扰, 进而提高视图融合算法的可解释性及以此为基础的三维模型分类能力.

1 相关工作

三维模型的典型表征形式有 B-rep 表示、三维网格模型、点云模型. 这 3 种表征模型均为非结构化的, 不能直接作为深度学习的输入模型. 因此, 面向三维模型检索的深度学习算法首先要解决的问题就是将三维模型转化为深度学习可接受的结构化表征. 具体地, 按照结构化表征方式的不同, 面向三维形状检索的深度学习算法可以分为 3 类.

(1) 基于低层次特征的深度学习算法. 这类算法提取三维模型的低层次特征并将其表示为一组多维向量, 如 Zernike 矩[1]和热核特征[2]; 然后以低层次特征集合为输入, 构造深度学习模型, 完成高层次特征提取和模型分类. 深度学习算法的典型优点就在于它能够根据被表征对象和分类需求的不同, 自动学习并构建客观对象的特征; 而这类算法在提取低层次特征时已经完成了一次模型抽象及表示, 未能充分利用深度学习算法的特点, 会在一定程度上丢失模型信息并干扰深度学习效果.

(2) 基于体素模型的深度学习算法. 这类算法通过体素化将三维物体表征为一个二值或实值的三维张量, 并以此为输入, 构建各种三维 CNN[3-8], 实现特征的自动提取和三维模型的分类, 已取得了非常好的分类效果. 只是这类表征方式存在高维、稀疏的特点, 在一定程度上影响了对应网络的分类性能. 为此, 有学者在体素表征和深度学习中引入了多分辨率的思想[9-10], 分类效果有所改善. 此外, 在引入方向信息的情况下, Sedaghat 等[6]构造的深度学习分类器在 ModelNet10 上的分类准确率高达 93.8%; 不足之处就在于该算法需要额外构建模型的方向信息, 数据集构建代价过大. Brock 等[11]则提出了将变分自编码器和 CNN 相结合的思想, 构建了基于变分自编码器的三维模型分类算法, 其在 ModelNet10 中的分类精度高达 97.14%, 只是耗时较长.

(3) 基于视图的深度学习算法. 这类算法通过投影的方式将三维模型表征为一组二维视图的集合, 并以此为基础构建深度学习模型完成特征学习及模型分类. 典型工作有基于全景视图的 DeepPano 算法[12], 基于几何图像的 Geometry Image[13] 算法, 基于多视图的 Su-MVCNN[14], Wang-MVCNN[15], VS-MVCNN[16], 基于成对图像的 Pairwise 算法[17], 以及华中科大研究人员所提出的利用 GPU 及倒排文件加速分类的实时三维物体识别算法[18]. 这类算法能够一定程度地保留三维形状的原始信息, 同时充分利用二维图像领域的海量数据库及性能优越的深度学习模型, 整体效果不错.

2　本文算法

大量算法说明, 基于视图的深度学习算法能够很好地适用于三维模型的分类任务. 其中, 多视图的表征要优于单视图的表征. 本文试图以多视图深度学习算法为研究对象, 通过加权投票实现多视图的有效融合, 具有更好的可解释性, 可改善三维模型的分类和检索效果. 本文算法框架如图 1 所示, 首先获取三维模型的多视图表征, 然后针对单个视图构建基于 CNN 的弱分类器完成视图分类及表征, 最后利用加权投票策略构建强分类器完成模型分类. 这种基于加权投票的分类思想既确保了三维物体各个视图间的相对独立性, 又避免个别有歧义视图对物体类别判断的误导, 且满足人类的普遍认知“如果从多个不同的角度去看 2 个物体是相似的, 那么这 2 个物体是相似的”.

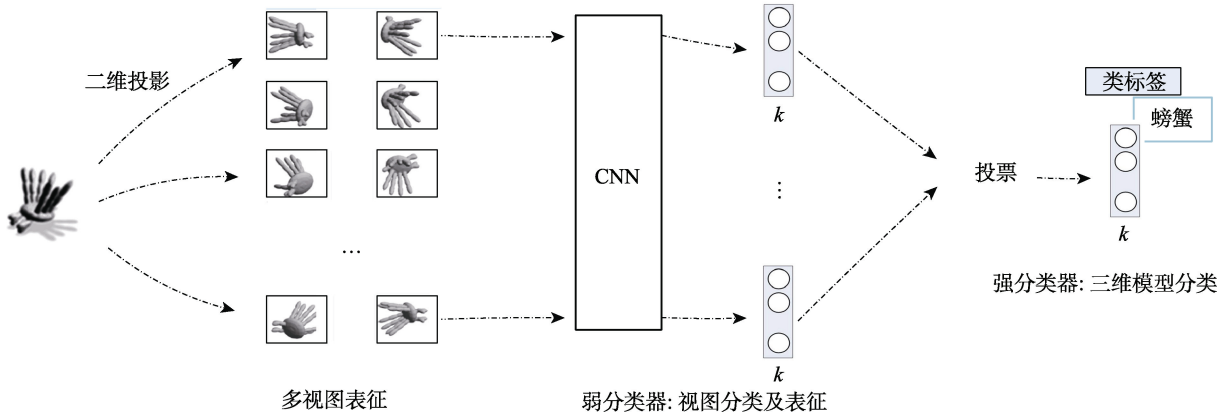


图 1　本文算法框架

2.1　多视图表征

三维模型的视图获取方式多种多样. 综合比较这些方式及其对应的分类结果可知, Su-MVCNN 所提出的 12 视图渲染方式是一种综合表现优秀的视图获取方式[14]. 因此, 本文中, 沿用该方式构建给定网格模型 M 的多视图表征 $V(M)=\{v_l, 1\leqslant l\leqslant n\}$; 其中, n 为视图数目.

以 12 个视图为例, 图 2 所示为三维模型渲染示意图.

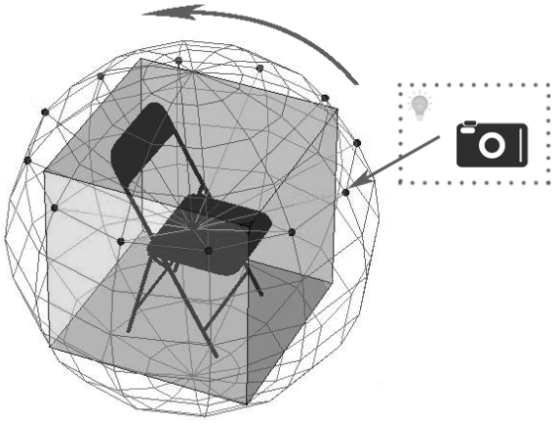


图 2　三维模型的渲染示意图

Step1. 模型预处理. 首先, 缩放并平移模型, 使得模型被限定在单位球内接立方体内部. 如图 2 所示, 球内接立方体构成模型的 AABB 包围盒. 通过此步可以将不同大小和不同位置的模型归一化至相同尺寸、相同位置.

Step2. 视点的确定. 如图 2 圆周上黑点所示, 在单位球斜向上 30°圆周上水平、均匀地放置 12 个虚拟摄像机, 并确保摄像机镜头垂直于球心同摄像机之间的连线.

