基于八叉树结构的三维体素模型检索

张满囤[1),2)] 燕明晓[1),2),3)] 马英石[1),2),3)] 王 红[1),2)] 刘 伟[3),4)] 黄向生[5)]

[1)](河北工业大学人工智能与数据科学学院 天津 300401)

[2)](天津市虚拟现实与可视计算国际联合中心 天津 300401)

[3)](维尔科宝(天津)科技有限公司 天津 300401)

[4)](河北工业大学机械工程学院 天津 300401)

[5)](中国科学院自动化研究所 北京 100190)

摘 要 随着 VR/AR 技术发展以及三维模型的广泛应用,实现三维检索具有越来越重要的现实意义.基于模型的检索较好地保留了模型的空间信息和几何特征,其不仅包含模型的表面信息而且还包含模型的内部属性.但是,基于模型的检索往往存在着高存储、高计算的问题.为了解决该问题,本文研究了三维模型预处理及三维模型表示的方法,提出了一种基于八叉树结构的三维体素模型检索方法,即将模型进行体素化处理后提取模型的粗粒度特征和细粒度特征,将两种特征进行融合用八叉树形式表达特征,输入到卷积神经网络中进行训练,最终通过特征的欧氏距离度量实现模型的检索.运用八叉树特征表示法,可以有效地节省体素化存储过程的空间占用量,而且也能保留原始三维网格模型的细节信息.同时考虑到计算性能,本文还在模型体素化的过程中做出一定的改进,通过仅对模型外表面进行体素化,实现了对体素化过程以及数据存储和卷积神经网络训练的优化,大大降低了时间开销.实验中将三维体素模型特征存储在八叉树结构中作为卷积神经网络的输入,结合 SOFTMAX 代价函数,通过大量的模型训练数据,对该卷积神经网络模型进行训练.与其他同类算法对比,证明了该算法在三维模型检索中的优越性.

关键词 特征融合;卷积神经网络;八叉树;模型检索;相似性匹配

中图法分类号 TP391 **DOI 号** 10.11897/SP.J.1016.2021.00334

3D Voxel Model Retrieval Based on Octree Structure

ZHANG Man-Dun[1),2)] YAN Ming-Xiao[1),2),3)] MA Ying-Shi[1),2),3)] WANG Hong[1),2)]
LIU Wei[3),4)] HUANG Xiang-Sheng[5)]

[1)](*School of Artificial Intelligence, Hebei University of Technology, Tianjin* 300401)

[2)](*Tianjin International Joint Center for Virtual Reality and Visual Computing, Tianjin* 300401)

[3)](*Weier Kebao (Tianjin) Science & Technology Co. Ltd, Tianjin* 300401)

[4)](*School of Mechanical Engineering, Hebei University of Technology, Tianjin* 300401)

[5)](*Institute of Automation, Chinese Academy of Sciences, Beijing* 100190)

Abstract With the development of VR/AR technology and the wider 3D applications, it is realized that 3D model retrieval is becoming more and more important. Model-based retrieval preserves the spatial and geometric features, which includes not only the surface information but also the internal properties of the model. However, there are concerns in relate to its high storage and high computation. Deep learning has demonstrated successful breakthroughs in the fields of speech recognition, graphic image classification and natural language processing etc. In this paper, after studying the 3D model pre-processing and 3D model representation, a method combined

收稿日期:2019-04-30;在线发布日期:2020-02-07.本课题得到国家自然科学基金(61573356)、天津市企业科技特派员项目(18JCTPJC58700)、河北省自然科学基金(F2019202054)资助.**张满囤**,博士,副教授,主要研究方向为计算机图形学、图像处理、三维模型检索和三维物体识别.E-mail:zhangmandun@scse.hebut.edu.cn.**燕明晓**,硕士研究生,主要研究方向为三维模型检索.**马英石**,硕士研究生,主要研究方向为三维模型检索.**王 红**,硕士研究生,主要研究方向为三维模型检索.**刘 伟**(通信作者),博士,副教授,主要研究方向为精度标定、图形图像处理、虚拟现实.E-mail:14531705@qq.com.**黄向生**,博士,副研究员,主要研究方向为人工智能、机器学习、态度感知与决策、自主学习.

with 3D voxel model and octree structure is proposed for 3D model retrieval. First of all, the coarse-*grained features and fine*-grained features of the voxelization model are extracted. After the fusion, the features which expressed in the form of octree are input into the convolutional neural network for training, and the Euclidean distance is the metric for evaluating and retrieving the model in the end. In order to form an octree for storing the 3D model, eight equal cubic meshes are able to be divided after the 3D model is scaled and aligned with a standard unit 3D boundary cubic volume. Such a mesh process will continue for each cubic volume, which include the 3D model, until the mesh quality reaches the requirement. By using the octree feature representation, not only the storage consumption is effective reduced due to the process of voxelization, the details of the original 3D mesh model are also preserved. The presented algorithm uses the improved Octree structure as the basic data structure of the model voxelization which is applied to the convolutional neural network for model classification. By designing a novel spatial octree, a 3D model is represented by which surface information was stored into the leaf nodes of the octree. The leaf nodes are able to be trained as initial data and evaluated through the improved octree neural network structure on GPU. IO-CNN is able to support various CNN structures with different 3D representations to extract and classify the 3D model for 3D model retrieval. With careful analyzing of 3D model, it is found that it is unnecessary to process interior part of the 3D model for voxelization if it is a closed 3D geometry. The voxelization of the interior part of the 3D model will never affect the representation of geometric features. In considering the computational performance, such modifications were made during the process of model voxelization. After voxelization, the normalized 3D model is represented by octree for obtaining the spatial information. An iterate process is carried out which 1 is set to the region which includes the 3D model and 0 is set to the region without the model respectively. By only voxelizing the outer surface of the model, the computational overhead is greatly reduced due to the optimized data storage and convolutional neural network training. The experiment has shown, with applying the SOFTMAX cost function, after a large amount of training data through convolutional neural network, the presented algorithm has more advance in 3D model retrieval than other similar algorithms.

Keywords　feature fusion; convolutional neural network; octree; model retrieval; similarity match

1　引　言

随着三维传感技术和三维建模技术的飞速发展，三维模型在三维游戏、虚拟现实、工业设计、影视娱乐等方面的应用越来越广泛. 三维模型数量的爆发式增长，使得如何高效准确地在海量三维模型中检索到需要的模型，具有重要的研究意义. 深度学习作为目前最为火热的研究方向，在语音识别、图形图像分类和自然语言处理等领域取得众多具有突破性的成果[1]. 近年来三维模型领域的工作者也一直在尝试使用神经网络模型结构来进行三维模型分类和检索工作. Babenko 等人[2]将深度学习知识引入图像检索领域，他们验证了从深度学习网络中提取出来的图像特征是可以直接用于检索的. Wan 等人[3]对深度学习在图像检索上的应用进行了一次较为全面的研究，发现通过在目标数据集上进行 fine-tuning 的训练后提取特征，可以取得比传统的人工特征更好的效果. 在三维模型与深度学习网络的相关结合应用中，3D ShapeNet[4]通过深度置信网络学习三维模型在空间中的概率分布，然后提取相应特征，再将其用于模型识别及检索，这样的方法取得了较好的效果. 当然，基于深度学习的三维模型检索研究中也存在着许多的技术难点需要进一步的探索和研究. 大致归结为以下三个方面：(1) 对于大数据集的检索工作，深度学习虽然可以达到较高的准确率，但是存在高存储、高计算的问题；(2) 对于三维模型的检索，应该探索更为合理的三维模型的形状特征；(3) 优化深度学习网络的算法复杂度，提高精确度.