Step3. 视图的渲染. 为获取更接近真实模型的视图信息, 在摄像机右上方放置一光源, 并利用冯氏光照模型和高洛德着色方案完成视图渲染[19]. 具体地, 本文采用 MATLAB 工具内置的三角网格绘制函数 trimesh 来完成视图渲染. 渲染时, 设置函数属性 *AmbientStrength* = 0.2, *DiffuseStrength* = 0.6, *SpecularStrength* = 0.0; 3 个属性分别对应冯氏光照模型中的环境光照、漫反射光照和镜面光照; 设置函数属性 *FaceLighting* 为 Gouraud, 即采用 Gouraud 方案对三角面着色.

至此, 给定三维网格模型 M 和视图数目 n, 可以通过以上 3 步获得其多视图表征 $V(M)=\{v_l, 1\leqslant l\leqslant n\}$. 如图 2 所示, 由于这 n 个视角均匀的位于三维模型不同视点, 相互之间具有较强的互补性和较低的相关性, 因而以此为基础获得的多视图表征构成了三维模型较为完整的描述.

2.2　弱分类器: 视图分类及表征

由于视图集 $V(M)$中的视图是三维物体不同视角下的几何投影, 具有相对独立性, 视图间最大池化、平均池化等像素级的算术运算和逻辑运算并无直接意义, 甚至可能造成图像有益信息的淹没和混淆. 因此, 如图 1 所示, 本文不对视图信息进行

任何层面的合成，而是首先构造弱分类器完成基于单视图的分类，然后在此基础上综合形成基于加权投票的强分类器完成三维模型的最终分类. 考虑到三维模型不同视角下的视图是独立对等的，本文面向各个视图的弱分类器间彼此共享参数，以进一步简化网络结构、减少网络参数、提高网络训练速度、增加网络的鲁棒性. 下面将讨论如何基于 M 的单个独立视图 v_l, $1\leqslant l\leqslant n$，构建弱分类器，完成视图分类及表征.

在图像识别领域，存在大量优秀的深度学习模型. 考虑到三维模型库的规模及视图的复杂性，本文选择 Jia 等[20]提出的 CaffeNet 作为面向单个二维视图分类及表征的深度学习模型. 如图 3 所示，该网络共包含 8 层，其中前 5 层为卷积层，中间 2 层为全连接层，最后 1 层为网络输出层 FC8 和 Softmax 分类层.

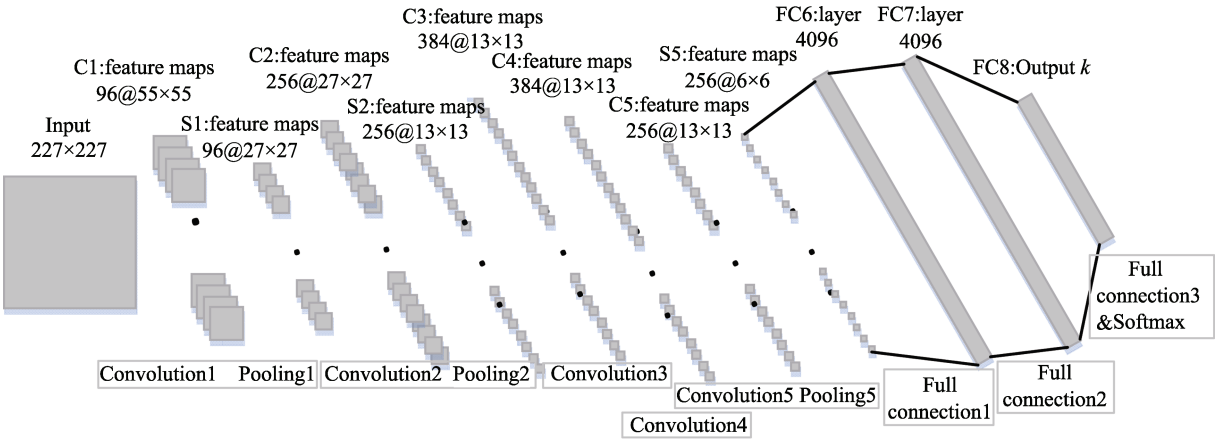


图 3 弱分类器 CaffeNet 的网络结构

CaffeNet 的训练分为 2 步：首先利用 ImageNet 丰富的图像资源[21]作为输入，对 CaffeNet 进行预训练；然后采用三维模型渲染的二维视图作为输入对获得的网络模型进行微调，使最终获得的网络模型有效适应三维模型对应的二维视图. 训练好该网络模型后，给定三维模型 M 的单个独立视图 v_l, $1\leqslant l\leqslant n$，以 FC8 层的输出 $D_l=\{d_{il}, 1\leqslant i\leqslant k\}$ 作为该视图的特征描述，其中 k 为类的数目；以 Softmax 层输出结果 $P_l=\{p_{il}, p_{il}=P(\omega_i\,|\,v_l), 1\leqslant i\leqslant k\}$ 作为其分类结果，即视图属于各个类的概率，其中，ω_i 表示类 i, k 为类的数目.

2.3 强分类器：基于投票的三维模型分类

给定三维模型 M 多视图表征中每个视图属于各个类的概率，本节旨在以其为输入，构建强分类器，完成三维模型的分类. 设依据视图属于每个类的概率分布情况，计算类 i 所获视图投票值为 $T_{\text{vote}}(\omega_i\,|\,M)$, k 为类的数目，则模型 M 的分类结果为

$$u=\arg\max_{i=1}^{k} T_{\text{vote}}(\omega_i\,|\,M)\rightarrow M\in\omega_u .$$

本文中，依据不同原理提供 3 种投票算法.

(1) 概率投票法. 将三维模型各个视图 v_l 属于某个类 i 的概率 p_{il} 作为该视图对相应类的投票值，累加计算类 ω_i 所获投票，设 n 为单个模型视图数目，则有 $T_{\text{vote}}(\omega_i\,|\,M)=\sum_{l=1}^{n} p_{il}$.

(2) 0-1 投票法. 对每一个视图 v_l 而言，依据概率 p_{il} 来看，三维模型最可能属于哪个类，就给哪个类投出一票；若三维模型以相近的概率属于多个类，则给这几个类都投出一票.

令 $L_k(\boldsymbol{P}_l)$ 表示视图 v_l 属于各类概率 $\boldsymbol{P}_l$ 中的第 k 大元素，则当视图 v_l 属于某个类 ω_i 的概率最大时，对该类投票，有 $T_{\text{vote}}(\omega_i\,|\,v_l)=1$；如果 v_l 属于某个类 ω_i 的概率与最大概率值接近时，该视图 v_l 同样可能属于类 ω_i，因此也对该类投出一票，则有 $T_{\text{vote}}(\omega_i\,|\,v_l)=1$；否则 $T_{\text{vote}}(\omega_i\,|\,v_l)=0$. 具体计算方法为

$$T_{\text{vote}}(\omega_i\,|\,v_l)=\begin{cases}1, & \begin{matrix}P_{il}=L_k(\boldsymbol{P}_l)\\ \text{且}L_1(\boldsymbol{P}_l)-L_k(\boldsymbol{P}_l)\leqslant\theta\end{matrix}\\ 0, & \begin{matrix}P_{il}=L_k(\boldsymbol{P}_l)\\ \text{且}L_1(\boldsymbol{P}_l)-L_k(\boldsymbol{P}_l)>\theta\end{matrix}\end{cases} \tag{1}$$

其中，θ 为阈值，取值区间为 $[0\ 1)$: $\theta=0$ 时，各个视图仅可以对概率最大的类投一票；$\theta>0$ 时，允许视图对更多接近最大概率值的类投票. 即，θ 取值较小时，只有一个或多个最为相似的类获得投票，限定严格，更多相近的类可能被忽略；θ 取值较大时，更多的类获得投票，考虑全面，但是投票结果

的可信度也有所降低. 具体取值可根据应用领域和实验要求选择.

得到每个视图 v_l 对类 ω_i 的投票结果 $T_{\text{vote}}(\omega_i \mid v_l)$ 后, 可统计对应三维模型 M 对类 ω_i 的投票结果为 $T_{\text{vote}}(\omega_i \mid M)=\sum_{l=1}^{n} T_{\text{vote}}(\omega_i \mid v_l),\ v_l \in \boldsymbol{V}(M)$.

(3) 加权 0-1 投票法. 在方法 2 中, 根据 θ 选取的不同, 有些视图可能投了多票, 有些视图可能投了一票, 不具有公平性. 为此, 在式(1)的基础上, 提出加权 0-1 投票法, 具体计算方法为

$$\begin{cases} T_{\text{vote}}(\omega_i \mid M)=\sum_{l=1}^{n} \Im_l T_{\text{vote}}(\omega_i \mid v_l),\ v_l \in \boldsymbol{V}(M) \\ \Im_l=\dfrac{1}{\sum_{i=1}^{n} T_{\text{vote}}(\omega_i \mid v_l)} \end{cases}.$$

其中, 视图 v_l 的权重为其投票数目的倒数, 以确保各个视图在投票中的相对平等地位.

以上 3 种投票法出发点有所不同, 实验中将进一步分析比较.

2.4　基于多特征的三维模型检索

三维模型的检索依赖模型间的相似度计算或距离度量. 本节将基于三维模型的视图表征及投票机制给出距离度量算法.

输入模型 x, y, 为保持对称性, 它们之间的距离 $d(x,y)$ 可以转换为模型 x 到 y 距离和模型 y 到 x 距离的平均值, 即 $d(x,y)=\dfrac{d(x\to y)+d(y\to x)}{2}$.

基于视图表示, 模型 x 到 y 距离 $d(x\to y)$ 又可以转换为它们所对应视图集 $V(x)$ 到 $V(y)$ 的距离. 本文选取 FC8 层的输出结果 $D_l=\{d_{il},\ 1\leqslant i\leqslant k\}$ 作为视图 v_l 的描述符. 相应地, 令 x_i 为模型 x 的第 i 个视图的描述符, y_j 为模型 y 的第 j 个视图的描述符, $1\leqslant i, j\leqslant n$, n 为视图数目. 本文提出 4 种不同的距离度量方式, 描述如下:

(1) 用视图集 $V(x)$ 内每个视图到 $V(y)$ 内所有视图的最短距离的平均值作为模型 x 到 y 之间的距离, 简称 A2A, 计算公式为

$$d(x\to y)=\frac{\sum_{i=1}^{n}\min_{j} d(x_i,y_j)}{n},\ 1\leqslant i, j\leqslant n.$$

(2) 用视图集 $V(x)$ 及 $V(y)$ 内所有视图间距离的最小距离作为模型 x 到 y 之间的距离, 简称 MinA2A, 计算公式为

$$d(x\to y)=\min_{i,j} d(x_i,y_j),\ 1\leqslant i, j\leqslant n.$$

其中, $d(x\to y)=d(y\to x)$, 故求其中一侧即可.

(3) 只考虑投票正确的视图间的距离, 用视图集 $V(x)$ 内投票正确的视图同视图集 $V(y)$ 内投票正确的视图间距离的平均值作为模型 x 到 y 之间的距离, 简称 V2V, 计算公式为

$$d(x\to y)=\frac{\sum_{x_i\in S(x)}\min_{y_j\in S(y)} d(x_i,y_j)}{|S(x)|},\ 1\leqslant i, j\leqslant n.$$

其中, $S(x)$ 为模型 x 投票正确的视图所对应的描述符集合; $|S|$ 为集合 S 的模.

(4) 只考虑投票正确的视图间的距离, 用视图集 $V(x)$ 内投票正确的视图同视图集 $V(y)$ 内投票正确的视图间距离的平均值作为模型 x 到 y 之间的距离, 简称 V2V, 计算公式为

$$d(x\to y)=\frac{\sum_{x_i\in S(x)}\min_{y_j\in S(y)} d(x_i,y_j)}{|S(x)|},\ 1\leqslant i, j\leqslant n.$$

同样地, 可以通过对称置换计算模型 y 到 x 距离 $d(y\to x)$. 其中, $d(x_i,y_j)$ 表示视图描述符 x_i 同 y_i 的距离, 可根据需要选择欧氏距离、余弦距离等不同的距离度量.

3　实验及结果分析

实验基于 Caffe 框架, 分别使用 CaffeNet 完成图像特征提取及分类, Python 处理相关数据, MATLAB 完成检索指标评价及部分可视化效果, MySQL 存储数据集特征并完成检索, 在 intel i7 2600k + gtx 1060 的 PC 机上测试完成. 下面将分别从分类和检索 2 个方面综合测评本文算法的性能.

3.1　分类实验

分类实验中, 分别选取刚性三维模型数据集 Princeton ModelNet10, ModelNet40[22], 非刚性三维模型数据集 SHREC10[23], SHREC11[24], SHREC15[25]对算法进行测试. 详情如表 1 所示.

表 1　本文选用数据集的基本信息

数据集	模型数目	类的数目	各类内模型数
ModelNet10	4899	10	平均 490
ModelNet40	12311	40	平均 308
SHREC10	200	10	20
SHREC11	600	30	20
SHREC15	1200	50	24

(1) ModelNet10, ModelNet40. CAD模型库, 采用官网数据, 针对 ModelNet10 和 ModelNet40 分别选取 3 991 和 9 843 个模型作为训练数据, 908 和 2 468 个模型作为测试数据;

(2) SHREC10, SHREC11, SHREC15. 非刚性三维模型库, 包含不同姿态的猫、狗等. 官网未给定训练数据和测试数据. 本文以类内 2/8 的比例随机划分测试样本和训练样本, 先后完成 5 次随机实验, 并以 5 次实验的平均准确率作为最终实验结果.

3.1.1 视图及投票方式选取

3.1.1.1 视图的选取

为了探究不同视图选取方式对分类结果的影响, 分别在图 2 所示的圆周上均匀设置 3, 6, 12 和 24 个虚拟摄像机; 由此得到 3 视图, 6 视图, 12 视图和 24 视图的三维模型多视图表征, 分别记为 3V, 6V, 12V, 24V. 此外, 为模拟在球面上均匀放置摄像机获取物体的多视图表征, 本文沿用 Su-MVCNN 所提出的 80 视图构造方式[14]: 在紧致包围物体的正 12 面体的每个顶点上放置 4 个不同角度摄像机, 形成三维模型的 80 视图表征, 记为 80V. 为了避免参数 θ 对分类结果的影响, 实验中选取概率投票法, 以三维模型的多视图为输入完成三维模型的分类, 结果如表 2 所示.