目前，现有的三维模型检索工作大致可以分为

两类:基于模型特征的检索[5-6]和基于视图特征的检索[7-10].基于视图特征的 3D 模型检索,一般情况下主要是通过两个 3D 对象之间视图特征的相似性匹配来完成.其核心思想是用一张或多张二维视图来表示该 3D 模型,从而实现模型的降维检索,将模型特征进行降维处理大大降低了模型检索的难度.而且选择代表视图来表示模型特征,能够减少模型的无用信息,提高模型检索的效率.但是,如何获取模型的代表视图、视图个数的确定、多个视图特征信息的融合以及各个视图之间的遮挡问题,是基于视图检索中存在的难以解决且不容忽视的问题.基于模型特征的检索是直接作用于模型对象的原生三维表示,如多边形网格、基于体素的离散化、点云或隐式曲面,能够较好地保留模型的空间信息和几何特征.本文采用三维体素作为原始的三维表示,学习高层次的三维模型特征[11].使用三维体素化表示的原因如下:首先,三维曲面具有不规则的顶点拓扑结构和任意的网格分辨率,而体素化被认为是能够将三维曲面转化为规则的并且充分保留三维曲面几何结构的最简单直观的离散方法.离散体素结构和规则体素结构使得卷积神经网络能够直接从三维形状中进行特征学习.其次,三维体素化能够提供视觉评价,通过三维领域的深度学习模型,获得常用的三维曲面表示,与基于视图特征的方法中的投影深度等三维模型的表示相比,这是体素化特有的性质.最后,将模型体素化不仅可以保留模型的表面信息还可以描述模型的内部信息.

本文对模型进行体素化处理后,构造一种特殊的八叉树三维模型表达方式,之后对模型八叉树进行二值化转换.通过将模型进行八叉树划分,获得模型空间位置的标记,对于未包括三维模型的区间进行置 0 处理,对于包括三维模型的区间进行置 1 处理,并重复迭代,以此来对体素化过程的三维模型进行粗细粒度的划分.然后对划分结果进行数值化表示,构造出模型的二值化三维矩阵,并将其作为卷积神经网络的输入数据,对其进行训练,生成各类模型的特征分类值,最终将待查询模型放入该网络中进行特征值的提取与比对,进行三维模型之间的相似性度量,从而完成模型的检索.

2 模型预处理

2.1 模型标准化

目前,常见的 3D 模型文件格式为 Object File Format(.off),其存储格式为 ASCII 编码.由于三维模型具有不同的空间尺度、不同的空间位置、以及旋转角度和拓扑结构等等,这些都会成为影响三维模型检索精度的重要因素,因此确定统一的三维模型表述方法,保证模型的空间尺度、空间位置以及旋转角度的统一性是三维模型特征提取的重中之重.为保证平移不变性,将模型置于坐标系原点处;为保证旋转的不变性,在一个标准的坐标平面中采用 PCA 变换方法将模型对齐;为保证尺度的不变性,需要将模型归一化到标准单位大小;为保证方位的不变性,需要将模型进行翻转变换.

平移处理需要解决模型的原点位置问题,需要对模型原点的位置进行归一化操作,模型重心平移可以将三维模型的重心平移到标准坐标系的原点上.本文采用对三维网格进行面积加权的方法增加对三维模型表面网格的采样数量,从而减少不同网格面积的三角形网格对模型重心位置产生的偏差.具体表示为式(1).

$$\begin{cases} C=\sum_{i}^{N} P_i \cdot T_i \\ T_i=S_i \Big/ \sum_{i}^{N} S_i \end{cases} \tag{1}$$

上述公式中,C 表示三维网格模型的重心位置,P_i 表示模型中的每个三维网格的重心,S_i 表示三维网格模型中每个三角形面片的表面积.通过这样的方式,可以将三维模型的重心平移到坐标的原点.

旋转处理的主要目的是修正相同模型在不同角度下的偏差.采用面积加权的主成分分析法进行处理.首先采用网格三维模型表面点集合的协方差矩阵来计算各自对应的特征值;然后对特征值进行排序,求出对应的特征向量($\boldsymbol{v}_1$,$\boldsymbol{v}_2$,$\boldsymbol{v}_3$).在特征值排序过程中使用降序排列,可以求得模型顶点分布最为广泛的主方向 $\boldsymbol{v}_1$,后面依次为该模型顶点分布的第二主方向 $\boldsymbol{v}_2$ 和第三主方向 $\boldsymbol{v}_3$.将该序列构成的矩阵进行转置变换可以得到矩阵 $\boldsymbol{R}$.整个过程需要计算协方差矩阵 $\boldsymbol{C}_p$,最终实现对三维网格模型的空间旋转的归一化.同时需要考虑到三角形面片面积的不同,对重心造成的影响同时会对旋转偏角产生一定的影响,在这里需要对面积进行加权来计算上述协方差矩阵,具体过程如式(2).

$$\boldsymbol{C}_p=\sum_{i=1,j=1}^{n,n}(s_i p_i-s_j q_j)(s_i p_j-s_j q_j)^{\mathrm{T}} \tag{2}$$

上述公式中,$\boldsymbol{C}_p$ 表示前面提到的协方差矩阵;p_i,q_j 分别表示每个网格三角形的重心,同时 s_i,s_j 分

别为每个网格三角形面片构成的三角形的面积. 通过这样的方法来对模型进行旋转处理可以得到具有旋转不变性的三维网格模型.

缩放预处理的目的就是将不同尺度的模型缩放到统一的尺度下，再进行后续的特征提取工作，这样能够保证模型特征的统一性和可用性. 本文使用的手段是先获取模型边界点的最大距离，然后根据模型边界的最大距离对模型的大小尺度进行调整，如式(3).

$$K=\sqrt{\frac{K_x^2+K_y^2+K_z^2}{3}} \tag{3}$$

上述公式中，$K_x=\frac{1}{S}\sum_{i=1}^{n}s_i p_{ix}$，$K_y=\frac{1}{S}\sum_{i=1}^{n}s_i p_{iy}$，$K_z=\frac{1}{S}\sum_{i=1}^{n}s_i p_{iz}$ 分别为 x，y，z 轴方向的缩放系数，K 表示整体缩放系数，p_{ix} 表示第 i 个三角形的重心位置到 YOZ 平面的距离，p_{iy} 表示第 i 个三角形的重心位置到 XOZ 平面的距离，p_{iz} 表示第 i 个三角形的重心位置到 XOY 平面的距离. 而 s_i 表示每个三维网格模型中的三角形网格的表面积，S 为三维网格模型表面所有三角形网格面积的总和，$S=\sum_{i=1}^{n}s_i$. 通过将模型中每个点的坐标除以缩放系数 K，缩放后点的坐标与原坐标的关系 $(x_i', y_i', z_i')=(x_i, y_i, z_i)\times K^{-1}$，可以将三维模型缩放到统一的尺度下，实现模型的归一化处理.