表 2 不同视图选取方式下的分类准确率 %

数据集	3V	6V	12V	24V	80V
ModelNet10	92.30	92.41	**92.85**	92.40	92.73
ModelNet40	91.25	91.86	91.78	92.99	**93.07**
SHREC10	99.00	**99.50**	**99.50**	**99.50**	**99.50**
SHREC11	96.30	98.30	99.50	98.70	**99.60**
SHREC15	97.40	99.00	99.40	98.80	**99.60**

由表 2 可见: (1) 整体来看, 12V 和 80V 取得了最好的分类效果. 在非刚性三维模型数据集上, 这 2 种方法相当. 就刚性三维模型而言, 在模型数目较少的 ModelNet10 上, 12V 分类效果优于 80V; 在模型数目较多的 ModelNet40 上, 80V 分类效果优于 12V. 由于 80V 的视图数目是 12V 的 6 倍以上, 其模型训练时间和测试时间都远远高于 12V, 因此本文中优先推荐 12V 方法. (2) 在同类型的方法中, 分类精度和视图数目并不成正比, 比较 3V, 6V, 12V 和 24V, 在 ModelNet10 上 12V 的分类效果最好, 在 ModelNet40 上 24V 的分类效果最好. 这是因为相比于数据集的规模, 过少的视图数目难以提供足够的视觉信息, 而过多的视图数目可能造成大量的冗余信息进而影响分类决策. 基于表 2 实验数据, 本文的后期实验, 如无特殊说明, 针对所有数据集, 视图数目统一为 12.

3.1.1.2 投票方式选取

本实验旨在比较投票方式及参数 θ 的选取对分类结果的影响. 如式(1)所示, 在“0-1 投票法”和“加权 0-1 投票法”中, 阈值 θ 决定了单个视图的投票数目: 当 $\theta=0$ 时, 每张视图仅投一票; 否则, 可投 1 或多票. 为此, 实验中, 以 12 视图为输入, 针对“0-1 投票法”和“加权 0-1 投票法”, 给出了 θ 在区间 $[0\ 0.9]$ 内变动时的一组分类结果, 而针对“概率投票法”则只给出了一个分类结果, 结果如表 3 所示.

由表 3 可见: (1) “概率投票法”不需要设定任何参数, 最为鲁棒, 分类性能与其他 2 种方法相当. (2) 无论选用哪种投票方式, 参数 θ 如何选取, 非刚性三维模型集 SHREC10, SHREC11, SHREC15 上的分类精度均无改变, 这从侧面反映了非刚性三维模型各个视角的特征信息都较丰富, 因此每个视图都具有较好的区分性. (3) 针对刚性三维模型集, 3 种投票方式在的分类精度有所区别, 但变化不大. “加权 0-1 投票法”在 ModelNet10 和 ModelNet40 取得最佳分类精度, 其中 ModelNet10 在 θ=0.7 时获得最高准确度 93.18%, ModelNet40 在 $\theta=0.5$ 时获得最高准确度 91.82%. (4) 针对非刚性三维模型, 相同的参数 θ, “加权 0-1 投票法”优于“0-1 投票法”, 这也证明了权值的加入避免了单个视图多次投票的问题, 分类效果更优. 基于表 3, 本文的后期实验如无特殊说明, 针对 ModelNet10, ModelNet40, SHREC10, SHREC11, SHREC15, 均采用“加权 0-1 投票法”, θ 取值分别为 0.7, 0.5, 0, 0, 0.

3.1.2 刚性三维模型分类实验对比

3.1.2.1 基于视图的分类算法对比

该实验旨在比较本文算法和其他基于视图的分类算法在刚性三维模型数据集 ModelNet10, ModelNet40 上的分类结果. 表 4 中, 其他算法的结果均来自 Princeton ModelNet 官方网站; 同时, 为保证公平性, 算法 Wang-MVCNN[15]的实验数据以 RGB 视图渲染方式为基准, 与其他多视图算法的视图渲染方式保持一致.

如表 4 所示: (1) 整体来看, 基于多视图表征的分类算法要优于单视图表征的, 这是因为多视图的表征方式往往可以捕捉到更加丰富的模型信息, 而单视图的表征方式则丢弃了过多的特征信息. (2)

表 3　不同投票方式及参数 θ 下的分类准确率　%

数据集	算法	θ					
		0	0.15	0.3	0.5	0.7	0.9
ModelNet10	概率投票法	92.85 (与 θ 无关)					
	0-1 投票法	92.96	92.96	92.74	92.85	92.96	93.18
	加权 0-1 投票法	92.96	92.74	92.96	92.96	**93.18**	92.96
ModelNet40	概率投票法	91.78 (与 θ 无关)					
	0-1 投票法	91.45	91.45	91.54	91.66	91.21	90.44
	加权 0-1 投票法	91.45	91.70	91.66	**91.82**	91.70	91.78
SHREC10	概率投票法	99.50 (与 θ 无关)					
	0-1 投票法	99.50	99.50	99.50	99.50	99.50	99.50
	加权 0-1 投票法	99.50	99.50	99.50	99.50	99.50	99.50
SHREC11	概率投票法	99.50 (与 θ 无关)					
	0-1 投票法	99.50	99.50	99.50	99.50	99.50	99.50
	加权 0-1 投票法	99.50	99.50	99.50	99.50	99.50	99.50
SHREC15	概率投票法	99.40 (与 θ 无关)					
	0-1 投票法	99.40	99.40	99.40	99.40	99.40	99.40
	加权 0-1 投票法	99.40	99.40	99.40	99.40	99.40	99.40

表 4　各种基于视图的分类算法分类准确率　%

算法	视图数	ModelNet10	ModelNet40
DeepPano[12]	1	88.66	82.54
Geometry Image[13]	1	88.40	83.90
CNN[14]	1		85.10
Pairwise[17]	12	93.20	91.10
FusionNet[26]	60	93.11	90.80
Qi-MVCNN[27]	20		91.40
Su-MVCNN[14]	12		89.90
Wang-MVCNN[15]	12		92.20
VS-MVCNN[16]	80	**93.50**	90.90
本文-12	12	93.18	91.82
本文-24	24	92.40	93.03
本文-80	80	92.73	**93.07**

综合来看, 本文算法分类准确率具有一定优势. 在 ModelNet10 数据集上, 其准确率仅次于 VS-MVCNN 算法, 两者精度相差 0.32%; 而在更加复杂的 ModelNet40 数据集上, 其准确率位居第一, 高出 VS-MVCNN 算法 2.17%, 有较强优势. 该实验充分说明了本文算法在刚性三维模型分类中的有效性.

3.1.2.2　与其他分类算法的对比

本实验旨在进一步比较本文算法和基于体素表征分类算法在刚性三维模型数据集 ModelNet10, ModelNet40 上的分类结果. 如表 5 所示, VRN Ensemble 算法在刚性三维模型数据集上均表现最优. 本文算法在 ModelNet10 上表现不如 ORION 算法和 LightNet 算法, 精度分别相差 0.62%和 0.21%; 在 ModelNet40 上的表现仅次于 VRN Ensemble 算法. 实验结果说明了本文算法的有效性, 同时也揭示了本文算法在刚性三维模型的分类中还存在一些缺陷.

表 5　各种基于体素的分类算法分类准确率　%

算法	ModelNet10	ModelNet40
3DShapeNets[3]	83.54	77.32
VoxNet[5]	92.00	83.00
ORION[6]	93.80	
LightNet[28]	93.39	86.90
PointNet[7]	77.60	
VRN Ensemble[11]	**97.14**	**95.54**
本文-12	93.18	91.82
本文-24	92.40	93.03
本文-80	92.73	93.07

3.1.2.3　各个类的分类结果分析

为进一步分析本文算法的特点, 统计了该算法在 ModelNet10 数据集上单个类的分类分布情况. 如图 4 所示, 在 bed, chair, monitor, toilet 这 4 个类上的算法的分类准确率均为 1.00; 在 sofa 和 bathtub 上也较高, 为 0.98; 在 dresser 上次之, 为 0.91; 在 desk, night_stand 和 table 上的分类准确率均低于 0.90, 分别为 0.86, 0.84 和 0.75. 其中, table 类中有 0.25 的模型都错分至 desk 类. 仔细分析 table 和 desk 类内模型, 如图 5 所示, 可以看出这 2 个类类内的测试实例在局部结构上有所区别, 但是整体形状上极其相似. 可以看出, 本文算法在模型的细分类及

局部区域识别方面能力不足，因而无法有效区分这些模型所属类别.

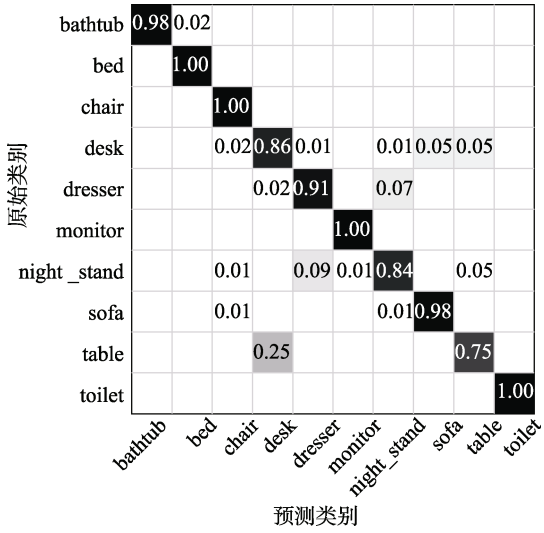


图 4 在 ModelNet10 各个类上的分类结果

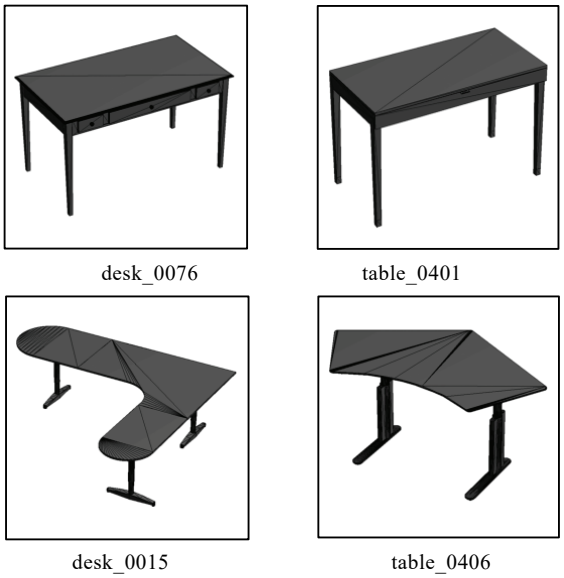


图 5 ModelNet10 内混淆模型举例

3.1.3 非刚性三维模型分类实验对比

本节将给出本文算法和其他算法在非刚性三维模型数据集 SHREC 上的分类结果，并进行对比分析. 表 6 中，其他算法的结果均来自文献[13]; 而 Su-MVCNN[14]作为经典的多视图算法并未在 SHERC 数据集上测试，因此为确保实验对比的有效性，本文基于 Caffe 框架实现了该算法，并针对非刚性三维模型数据集 SHERC10，SHERC11，SHERC15 进行实验. 如表 6 所示，针对非刚性三维模型，基于 CNN 的算法要远好于传统的基于人工提取特征的算法；且本文算法的分类性能要好于基于多视图的 Su-MVCNN 算法[14]和基于体素的 3DShapeNets 算法[3]. 这也说明了本文算法相对独立地考虑三维模型所对应的各个视图，利用加权投票的思想形成最终分类结果，可以有效地利用多视图特征形成决策.

表 6 SHREC 数据集上的分类准确率 %

类型	算法	SHREC 10	SHREC 11	SHREC 15
人工提取特征	LFD[29]		65.80	
	SPM[30]		82.50	
	Conf[31]		85.00	
体素表示&深度学习分类	3DShapeNets[3]		48.40	
多视图表征&深度学习分类	Su-MVCNN[14]	*99.00*	*97.70*	*98.30*
	本文-12	**99.50**	99.50	99.40
	本文-24	**99.50**	**99.60**	**99.60**

3.1.4 分类结果分析与讨论

在针对非刚性三维模型数据集 SHREC 的分类中，本文算法准确率均在 99.40%以上，而针对刚性三维模型数据集 ModelNet 的分类准确率则较低. 通过分析比较发现 3 点原因:

(1) 观察发现，刚性三维模型具有的特点：a. 多由规则的几何体构成，模型表面的几何信息比较单一，多是平面或者规则的曲面；b. 局部特征往往集中于特定的功能区域，而不是分散于模型的整个表面，因此不同视角下视图捕捉的特征信息较为相似，区别力弱. 相应地，非刚性三维模型具有的特点是: a. 多由不规则的几何体构成，模型表面的几何信息较丰富，多是不规则的曲面；b. 形体变化多，局部特征分布广. 不同视角下捕捉的特征信息相对丰富，差异较大，区别力强. 因此，基于多视图的算法在非刚性三维模型中表现更优.

(2) 非刚性数据集内的不同类模型间整体差异较大，而刚性模型数据集内的部分类间模型整体相似，只是在局部区域上有所区别，如图 6 所示. 本文算法可以独立、均等地看待三维模型的多个视图，而多数视图都无法捕捉到模型的这些局部细节信息，在投票后难以有效地区分这类模型的类别信息. 因此，后期考虑进一步引入多尺度视图以增强本文算法的局部识别能力.