模型的翻转处理主要用来修正三维模型的主方向，通常情况下因为三维模型的自有坐标轴的定义有差别，或者是同一模型设置的方向有区别而造成不同模型的主方向的差别. 本文使用的翻转处理方法是首先计算模型表面点位于某个平面正方向的距离和以及位于平面下的点的距离和，通过比较大小来定义模型的正反向，再进行模型翻转，从而保证模型的翻转不变性. 文中定义了三个主方向的特征量 f_x，f_y，f_z，计算翻转矩阵的公式如式(4)所示.

$$\begin{cases} f_x=\frac{1}{S}\sum_{i=1}^{n}sign(x_{Ai}+x_{Bi}+x_{Ci})\cdot s_i\cdot\left(\frac{x_{Ai}+x_{Bi}+x_{Ci}}{3}\right)^2 \\ f_y=\frac{1}{S}\sum_{i=1}^{n}sign(y_{Ai}+y_{Bi}+y_{Ci})\cdot s_i\cdot\left(\frac{y_{Ai}+y_{Bi}+y_{Ci}}{3}\right)^2 \\ f_z=\frac{1}{S}\sum_{i=1}^{n}sign(z_{Ai}+z_{Bi}+z_{Ci})\cdot s_i\cdot\left(\frac{z_{Ai}+z_{Bi}+z_{Ci}}{3}\right)^2 \end{cases} \tag{4}$$

在上述公式中，s_i 表示三维网格模型的三角形网格的面积，S 为三维网格模型表面所有三角形网格面积的总和，x_{Ai} 表示第 i 个三角形网格中的 A 点的 x 坐标. 利用这种方法可以获得三维模型的翻转矩阵 $\boldsymbol{F}$，该矩阵的表示如式(5)所示.

$$\boldsymbol{F}=\begin{bmatrix} sign(f_x) & 0 & 0 \\ 0 & sign(f_y) & 0 \\ 0 & 0 & sign(f_z) \end{bmatrix} \tag{5}$$

通过进行翻转矩阵的运算就完成了模型的翻转处理标准化. 经过对模型进行平移、旋转、尺度归一和翻转处理，可以得到能够进行后续实验的标准化模型. 一个完整的坐标模型标准化的过程由式(6)表示.

$$\tau(I)=K^{-1}\cdot\boldsymbol{F}\cdot\boldsymbol{R}\cdot(I-C) \tag{6}$$

其中，K 为缩放系数，$\boldsymbol{F}$ 为一个对角矩阵形式的翻转矩阵，$\boldsymbol{R}$ 是对应 PCA 变换的旋转矩阵，I 是原始模型的坐标，C 是坐标原点，也是三维模型的重心位置.

2.2　模型体素化

体素化(Voxelization)是将物体的几何表示形式转换成最接近该物体的体素表示形式，产生体数据集[12]. 首先将一个连续的三维空间 R^3 以分块的形式转换为一个离散的三维空间 D^3，离散空间 D^3 的基本单元就是一个个边长为 l 的立方体，这些立方体即为体素. 体素 (x,y,z) 对应到连续的空间上，就包含了一个小的空间区域 $\{(u,v,w)\}$，其中：

$$x-1<u\le x,\ y-1<v\le y,\ z-1<w\le z \tag{7}$$

经过这样的处理，可以得到一个三维模型的空间分布信息. 本文中首先要确定一个固定长度为 L 的立方体盒子，盒子要刚好能够容纳下三维模型，然后要对该模型的所有顶点进行统计，找出其中最远的两点距离 d，之后在前面归一化的三维模型基础上，将模型中所有顶点的坐标值都乘以系数 d/K(K 为归一化的放缩系数)，然后可以得到新的顶点坐标值，这里将新的顶点坐标集合设为 p''，则 p'' 可以表示为式(8).

$$p''=\left\{p''=\frac{p'\times d}{K}\in p'\right\} \tag{8}$$

其中 p' 表示标准化后的模型顶点. 经过这样的操作，使得模型刚好能够处于立方体的中心位置，且被立方体包围. 体素化将连续的矢量转化成为离散的点，对于每一个体素，取值可以是二值化的，也可以是多值化的. 对于二值化的体素，若体素的取值为 0，则表示当前位置没有包含模型部分；若体素值为 1，则表示当前位置包含模型部分. 本文运用的是三维网格模型二值体素化的表示方法.

(1) 顶点体素化

对于网格三维模型而言，连续空间中任意一个

顶点都有一个在离散空间中与之对应的体素点，同理对顶点的体素化过程，也是一对一的映射过程[13]. 本算法需要将模型的连续点映射到体素，从而实现三维模型的体素化. 假设通过坐标尺度归一化处理后的三维模型刚好能被一个 $L\times L\times L$ 的立方体包围盒所包围，如果将该立方体包围盒分为：$n\times n\times n$ 个体素，那么每个体素的大小为 $w\times w\times w$，其中 $w=L/n$. 因此对于三维网格模型上的连续点 (x,y,z)，映射到体素 (x',y',z') 的过程如式(9)所示.

$$x'=\left[\frac{x}{w}\right]$$
$$y'=\left[\frac{y}{w}\right] \tag{9}$$
$$z'=\left[\frac{z}{w}\right]$$

(2) 边缘体素化

对于三维网格模型边的体素化，可以分为两种情况：第一种是边的长度 l 小于体素的边长 w，在这种情况下，要么边的两个端点在同一个体素内，要么两个顶点在两个体素内，此时根据实际情况进行体素化即可. 第二种是边的长度 l 大于体素的边长 w，这时需要将这个边进行等分，将 l 分成 n_l 个新的边 l'，使得每个 l' 都小于 w 即可. 这样就可以按照第一种情况下的方法实现对边的体素化处理.

(3) 三角形面片体素化

对于网格三维模型表面的三角形而言，如果它的三个边的长度都小于体素的宽度 w，那么三角形的三个顶点会映射到同一个体素，或是相邻的体素当中，此时仅对三个顶点进行体素化，就可以实现三角形的体素化.

如果三角形的边长超过体素的宽度 w，需要对三角形进行切割，将其分成多个小三角形，其中小三角形的边长应满足任意边的边长都不超过体素宽度 w. 之后按照三角形边长小于体素宽度 w 的方式对其进行体素化.

至此，就实现了三维网格模型的体素化过程. 其具体形式如图 1 所示.

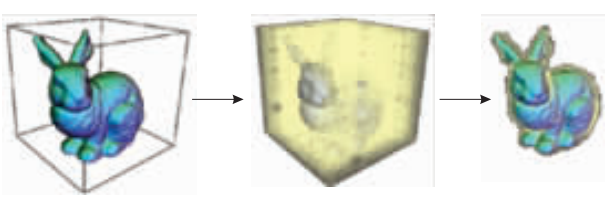

图 1 模型体素化

3 特征表示

3.1 特征提取方法

计算机图形学领域出现了很多关于三维模型的检索方法. 检索过程中对三维模型特征进行提取的部分称为特征提取. 一般来说，模型的几何特征、拓扑特征、视觉特征以及一些融合特征都可以作为模型对比和描述的标准，而且这些特征一般都表现为高维向量，也称为特征向量，该向量就是模型特征的载体. 除此之外一些研究方法中也使用模型的图结构作为模型特征的载体. 这些特征提取的算法都被称为特征描述子. 它可以量化地表示三维模型结构，使得三维模型之间的距离度量成为可能. 也正是因为这一点，特征提取成为了三维模型检索技术研究的关键内容.

任意两点距离特征(D2 Shape Distribution，D2)算法是基于统计特征的特征提取方法[14]. D2 距离即模型表面任意一个随机点到另一个随机点的距离. 常用手段是先将三维模型的不同特征进行采样，然后利用直方图存储这些统计特征，最后利用统计后的特征向量进行模型之间的相似性度量. Osada 等人[15]提出了一种基于模型几何形状分布(Shape Distribution)进行采样的特征提取算法. D2 算法也属于形状分布算法，它从测量 3D 模型的几何属性角度，来解决三维模型特征提取问题，其形状函数中采样的最终表示形式为概率分布. D2 算法的优点是可以快速、轻松地计算样本，并且所得到的分布形状对于相似性变换、噪声和曲面细分等是不变的，具有尺度不变性的优势.