(3) ModelNet40 部分实验数据存在分类较为模糊的问题，如 flower_pot 类内有的模型仅仅包含

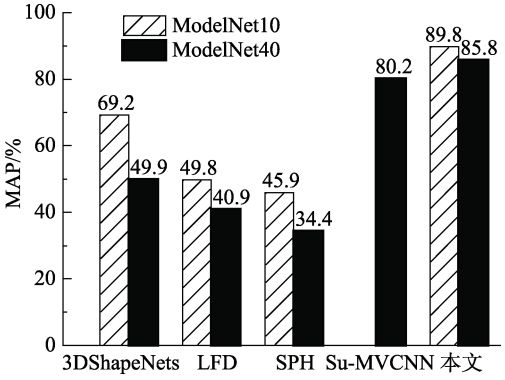


图 6　刚性三维模型检索算法 MAP 对比

花盆, 而有的模型既包含花盆又包含花, 与 plant 类造成了混淆. 本文算法针对 flower_pot 类的分类准确率仅为 20.00%, 这也影响了算法的整体分类性能. 因此, 在后期, 需区别对待模型内部不同局部区域, 甚至可能需要忽略部分局部区域(如 flower_pot 中的花), 而只考虑部分显著特征区域.

3.2　检索实验

(1) 视图及数据选择. 检索实验中, 令视图数目 $n=12$, 选取数据集 ModelNet 和 SHREC 完成测试. 其中, 针对 ModelNet 数据集, 同 Su-MVCNN[14] 一样, 每个类内选择测试集的前 20 个模型和训练集的前 80 个模型组成检索实验的数据子集; 针对 SHREC 数据集, 由于 5 次随机分类实验结果接近, 因此, 检索中随机选取其中 1 组数据来评价本文算法的检索性能.

(2) 检索步骤. 给定一个三维模型, 首先将其转化为 12 个视图, 分别输入训练好的 CaffeNet 网络中, 生成其描述符及视图分类信息; 然后利用投票机制形成模型分类, 并基于 4 种不同距离度量算法计算模型之间的距离, 生成邻接矩阵, 排序并返回检索结果.

3.2.1　4 种距离度量算法的检索实验对比

以平均精度均值(mean average precision, MAP)为评价标准, 表 7 对比了本文 4 种检索算法的检索准确率. 由表 7 可见:

(1) 基于 4 种不同的距离度量算法实验结果略有差异. 相对来说, 效果最好的是基于 V2A 的算法, 效果最差的是 A2A 算法. 这是因为, A2A 算法简单地考虑模型的所有视图, 且通过平均距离模糊化了模型视图间的相似性; V2A 算法考虑了投票正确的那些视图, 而忽略了投票错误的视图, 即对模型多个视图的代表性进行了甄别, 这在一定程度上提高了检索的准确性.

(2) 作为考虑代表性视图的 2 种算法, V2A 略优于 V2V 算法. 这是由于在 V2A 中, 对于给定模型的代表性视图, 将其与另外一个模型的距离转换为与其所有视图中最接近视图的距离, 这也符合人类认知中从某个显著角度观察物体相似性的基本原则; 而 V2V 中, 却将双方视图均限定在投票正确的视图集内, 限制过严.

表 7　本文算法检索 MAP 对比　　%

数据集	A2A	MinA2A	V2A	V2V
ModelNet10	89.19	**90.26**	89.78	89.27
ModelNet40	85.23	85.16	**85.84**	84.91
SHREC10	98.74	**100.00**	**100.00**	**100.00**
SHREC11	98.62	98.70	**99.08**	98.95
SHREC15	99.30	99.28	**99.53**	99.44

3.2.2　刚性三维模型检索实验对比

(1) 检索算法的选择. 为了更加全面地进行比较, 本文选择 LFD[29], 3DShapeNets[3], Su-MVCNN[14] 和基于球面调和描述子(spherial harmonics, SPH)[32] 的检索算法进行对比, 实验中, 本文算法统一基于 V2A 距离度量.

(2) 检索指标的选择. 同 3D ShapeNets[3]一样, 在刚性三维模型检索中, 本文选取了 MAP 及查全查准曲线(P-R 曲线)作为检索性能评价标准.

(3) 检索结果及分析. 图 6 和图 7 分别给出了本文算法和 4 种典型算法的 MAP 和 P-R 曲线. 通过对比可以看出, 针对刚性三维模型, 本文基于 CNN 和投票机制算法的检索效果明显优于其他算法. 尤其是针对 ModelNet10, 本文算法的平均准确率高达 89.78%. 这是因为: a. 本文检索算法中相对独立地考虑了各个视图间的距离. 而得益于丰富的 ImageNet 资源和相对成熟的 CaffeNet, 本文网络提取的特征良好地表达了视图属性, 确保了视图表征的准确性. b. 本文仅考虑三维模型代表性视图间的距离, 而视图的代表性与模型分类紧密相关, 高准确性的分类结果确保了高精度的检索结果.

(4) 各个类的检索结果分析. 图 8 以 ModelNet10 为代表, 给出了各个类的 P-R 曲线. 可以看出, 各个类的检索性能差异较大. 通过与表 5 对比可见, 各个类的检索准确率同分类准确率具有极大的相关性, 检索性能最差的类也正是分类准确率最低的类, 依次是类 table, desk 和 dresser.

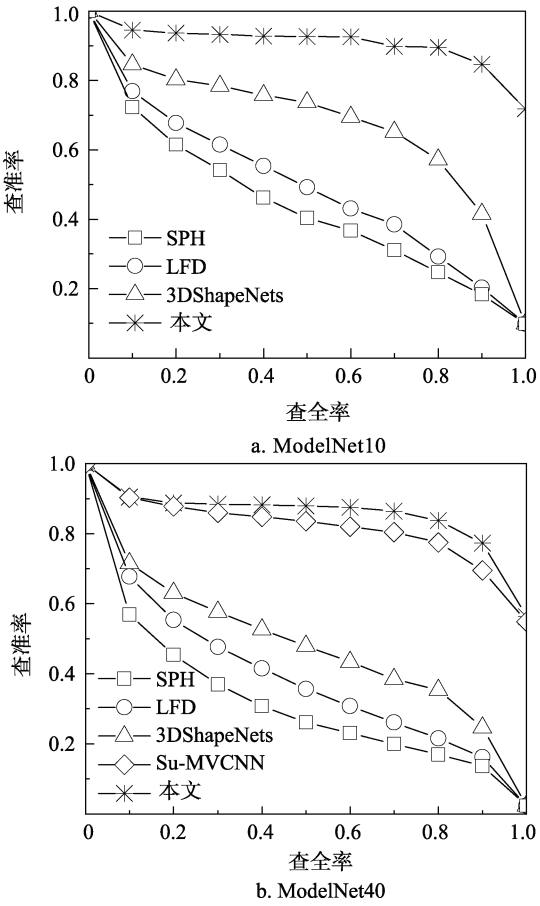


图 7 各种算法在刚性三维模型检索中的 P-R 曲线

3.2.3 非刚性三维模型检索实验对比

(1) 检索算法的选择. 针对非刚性三维模型, 本文从 SHERC10, SHERC11, SHERC15 数据集的官方网站上分别遴选了 3 个具有代表性(通常, 差的一种, 中等的一种, 好的一种)的检索算法与本文算法进行比较. 实验中, 本文算法统一基于 V2A 距离度量.