多方位深度傅里叶描述子(Multiple Orientations Depth Flourier Descriptor，MODFD)算法[16]提出了一种基于多边形集合(三维网格模型)的三维模型相似性度量的检索算法. 三维模型相似性度量的一个主要问题就是要表达出这些三维模型的形状多样性. 对于刚体模型而言，三维模型是易于定义和处理的，但其它的非刚体模型如三维网格模型却存在许多问题. 事实上，三维网格模型通常是不定义三维形状，而是由独立的多边形、线条和流形网格集合而产生的三维模型的错觉. MODFD 算法的三维模型相似性度量最显著的特点是它接受三维网格模型和其他定义形式的三维模型. 该方法仅使用模型的渲染外观作为三维模型相似性度量的基本元素. 它通过进行归一化处理消除了由尺度和位置带来的模型差

异，结合立体角离散采样和旋转不变二维图像相似性比较算法消除了三个旋转自由度. 该算法实现过程为首先进行模型规范化处理，然后计算一组从42个视点观察到的三维模型深度图像，以近似地使用离散型的视图来覆盖整个模型的所有可能视图，然后计算每个视点的特征向量. 每个视图的特征向量是基于二维图像的旋转不变性来构造的傅里叶描述子. 最终向量由42个特征向量组合成模型的整体特征.

绝对角距离直方图(Absolute Angle Distance histogram，AAD)算法[16]提出了一种基于三维模型视觉特征相似度的形状特征描述，用于进行三维模型之间的相似性度量. 其方法可以概述为：首先将基于输入曲面的模型转换为定向点集模型，然后计算每一对点的距离和方向并生成二维联合柱状图. 该算法的优点是：可以对非刚体或非流形模型进行计算；该算法对模型的相似变换具有不变性；同时该算法对拓扑变化和几何误差以及退化具有一定的鲁棒性. AAD算法是基于角距离直方图特征算法的改进，改进了角距离直方图(Angle Distance histogram，AD)算法的方向向量描述子对方向的敏感度. 对于AD算法，如果要比较的模型之间具有一致的表面方向，例如，多边形之间顶点的遍历顺序是一致的，则AD形状功能表现良好. 但是，如果数据库所包含的模型具有方向不一致的曲面，AD算法的性能将会受到严重影响，使用不同形状建模工具生成的模型在确定表面方向时可能有不同的规则.

3.2　基于八叉树的模型表示

在Oct-Net[17-19]的启发下，本文提出了一种基于八叉树的三维模型表示方法，并将其作为卷积神经网络的输入. 八叉树的逻辑为：假设要表示的模型V可以放在一个充分大的正方体C内，C的边长为$2n$，则它的八叉树可以用以下的递归方法来定义，八叉树的每个节点与C的一个子立方体对应，树根与C本身相对应，如果$V=C$，那么V的八叉树仅有树根，如果$V\neq C$，则将C等分为八个子立方体，每个子立方体与树根的一个子节点相对应. 只要某个子立方体不是完全空白或完全为V所占据，就要被八等分，从而对应的节点也就有了八个子节点. 这样的递归判断和分割一直要进行到节点所对应的立方体或是完全空白，或是完全为V占据，或是其大小已是预先定义的体素大小. 需要对C与V之交设置某一个阈值，使体素或认为是空白的，或认为是V占据的.

为了构造一个输入三维模型的八叉树，本文首先将三维形状均匀地缩放成一个轴向对齐的单元三维边界立方体，然后对其进行细化八等分. 在每个步骤中，遍历当前深度$l(l\leq 5)$处三维形状边界所占据的所有非空八分区，并在下一个深度$l+1$处将它们细分为八个子八分区. 重复这个过程，直到达到预定义的八叉树深度d. 由此可以得到模型的八叉树分割，从第一层开始标记，若是分割的八分区覆盖了模型则将此分区值赋值为1，若未覆盖则赋值为0. 按照层数依次标记，最终得到模型的八叉树数值表达式.

3.3　粗粒度特征与细粒度特征融合

本文通过对三维网格模型进行体素化处理，可以得到三维网格模型的体素化表示. 虽然这样能够简单地得到体素，并将其作为卷积神经网络的输入对象进行训练，但是实际过程中这样简单地模型体素化，并不能满足详细表达三维模型的要求. 其原因如下：如果仅使用粗粒度体素表示三维模型，体素化之后会失去三维网格模型的细节信息；如果仅使用细粒度体素表示三维模型，那么计算机的内存使用会过于庞大，无法满足计算需求.

因此，找到适合的体素分割粒度是解决问题的关键. 通过分析三维模型，发现对于封闭的三维模型而言，其内部的封闭空间是不需要进行体素化表示的. 因为即使对三维模型内部进行了体素化表示，也不会对三维模型的外观特征产生任何影响[20-22]. 所以本文使用一种单独对三维模型表面进行体素化的表示方法来节省三维模型体素的存储空间，同时对三维模型进行八叉树分割. 如果模型表面细节表示较为丰富，那么就对该区域的网格进行更细粒度的体素化；如果该区域表面较为平滑，就进行粗粒度的体素化表示. 将模型的粗粒度特征和细粒度特征进行结合，可以有效地节省体素化存储过程的空间占用量，同时也能保留原始三维网格模型的细节信息.

在三维网格模型的体素化过程中，对于单位区域内包含的三角形面片数量多于n的区域，称为细粒度区域；单位区域内包含的三角形面片数量少于n的区域，称为粗粒度区域. 实验中按照以下步骤对原始的三维网格模型进行体素化处理：

首先对三维网格模型进行八叉树分割，然后对每个区域进行均匀分割. 如果分割后在该立方体内包含的三维模型的三角形面片数量多于n，那么就继续对分割后的区域进行八叉树分割，并重复顶点

体素化的操作;如果在分割后的立方体内包含的三维模型的三角形面片数量少于 n,那么就停止进行八叉树分割.

然后将分割后的三维模型进行八叉树区域内的体素化.

最终经过 $m(m\leq 5)$次的迭代,可以获得不同粒度的模型体素化结果,如图 2 所示.

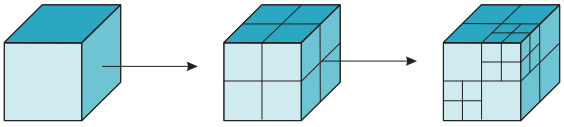

图 2 模型八叉树分割

在对体素化的三维模型进行数值化表示的时候,本文选择对包括三维模型的区域进行置 1 操作,对不包含三维模型的区域进行置 0 操作.用同样的方法进行迭代处理分割模型,具体的实现过程如图 3 和图 4 所示.