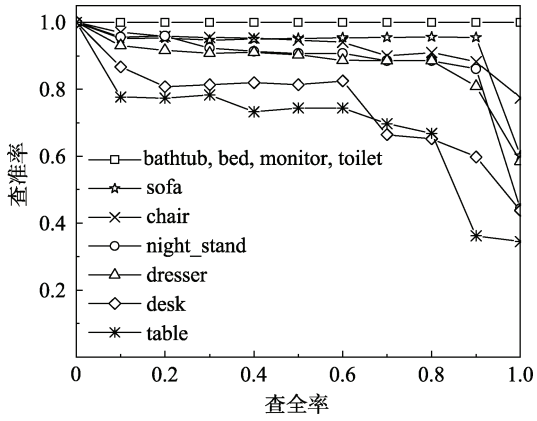


图 8 ModelNet10 各个类的检索 P-R 曲线

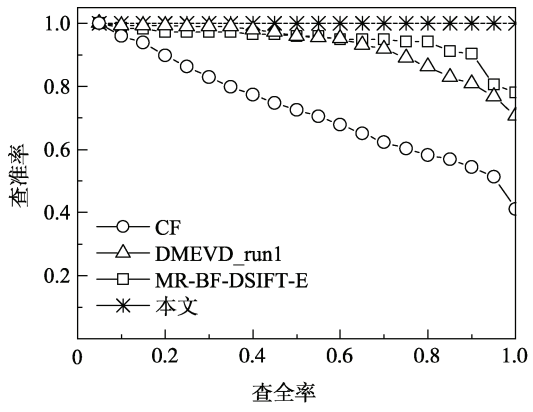
(2) 检索结果及分析. 基于 SHREC 数据集官方网站[23-25], 本文选取了 NN, FT, ST, E, DCG 以及 P-R 曲线作为检索性能评价标准, 检索结果如表 8 及图 9 所示. 实验表明, 与其他算法相比, 本文算法在 SHREC10 及 SHREC11 上的检索性能整体最优, 这充分验证了本文算法既适用于刚性三维模型, 也适用于非刚性三维模型. 另一方面, 由于本文算法在模型整体相似、局部不同时, 其识别能力较弱. 针对非刚性三维模型, 在诸如类 man 和 woman 整体相似模型的分类及检索中存在较多错误, 造成了本文检索准确率略差于算法 SV-LSE-kpaca50.

3.3 检索效率评价

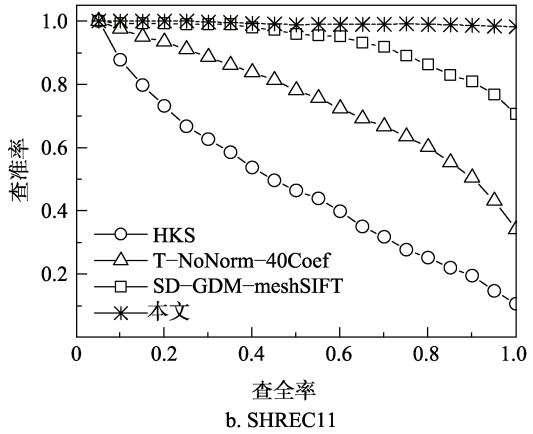
检索时, 需要遍历模型库内所有模型, 计算待检索模型与库中模型之间的 L_2 距离, 而时间主要

表 8 非刚性三维模型数据集检索结果对比 %

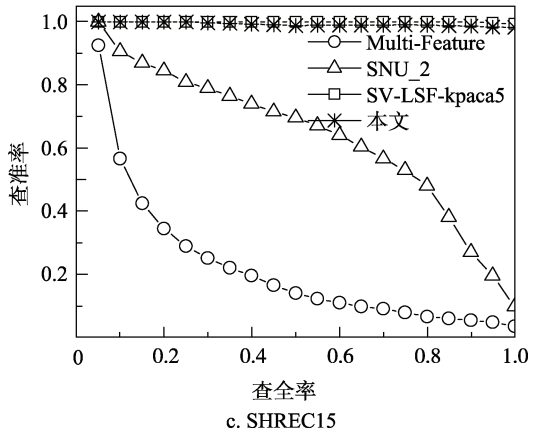
数据集	算法	评价标准				
		NN	FT	ST	E	DCG
SHREC10	MR-BF-DSIFT-E	0.985 0	0.909 2	0.963 2	0.705 5	0.976 3
	DMEVD_run1	**1.000 0**	0.861 1	0.957 1	0.701 2	0.977 3
	CF	0.920 0	0.634 7	0.780 0	0.552 7	0.878 1
	本文	**1.000 0**	**1.000 0**	**1.000 0**	**0.828 6**	**1.000 0**
SHREC11	T-NoNorm-40Coef	0.955 0	0.671 7	0.802 6	0.579 1	0.897 2
	HKS	0.836 7	0.406 1	0.497 3	0.352 5	0.730 4
	SD-GDM-meshSIFT	**1.000 0**	0.972 0	0.990 1	0.735 8	**0.995 5**
	本文	0.975 0	**0.977 8**	**0.997 2**	**0.828 6**	0.990 2
SHREC15	SV-LSF-kpaca50	**1.000 0**	**0.997 2**	**0.999 7**	**0.835 7**	**0.999 7**
	SNU_2	0.899 2	0.565 7	0.670 6	0.518 1	0.833 5
	Multi-Feature	0.450 8	0.186 4	0.262 5	0.184 6	0.525 9
	本文	0.996 0	0.989 0	0.996 0	0.777 8	0.994 9



a. SHREC10



b. SHREC11



c. SHREC15

图 9　各种算法在非刚性三维模型检索中的 P-R 曲线

耗费在 L_2 距离的计算上. 理论分析本文算法的检索时间复杂度为 $O(nm^2)$, 其中, n 为库内待检索模型的数目, m 为三维模型所对应的视图数目. 为进一步验证, 本文先后构造了包含 50, 100, 200, 400, 800 个三维模型的检索库; 针对每个三维模型, 分别构造了包含 6 个视图, 12 个视图和 80 个视图的投票算法; 并以第 3.1 节中所述硬件配置为基础, 运行了本文所提出的 V2A 检索算法, 其运行时间如表 9 所示.

表 9　本文算法检索效率分析　　s

视图数目	模型数				
	50	100	200	400	800
6	0.055	0.108	0.217	0.431	0.862
12	0.235	0.466	0.939	1.871	3.767
80	10.001	19.778	39.937	81.683	159.060

从表 9 可以看出, 在被检索模型数目逐渐增多时, 检索所需时间呈线性增长; 在表征三维模型的视图数目逐渐增加时, 检索所需时间呈二次增长. 因此, 在实际应用中: (1) 可以利用待检索模型属于各个类的概率, 选取部分相关较大的类中的模型作为待检索模型, 从而极大地缩小检索范围, 提高检索效率; (2) 在满足检索准确率的情况下, 选择尽可能少的视图数目, 以进一步提高检索效率; (3) 随着 GPU 的发展和普及, 可以利用 GPU 来加速检索中 L_2 距离的计算, 从而数倍乃至数十倍地提高检索效率.

4　总结和展望

本文提出了一种基于 CNN 和投票机制的三维模型检索算法: 将三维模型转换为二维视图集合, 并利用 CNN 建立视图分类和表征, 利用投票机制完成三维模型的分类; 基于视图分类、表征及投票结果, 提出了 4 种不同的距离度量算法, 并以此为基础完成三维模型检索. 实验表明, 本文算法简单、有效, 同时适用于刚性三维模型和非刚性三维模型的分类及检索.

实验中发现, 本文算法存在局部特征识别及细分类能力较弱的问题. 下一步将针对这一问题展开研究, 提升本文算法在细分类方面的检索能力.

参考文献(References):

[1] Qin F W, Li L Y, Gao S M, *et al*. A deep learning approach to the classification of 3D CAD models[J]. Journal of Zhejiang University: Science C, 2014, 15(2): 91-106

[2] Xie J, Dai G X, Zhu F, *et al*. DeepShape: deep-learned shape descriptor for 3D shape retrieval[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(7): 1335-1345

[3] Wu Z R, Song S R, Khosla A, *et al*. 3D ShapeNets: a deep representation for volumetric shapes[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2015: 1912-1920

[4] Xu X, Todorovic S. Beam search for learning a deep convolutional neural network of 3D shapes[C] //Proceedings of the 23rd International Conference on Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2016: 3506-3511

[5] Maturana D, Scherer S. VoxNet: a 3D convolutional neural network for real-time object recognition[C] //Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. Los Alamitos: IEEE Computer Society Press, 2015: 922-928

[6] Sedaghat N, Zolfaghari M, Brox T. Orientation-boosted voxel nets for 3D object recognition[OL]. [2018-03-02]. https://arxiv.org/abs/1604.03351

[7] Garcia-Garcia A, Gomez-Donoso F, Garcia-Rodriguez J, *et al.* PointNet: a 3D convolutional neural network for real-time object class recognition[C] //Proceedings of the International Joint Conference on Neural Networks. Los Alamitos: IEEE Computer Society Press, 2016: 1578-1584

[8] Xie Zhige, Wang Yueqing, Dou Yong, *et al*. 3D feature learning via convolutional auto-encoder extreme learning machine[J]. Journal of Computer-Aided Design and Computer Graphics, 2015, 27(11): 2058-2064(in Chinese)
(谢智歌, 王岳青, 窦 勇, 等. 基于卷积-自动编码机的三维形状特征学习[J]. 计算机辅助设计与图形学学报, 2015, 27(11): 2058-2064)

[9] Riegler G, Ulusoy A O, Geiger A. OctNet: learning deep 3D representations at high resolutions[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2017: 6620-6629

[10] Wang P S, Liu Y, Guo Y X, *et al*. O-CNN: octree-based convolutional neural networks for 3D shape analysis[J]. ACM Transactions on Graphics, 2017, 36(4): Article No.72

[11] Brock A, Lim T, Ritchie J M, *et al*. Generative and discriminative voxel modeling with convolutional neural networks[OL]. [2018-03-02]. https://arxiv.org/abs/1608.04236

[12] Shi B G, Bai S, Zhou Z C, *et al.* DeepPano: deep panoramic representation for 3-D shape recognition[J]. IEEE Signal Processing Letters, 2015, 22(12): 2339-2343

[13] Sinha A, Bai J, Ramani K. Deep learning 3D shape surfaces using geometry images[C] //Proceedings of the European Conference on Computer Vision. Heidelberg: Springer, 2016: 223-240

[14] Su H, Maji S, Kalogerakis E, *et al*. Multi-view convolutional neural networks for 3D shape recognition[C] //Proceedings of the IEEE International Conference on Computer Vision. Los Alamitos: IEEE Computer Society Press, 2015: 945-953

[15] Wang C, Pelillo M, Siddiqi K, *et al*. Dominant set clustering and pooling for multi-view 3D object recognition[C] //Proceedings of the British Machine Vision Conference. Heidelberg: Springer, 2017: 1-11

[16] Ma Y X, Zheng B, Guo Y L, *et al*. Boosting multi-view convolutional neural networks for 3D object recognition via view saliency[C] //Proceedings of the Chinese Conference on Image and Graphics Technologies. Heidelberg: Springer, 2017: 199-209

[17] Johns E, Leutenegger S, Davison A J. Pairwise decomposition of image sequences for active multi-view recognition[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2016: 3813-3822

[18] Bai S, Bai X, Zhou Z, *et al.* GIFT: a real-time and scalable 3D shape search engine[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2016: 5023-5032

[19] Phong B T. Illumination for computer generated pictures[J]. Communications of the ACM, 1975, 18(6): 311-317

[20] Jia Y Q, Shelhamer E, Donahue J, *et al*. Caffe: convolutional architecture for fast feature embedding[C] //Proceedings of the 22nd ACM International Conference on Multimedia. New York: ACM Press, 2014: 675-678

[21] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks[C] //Proceedings of the 25th International Conference on Neural Information Processing Systems. New York: Curran Associates Inc, 2012, 1: 1097-1105

[22] Shilane P, Min P, Kazhdan M, *et al*. The princeton shape benchmark[C] //Proceedings of the Shape Modeling Applications. Los Alamitos: IEEE Computer Society Press, 2004: 167-178

[23] Lian Z, Godil A, Fabry T, *et al*. SHREC'10 track: non-rigid 3D shape retrieval[C] //Proceedings of the 3rd Eurographics Conference on 3D Object Retrieval. Aire-la-Ville: Eurographics Association Press, 2010: 101-108

[24] Lian Z, Godil A, Bustos B, *et al*. SHREC'11 track: shape retrieval on non-rigid 3D watertight meshes[C] //Proceedings of the 4th Eurographics Conference 3D Object Retrieval. Aire-la-Ville: Eurographics Association Press, 2011: 79-88

[25] Lian Z, Zhang J, Choi S, *et al*. SHREC'15 track: non-rigid 3D shape retrieval[C] //Proceedings of the Eurographics Workshop on 3D Object Retrieval. Aire-la-Ville: Eurographics Association Press, 2015: 257-266

[26] Hegde V, Zadeh R. FusionNet: 3D object classification using multiple data representations[OL]. [2018-03-02]. https://arxiv.org/abs/1607.05695

[27] Qi C R, Su H, Nießner M, *et al*. Volumetric and multi-view CNNs for object classification on 3D data[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2016: 5648-5656

[28] Zhi S F, Liu Y X, Li X, *et al*. Towards real-time 3D object recognition: a lightweight volumetric CNN framework using multitask learning[J]. Computers & Graphics, 2018, 71: 199-207

[29] Chen D Y, Tian X P, Shen Y T, *et al*. On visual similarity based 3D model retrieval[J]. Computer Graphics Forum, 2003, 22(3): 223-232

[30] Shen L, Makedon F. Spherical mapping for processing of 3D closed surfaces[J]. Image and Vision Computing, 2006, 24(7): 743-761

[31] Gu X F, Wang Y L, Chan T F, *et al*. Genus zero surface conformal mapping and its application to brain surface mapping[J]. IEEE Transactions on Medical Imaging, 2004, 23(8): 949-958

[32] Kazhdan M, Funkhouser T, Rusinkiewicz S. Rotation invariant spherical harmonic representation of 3D shape descriptors[C] //Proceedings of the Eurographics/ACM SIGGRAPH Symposium on Geometry Processing. Aire-la-Ville: Eurographics Association Press, 2003: 156-164