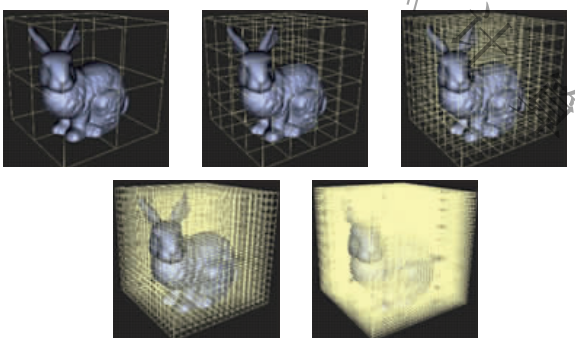

图 3 基于八叉树的体素分割

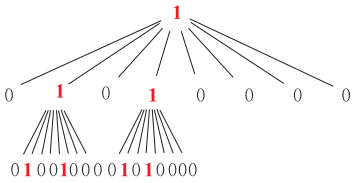

图 4 八叉树数值化表示

通过图 4 所示的二值化编码形式,可以获得每个三维模型的深度八叉树结构.然后将八叉树二值化后的三维模型作为卷积神经网络的输入,进行卷积神经网络的训练及分类操作.

4 卷积神经网络

4.1 网络层数及参数设置

输入层的作用是对要训练的数据进行准备工作,实验中将体素化的模型数据作为卷积神经网络的输入数据,它们的表示形式为 $N\times C\times W_1\times W_2\times W_3$ 固定大小的体素化数据,其中 N 表示输入模型的数量,C 表示通道数,W_1,W_2,W_3 分别表示的是整个体素化模型的 x,y,z 方向体素的数量.实验中设置为 $N=9461$,$W_1=W_2=W_3=100$,为了提高数据准备的效率,本文实验使用 Caffe[23] 框架,它在输入处理上采用的是“预读机制”,即在当前网络使用当前批次数据进行训练时,输入层单独运行一个进程来进行下一批次数据的读取操作.

卷积层的作用是用来接收前面输入层的数据[24],对于输入层的原始输入数据,使用特定卷积核对其进行卷积操作,得到大小为 $N\times C\times d_1\times d_2\times d_3$ 的卷积图,其中 C 在输入层中表示通道的个数,在卷积层中表示卷积核的数量.这里要保证卷积核的数量与通道数一致,同时 d_1,d_2,d_3 表示的是对前一层的输入图进行卷积操作后的卷积图的大小.例如当前卷积层的第 j 个卷积核的参数为 k_j,输入层的第 i 个卷积图为 x_i,因此经过当前这个卷积核进行卷积操作后的卷积,表示为式(10).

$$y_j=f\left(\sum_{i=1}^{C}x_i * k_j+b_j\right) \tag{10}$$

其中,$f(\cdot)$表示激活函数,实验中设置 5 层卷积层 conv1,conv2, conv3,conv4 conv5.采用 ReLu 激活函数.

池化层的主要工作就是对该层的输入数据进行采样[25],然后将采样后的结果作为下一层的输入数据并重新进行卷积操作.在实验中设置 3 层池化层,卷积核大小为 3×3,步长为 2.这样可以将八叉树分割的数据与未进行八叉树分割的数据进行对齐,同时还可以降低数据的规模.一般来说,采用下采样操作来进行最大池化处理,这样能够保证不会漏掉原始三维模型的任何边缘以及顶点内容池化层的结果用采样区域内的一个具有代表性的值来表示.对于任何一个位置的输出层数据 $y(i^*,j^*,k^*)$,都有如式(11)所示.

$$y(i^*,j^*,k^*)=\max\left\{\begin{matrix} i\in\left[(i^*-1)\cdot\left[\frac{a}{2}\right],i^*\cdot\left[\frac{a}{2}\right]\right] \\ x(i,j,k),j\in\left[(j^*-1)\cdot\left[\frac{b}{2}\right],j^*\cdot\left[\frac{b}{2}\right]\right] \\ k\in\left[(k^*-1)\cdot\left[\frac{c}{2}\right],k^*\cdot\left[\frac{c}{2}\right]\right]\end{matrix}\right\} \tag{11}$$

本网络经过 3 个全连接层,输出一个 1000 维的特征向量,本文实验所用的具体网络设置如表 1 所示.

表 1　本文网络结构及参数设置

	kernel_size	*stride*	*pad*	*num_output*	*Act-function*
Conv1	11×11	4	0	96	ReLu
Pool1	3×3	2	—	—	—
Conv2	5×5	1	2	256	ReLu
Pool2	3×3	2	—	—	—
Conv3	3×3	1	1	384	—
Conv4	3×3	1	1	384	ReLu
Conv5	3×3	1	1	256	ReLu
Pool5	3×3	2	—	—	—
Fc6	—	—	—	4096	—
Fc7	—	—	—	4096	—
Fc8	—	—	—	1000	—

4.2　学习率设置策略

由于模型的数据量巨大，在实验中选择对模型进行批量处理，实验设置 *batch_size* 为 128，*epoch* 设置为 100，设置 1000 次迭代. 设置初始学习率为 0.001. 由于本文使用的是 Caffe 深度学习框架，Caffe 框架中学习率机制主要有 fixed、step、inv、multistep、exp 和 ploy 这 6 种. fixed 即固定学习率，在整个优化过程中学习率不变. step 采用均匀降低的方法，每次降低为原来的某倍数. multistep 采用非均匀降低策略，指定降低的 step 间隔，每次降低为原来的一定倍数. exp 是一种指数变化，$new_{lr}=base_{lr}\times(gamma^{iter})$，由公式可知这是连续变化，*gamma* 越大则衰减越慢. inv 也是一种指数变换，参数 *gamma* 控制曲线下降的速率，$new_{lr}=base_{lr}(1+gamma\times iter)^{-power}$. poly 的学习曲率的形状主要由参数 *power* 的值来控制，$new_{lr}=base_{lr}*(1-iter/maxiter)^{power}$. 当 $power=1$ 的时候，学习率曲线为一条直线. 当 $power<1$ 的时候，学习率曲线是凸的，且下降速率由慢到快. 当 $power>1$ 的时候，学习率曲线是凹的，且下降速率由快到慢. 本文用这 6 种方法进行对比实验，观察 6 种方法的检索准确率，结果如图 5 所示.

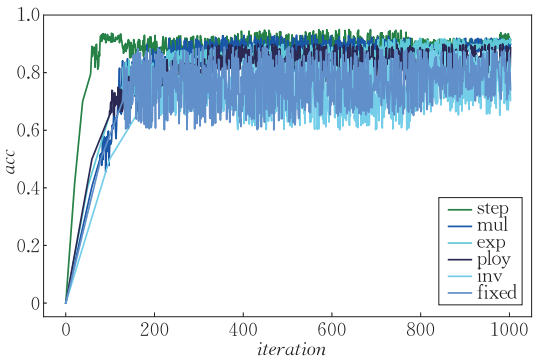

图 5　学习率设置对比图

实验时，初始学习率都设置为 0.001. step 方法中设置 *stepsize* 为 1000，*gamma* 为 0.1，multistep 方法设置 *gamma* 为 0.5，*stepvalue*=200，400，600，800. exp 方法设置 *gamma* 为 0.7，inv 方法设置 *gamma* 为 0.5，*power* 为 0.25. poly 方法设置 *power* 为 0.8. 由图 5 结果可知，step 方法的收敛效果最好，因此在学习率变更时，本文选择 step 方法.

5　实　验

5.1　实验平台

本文的实验主要在 Windows 10 64 位操作系统上进行，设备的配置信息：处理器为两颗 Intel E5-2430 服务器芯片，GPU 使用 NVIDIA GTX 1080Ti，内存 16GB，硬盘为 80G SSD，实验使用 Caffe 深度学习框架，CUDA 版本为 8.0，cuDNN 版本为 5.0.

5.2　实验结果分析

提取三维模型的特征并构造出三维模型的特征向量是计算模型之间的相似性的第一步. 在构造出模型的特征向量之后，通过特征向量之间的距离来表示三维模型之间的相似度，不相似的三维模型之间的距离较大，相似的三维模型之间的距离较小. 根据相似度值的大小对候选的相似模型进行排序，距离最近的也就是匹配度最高的三维模型，这个过程就是三维模型的相似性度量.

检索部分本文采用欧氏距离来进行两个不同模型之间的相似性匹配任务，空间中两个特征向量可以表示为 $\boldsymbol{X}=(x_1,x_2,x_3,\cdots,x_n)$，$\boldsymbol{Y}=(y_1,y_2,y_3,\cdots,y_n)$，用 $D(\boldsymbol{X},\boldsymbol{Y})$ 的值来表示两个特征之间的距离，距离表示为式(12).

$$D(\boldsymbol{X},\boldsymbol{Y})=\sqrt{\sum_{i=0}^{n}(x_i-y_i)^2} \tag{12}$$

对于任意一个查询模型 Q，将其与数据库 M 中的模型进行距离度量，最终得到得匹配模型 Q^*. x_i，$y_j(0\leq i\leq n,0\leq j\leq m)$ 分别代表 Q 和 M 的特征. Q^* 的计算过程可以表示为式(13)、(14).

$$S(Q,M)=\arg\min D(x_i,y_j) \tag{13}$$

$$Q^*=\arg\max_{Q^*\in M} S(Q,M_j) \tag{14}$$

三维模型相似性度量的评价标准是检索的准确度和检索的完整度，相关领域的衡量标准是查准率与查全率.

查准率表示的是在所有被预测为正的样本中实际为正的样本的概率，用 *Precision*[26] 来表示.

查全率表示的是在实际为正的样本中被预测为

正样本的概率，通常用 *Recall*[26] 来表示.

本文在模型训练的过程中使用的是 Princeton Shape Benchmark(PSB)数据集下的 ModelNet40① 模型数据集来进行实验，该数据集包含 40 个大类模型，每个分类下都有 train 集和 test 集，本文通过训练 train 集的模型，再通过 test 数据集进行测试，train 集中大概包括 9461 个模型. 实验中为了验证本文算法的有效性，从 40 个分类中每个分类选择 20 个模型，共 800 个模型，其中每个分类的正样本数与负样本数比例为 3∶1. 实验中使用欧氏距离作为度量三维模型相似性的度量手段. 首先计算待查询模型特征与数据集中的模型特征的欧氏距离，然后通过欧氏距离计算相似度，最终返回检索结果. 最终得到的查全率和查准率之间的关系如图 6 所示.

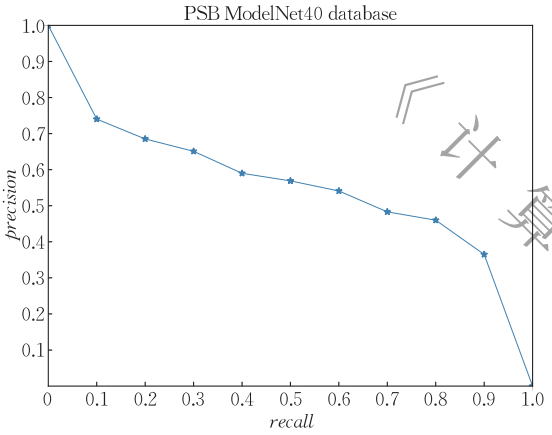

图 6 ModelNet40 下卷积神经网络检索结果

为了评估三维模型检索性能，本文采用了以下流行的标准作为性能指标：

(1) Precision-Recall curve(PR 曲线)是一种综合显示检索性能的曲线. PR 曲线通过改变区分相关度和不相关度的阈值，揭示了查准率与查全率之间的关系.

(2) Area Under the PR curve(AUC)可以作为综合性能指标. AUC 值越高表示性能越好.

(3) Nearest Neighbour(NN)可以用来表示最接近匹配模型的检索精度.

(4) First Tier(FT)表示前 N 个匹配结果的查全率，其中 N 为数据集中相关模型的总数.

(5) Second Tier(ST)表示前 $2N$ 个匹配结果的召回量，其中 N 为相关模型的总数.

(6) F-measure(F)是对检索的查准度和查全率的综合评价的方法.

(7) Discounted Cumulative Gain(DCG)的定义是为排名靠前的结果分配更高的权重，因为用户最有可能使用第一个显示的结果.

(8) Average Normalized Modified Retrieval Rank (ANMRR)是一个综合衡量排名名单的表现. 较低的值反映较高的精度.

为了评估本文算法的性能，将本文的方法与一些先进的方法进行对比，其他对比的算法如下：Nearest Neighbor(NN)：最接近查询所属类别的查询的百分比. Adaptive Views Clustering(AVC)：考虑到并非所有视图都具有同等的重要性，AVC 通过自适应聚类算法选择最优的二维视图进行表示和检索. 该算法采用概率贝叶斯模型来提高性能[27]. Camera Constraint-Free View-based(CCFV)：对于每个查询对象，所有查询视图都聚集在一起以生成视图集群，然后视图集群用于构建查询模型. CCFV 模型是在查询高斯模型的基础上，结合正匹配模型和负匹配模型生成的[28].

实验中用 NN、FT、F 和 ST 四种指标来评估 NN、AVC、CCFV 和本文方法在 ModelNet40 上的性能，在实验中，对于每个查询示例，随机选择 50 个作为正样本进行训练，然后从其他类别随机抽取 10 个样本作为负样本. 结果如图 7 所示.

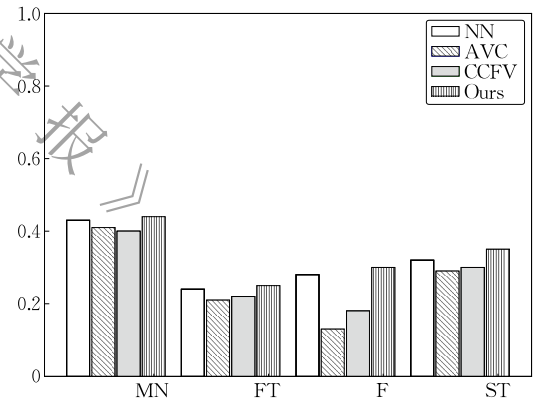

图 7 几种方法的性能对比

图 7 展示了不同标准下的性能，从图 7 可以看出本文的算法在 NN、FT、F、ST 方面都取得了较好的结果，在 NN、FT、F、ST 下与其他三种算法相比分别取得了 1%～4%，1%～5%，2%～17%，2%～6% 的增益. 从图 7 中也可以得出以下结论：(1) CCFV 和 AVC 都可以看做是统计模型，CCFV 利用高斯模型来描述特征分布，AVC 使用贝叶斯模型来描述特征分布，不同的是 AVC 将每个视图视为独立的个体，而 CCFV 采用高斯模型考虑了视图在特征空

① The Princeton ModelNet. http://modelnet.cs.princeton.edu/2015.3

间中的变化.最后的实验结果也证明了 CCFV 优于 AVC；(2) NN 的表现要优于 CCFV 和 AVC 因为 NN 应用了 CNN 的特性，CNN 网络的优点使得 NN 的性能要优于 CCFV 和 AVC；(3) 本文的算法在各种指标下性能都是最优的，因为粗粒度特征和细粒度特征融合能够更完整的表示模型，同时使用八叉树表示法节省了计算的时间以及存储的空间，所以，本文算法的性能优于其它 3 种.

本文算法与 D2 特征提取的三维模型检索算法、MODFD 特征提取的三维模型检索算法、AAD 特征提取的三维模型检索算法、3D Shape Net 检索算法、Multi-View CNN(MVCNN)检索算法[29] 和 Volumetric CNN(VolCNN)[30] 检索算法和 VoxNet[31] 在 ModelNet40 数据集上进行对比. ModelNet40 数据集含有 40 个大类，将每个类别中的模型从训练集选出 100 个模型进行训练，从测试集中选出 20 个模型进行测试，对于每个被检索的模型给出 5 个检索结果，求平均值，该平均值即为检索正确率.

表 2 中对比实验一共有 8 种，D2 算法、MODFD 算法、AAD 算法、3D Shape Net 算法、MVCNN 算法、VolCNN 算法、VoxNet 算法和本文算法. 由表 2 可知，前 3 种基于传统的检索方法表现效果较差. MVCNN 算法提出了一种新颖的 CNN 结构，它将三维形状的多个视图的信息组合成一个紧凑的形状描述符，利用 CNN 网络的优势在实验中取得了 80.2%的检索准确率. VolCNN 算法是通过与多视图相结合卷积神经网络并使用带有多方向池的 3DCNN 来获取形状表征，实验中达到了 79.53%的正确率. 3D Shape Net 是一种较好的基于 3D 卷积神经网络对体素化模型进行分类的算法，该算法在将三维模型进行体素化处理后，再选择多个观测视点对体素模型进行二维视图化，之后将多视点视图构造成卷积网络的输入数据，最后将视图特征向量与数据集中三维模型进行训练后输出的特征向量进行对比. 该算法的识别正确率在 ModelNet40 上达到了 71.0%. VoxNet 算法是基于模型特征的检索方法，将三维网络模型转化为体素模型，用 32×32×32 的体素模型进行实验，使用密集的数组来执行所有的 CNN 处理. 实验中达到 83.0%的准确率. 但是在三维空间并不是所有的体素都含有模型信息，所以对所有的体素进行训练计算是得不偿失的. 本文的基于八叉树结构的三维体素模型检索算法利用八叉树结构自适应的空间剖分来压缩存储，构造的是完整的三维模型体素数据，包含着模型外表面的所有细节. 进行二值化处理能够巧妙地让体素数据作为卷积神经网络的输入数据，经过卷积神经网络的处理后得到 1000 维的特征向量，最后计算特征向量间的相似性实现模型检索工作. 该方法的识别准确率达到了 88.7%高于前面提到的 7 种方法.

表 2　不同算法在 ModelNet40 数据集上的实验结果对比

序号	特征提取算法名称	测试集 ModelNet40 检索正确率/%
1	D2	68.87
2	MODFD	65.18
3	AAD	64.63
4	3D Shape Net	71.0
5	MVCNN	80.2
6	VolCNN	79.53
7	VoxNet	83.0
8	Ours	88.7

为了进一步验证本文算法的鲁棒性，通过进行实验后得到了表 2 中对应的 8 种特征提取算法识别结果对应的受试者工作特征曲线(Receiver Operating Characteristic Curve，ROC). ROC 曲线以假正率 *FPR*(False Positive Rate)为横轴，以真正率 *TPR*(True Positive Rate)为纵轴，描绘出正样本正确识别概率随负样本误识别成正样本概率的变化趋势. 在 ROC 曲线中，曲线上的点越靠近左上角说明真正率越高、假正率越低，算法区分能力越强，即提取的个体差异信息越明显.

实验中，首先将用于测试的 40 类三维模型数据集组合成正样本对(同一类样本对)与负样本对(不同类样本对)序列. 根据不同算法提取特征并计算样本对距离，通过固定负样本对的误识别率即 *FPR*，选出对应的距离阈值，然后以同样的阈值检测正样本对的接受率. 如图 8 为通过表 2 中 8 种不同特征提取算法分别选定 13 个(*FPR*，*TPR*)点拟合得到的 ROC 曲线图.

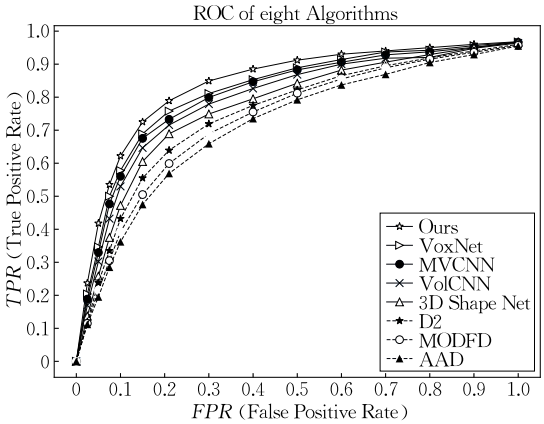

图 8　8 种特征提取算法在三维模型数据集的实验结果的 ROC 图

图 8 实验结果显示:本文的基于深度学习的三维体素模型检索算法对应的 ROC 曲线位于所有曲线的最左、最上的位置,即 FPR 一定时,该算法下得到 TPR 最高,同时在 TPR 一定时,该算法下对应的 FPR 最低.这说明本文的特征提取算法相较于其它 7 种算法有明显优势,也体现出本文算法的可行性和有效性.

6 总 结

本文从基于模型特征的角度进行三维模型检索工作,提出了一种基于八叉树结构的三维体素模型检索方法.具体实现为将三维网格模型进行体素化处理,提取三维模型的粗粒度特征和细粒度特征进行融合,以八叉树结构来存储模型.本文提出了一种特殊的模型体素化方式:八叉树分割法,该方法对包含模型的体素置 1 进行迭代分割,对不包含模型的体素置 0,经过八等分细化迭代处理完成三维模型的数值化表达.然后将其作为卷积神经网络的输入数据,实现对网络模型的训练和后续的分类预测.最终对网络提取出的模型特征向量,进行欧氏距离度量计算模型间的相似性,从而完成模型的检索工作.本文方法与其他算法相比在 ModelNet40 数据集中的识别准确率可以达到 88.7%,充分体现了本文方法的优越性和有效性.

致 谢 在此感谢审稿人对本文提出的宝贵意见!

参 考 文 献

[1] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature, 2015, 521(7553): 436-444

[2] Babenko Y A, Lempitsky V. Aggregating local deep features for image retrieval//Proceedings of the IEEE International Conference on Computer Vision. Los Alamitos, USA, 2015: 1269-1277

[3] Wan J, Wang D, Hoi S C H, et al. Deep learning for content-based image retrieval: a comprehensive study//Proceedings of the 2014 ACM Conference on Multimedia (MM). Florida, USA, 2014: 157-166

[4] Wu N Z, Song S, Khosla A, et al. 3D ShapeNets: A deep representation for volumetric shapes//Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Boston, USA, 2015: 1912-1920

[5] Bai Liu, Song Chao-Chao. 3D model retrieval based on CGS and genetic algorithm. Journal of Graphics, 2016, 37(6): 754-758(in Chinese)
(白柳, 宋超超. 基于体素构造和遗传算法的三维模型检索. 图学学报, 2016, 37(6): 754-758)

[6] Lu W, Zhang X, Liu Y. L_1-medial skeleton-based 3D point cloud model retrieval. Multimedia Tools and Applications, 2019, 78(1): 479-488

[7] Li B, Lu Y, Li C, et al. SHREC'14 track: Extended large scale sketch-based 3D shape retrieval//Proceedings of the Eurographics Workshop on 3D Object Retrieval. Switzerland, 2014: 121-130

[8] Wang F, Kang L, Li Y. Sketch-based 3D shape retrieval using convolutional neural networks//Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Boston, USA, 2015: 1875-1883

[9] Liu A A, Nie W Z, Su Y T. 3D object retrieval based on multi-view latent variable model. IEEE Transactions on Circuits and Systems for Video Technology, 2019, 29(3): 868-880

[10] Liu A A, Shi Y, Nie W Z, et al. View-based 3D model retrieval via supervised multi-view feature learning. Multimedia Tools and Applications, 2018, 77(3): 3229-3243

[11] Wu J J, Wang J L, Xie B. A content-based 3D shape retrieval system for voxel models//Proceedings of the International Symposium on Signal Processing Biomedical Engineering, and Informatics (SPBEI). Hangzhou, China, 2014: 582-588

[12] Zhao Fang-Lei, Jing Shi-Kai, Li Xiang-Qian, et al. Triangular mesh model surface voxelization algorithm based on triangle subdivision. Computer Integrated Manufacturing Systems, 2017, 23(11): 2399-2406(in Chinese)
(赵芳垒, 敬石开, 李向前等. 基于三角形细分的三角网格模型表面体素化算法. 计算机集成制造系统, 2017, 23(11): 2399-2406)

[13] Miyagi R, Aono M. Sliced voxel representations with LSTM and CNN for 3D shape recognition//Proceedings of the 9th Annual Summit and Conference of the Asia-Pacific-Signal-and-Information-Processing-Association (APSIPA ASC). Kuala Lumpur, Malaysia, 2017: 320-323

[14] Cheng H C, Lo C H, Chu C H, Kim Y S. Shape similarity measurement for 3D mechanical part using D2 shape distribution and negative feature decomposition. Computers in Industry, 2010, 62(3): 269-280

[15] Osada R, Funkhouser T, Chazelle B, et al. Shape distributions. ACM Transactions on Graphics, 2002, 21(4): 807-832

[16] Zhou Kun, Gong Minmin, Huang Xin, Guo Baining. Data-parallel octrees for surface reconstruction. IEEE Transactions on Visualization and Computer Graphics, 2011, 17(5): 669-681

[17] Liu Z M, Chen Y Y, Hidayati S, et al. 3D model retrieval based on deep autoencoder neural networks//Proceedings of the 1st International Conference on Signals and Systems (ICSigSys). Bali, Indonesia, 2017: 290-296

[18] Perdomo O, Otálora S, González F A, et al. Oct-Net: A convolutional network for automatic classification of normal and diabetic macular edema using SD-OCT volumes//Proceedings

of the 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018). Washington, USA, 2018: 1423-1426

[19] Riegler G, Ulusoy A O, Geiger A. OctNet: Learning deep 3D representations at high resolutions//Proceedings of the 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2017). Honolulu, USA, 2017: 6620-6629

[20] Li X X, Cao Q, Wei S. 3D object retrieval based on multi-view convolutional neural networks. Multimedia Tools and Applications, 2017, 76(19): 20111-20124

[21] Wang Peng-Shuai, Sun Cshun-Yu, Liu Yang, Tong Xin. Adaptive O-CNN: A patch-based deep representation of 3D shapes analysis. ACM Transactions on Graphics (SIGGRAPH Asia), 2018, 37(6): 1-11

[22] Eitz M, Hays J, Alexa M. How do humans sketch objects?. ACM Transactions on Graphics, 2012, 31(4): 1-10

[23] Je S, Nguyen H H, Lee J. Image recognition method using modular systems//Proceedings of the International Conference on Computational Science & Computational Intelligence. Las Vegas, USA, 2015: 504-508

[24] Nie W, Xiang S, Liu A. Multi-scale CNNs for 3D model retrieval. Multimedia Tools & Applications, 2018, 77(17): 22953-22963

[25] Yang Z X, Tang L, Zhang K, et al. Multi-view CNN feature aggregation with ELM auto-encoder for 3D shape recognition. Cognitive Computation, 2018, 10(6): 908-921

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

[27] Ansary T F, Daoudi M, Vandeborre J. A Bayesian 3-D search engine using adaptive views clustering. IEEE Transactions on Multimedia, 2007, 9(1): 78-88

[28] Gao Y, Tang J, Hong R, et al. Camera constraint-free view-based 3-D object retrieval. IEEE Transactions on Image Processing, 2012, 21(4): 2269-2281

[29] Su H, Maji S, Kalogerakis E, et al. Multi-view convolutional neural networks for 3D shape recognition//Proceedings of the IEEE International Conference on Computer Vision. Santiago, Chile, 2015: 945-953

[30] Qi C R, Hao S, Niessner M, et al. Volumetric and multi-view CNNs for object classification on 3D data//Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Seattle, USA, 2016: 5648-5656

[31] Maturana D, Scherer S. VoxNet: A 3D convolutional neural network for real-time object recognition//Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Hamburg, Germany, 2015: 922-928

ZHANG Man-Dun, Ph. D., associate professor. His research interests include computer graphics, image processing, 3D model retrieval and 3D object recognition.

YAN Ming-Xiao, M. S. candidate. Her research interest is 3D model retrieval.

MA Ying-Shi, M. S. His research interest is 3D model retrieval.

WANG Hong, M. S. candidate. Her research interest is 3D model retrieval.

LIU Wei, Ph. D., associate professor. His research interests include precision calibration, graphic image processing and virtual reality.

HUANG Xiang-Sheng, Ph. D., associate researcher. His research interests include artificial intelligence, machine learning, situational awareness and decision making and self-learning.

Background

In this paper, we research on 3D voxel model retrieval based on octree structure. With the development of VR/AR technology and the wider 3D applications, it is realized that 3D model retrieval is becoming more and more important. At present, the realization of the 3D retrieval problem can be divided into two aspects from characteristics: View-based and Model-based.

View-based retrieval methods proposed several visual descriptors, such as light field descriptors (LFDs), elevation descriptors (EDs), visual feature packages (BoVF) and compact multi-view descriptions (CMVDs). The training vector machine is then performed on different classifier models based on the extracted features, such as nearest neighbor (NN), linear support, and so on. Based on the 3D retrieval of the view, the 3D model is transformed into a 2D image by dimensionality reduction, and feature extraction and feature similarity measurement are performed on the level of 2D.

Model-based retrieval directly affects the original 3D representation of the object, such as polygon mesh, voxel-based discretization, point cloud or implicit surface. Compared with

view-based retrieval, the original data of the model extracting a higher level of feature representation can be better preserved by model-based. Voxelization can be considered as the simplest and most intuitive discrete method to transform a 3D surface into a regular structure, fully retaining the geometry of the 3D surface. However, model-based feature representations are often accompanied by problems with high storage and high computation.

In order to solve the problem of high storage, this paper proposes an octree-based structure as the input of the convolutional neural network, voxel representation of the 3D model, greatly reducing the time overhead. Voxelization model after segmentation is numerically represented by 0 and 1. The data is stored in the octree structure, then the model is subjected to five iterations of the segmentation, finally the voxelized representation result of the model can be obtained. The model's voxelization representation results are input into the convolutional neural network result SOFTMAX cost function for model training. The results show the method proposed by this paper is better than other similar retrieval methods.

Of course, this article still has a lot of room for improvement. Using more and more data sets to verify the algorithm, improving the model's amount of storage structure to save storage space and computing time is the focus of the next step.

This article was funded by the National Natural Science Foundation of China (61573356), the Tianjin Enterprise Science and Technology Commissioner Program (18JCTPJC58700), and the Natural Science Foundation of Hebei Province (F2019202054).