

深度聚类索引下的海量地震数据快速三维可视化

汤文琳^{1,2,3}, 谢 凯^{1,2,3}, 文 畅^{3,4}, 贺建彪⁵

(1. 长江大学 电子信息学院, 湖北 荆州 434023; 2. 长江大学 电工电子国家级实验教学示范中心, 湖北 荆州 434023;

3. 长江大学 西部研究院, 新疆 克拉玛依 834000; 4. 长江大学 计算机科学学院, 湖北 荆州 434023;

5. 中南大学 计算机学院, 长沙 410083)

摘要: 地震数据的三维可视化能够直观反映地质的相关结构信息, 为地质勘探等研究提供数据支持。针对传统体绘制算法在集中载入海量数据时存在显示延迟、画面跳跃、卡顿等问题, 提出一种快速三维可视化算法。使用变分自编码器和深度聚类学习数据的空间特征表示, 通过迭代优化目标函数提高聚类性能, 以解决因空间数据分布不均造成的节点重叠问题。建立高效的索引结构, 提高数据实时读取的效率, 通过时序卷积网络预测下一个视点位置, 提前将潜在数据载入内存, 避免因海量数据集中加载导致画面卡顿、跳跃。使用基于双层视锥体的视点动态划分调度模型, 剔除不必要的绘制节点及减轻系统负荷, 从而提高数据渲染速度和流畅度。实验结果表明, 该算法在索引结构上查询数据块的时间相比希尔伯特R树算法减少了64.14%~66.37%, 预测视点的正确率相比拉格朗日插值算法提高了12.08%~22.70%, 实时帧率在较大规模的子集上也能够相对稳定平滑, 在保证图像质量的前提下整体系统的渲染性能达到预期效果。

关键词: 三维可视化; 深度学习; 深度聚类; 希尔伯特R树; 时序轨迹预测; 视锥体裁剪

开放科学(资源服务)标志码(OSID):



中文引用格式: 汤文琳, 谢凯, 文畅, 等. 深度聚类索引下的海量地震数据快速三维可视化[J]. 计算机工程, 2022, 48(11): 275-283.

英文引用格式: TANG W L, XIE K, WANG C, et al. Fast 3D visualization of massive seismic data in deep clustering index[J]. Computer Engineering, 2022, 48(11): 275-283.

Fast 3D Visualization of Massive Seismic Data in Deep Clustering Index

TANG Wenlin^{1,2,3}, XIE Kai^{1,2,3}, WEN Chang^{3,4}, HE Jianbiao⁵

(1. School of Electronic Information, Yangtze University, Jingzhou, Hubei 434023, China;

2. National Electrical and Electronic Experimental Teaching Demonstration Center, Yangtze University, Jingzhou, Hubei 434023,

China; 3. Western Research Institute, Yangtze University, Karamay, Xinjiang 834000, China; 4. School of Computer Science,

Yangtze University, Jingzhou, Hubei 434023, China; 5. College of Information Science and Engineering,

Central South University, Changsha 410083, China)

[Abstract] The 3D visualization technology of seismic data can intuitively analyze geological structure information, which can provide data support for geological exploration and other research. To solve the problems of display delay and image jump lag when loading large scale data in traditional volume-rendering algorithms, this study proposes a fast 3D visualization algorithm of massive seismic data based on deep clustering index. Firstly, the spatial feature representation of data is learned by variational autoencoding and deep clustering; then, the clustering performance is improved by iterative optimization of objective function. Thus, the problem of overlapping nodes caused by uneven spatial data distribution is solved, and an efficient index structure is established to improve the efficiency of real-time data reading. Additionally, a temporal convolutional network is used to predict the position of the next viewpoint, and the potential data is loaded into memory in advance to avoid the problem of picture-stuck jump caused by centralized loading when the data amount is too large. Finally, combined with the viewpoint dynamic-division scheduling model based on double-layer visual cone, redundant nodes are removed to reduce the system load, and improve the data-rendering speed and fluency. The experimental results show that, compared with the Hilbert R-Tree (HRT), the time of querying data sub-blocks on the index structure of algorithm in this paper is reduced by 64.14% ~ 66.37%, and the accuracy of predicting viewpoints is improved by 12.08% ~ 22.70%, compared with the Lagrange

基金项目: 新疆维吾尔自治区自然科学基金项目(2020D01A131); 湖北省教育厅项目(B2019039)。

作者简介: 汤文琳(1997—), 女, 硕士研究生, 主研方向为图形图像处理、人工智能; 谢 凯(通信作者), 教授、博士; 文 畅, 讲师、硕士; 贺建彪, 副教授。

收稿日期: 2021-11-01 修回日期: 2021-12-30 E-mail: pami2009@163.com

interpolation algorithm. The real-time frame rate can also be relatively stable and smooth on a large subset. The rendering performance of the whole system achieves the desired effect under the premise of ensuring the image quality.

[Key words] 3D visualization; deep learning; deep clustering; Hilbert R-Tree (HRT); sequence trajectory prediction forecasting; view frustum culling

DOI: 10.19678/j.issn.1000-3428.0063095

0 概述

三维可视化技术在计算机图形学的发展过程中起重要作用,是对数据对象进行多方位呈现和分析的有效方法,已经渗透到医学^[1-2]、海洋^[3]、地质^[4]、油气勘探等领域。三维可视化技术中的体绘制算法^[5-6]已应用在地质勘探方面,能够清晰地描绘地质体内部层次细节信息和特征,为相关研究人员提供数据支持。

目前,三维可视化技术的发展面临诸多问题:一方面,传统体绘制算法复杂度较高,需要较大的内存空间,对计算机硬件的要求较高;另一方面,对于一定规模的体数据而言,传统算法的计算速度较慢,在进行交互显示时容易出现画面延时、波动卡顿等问题。因此很多三维可视化方案均存在一定局限性。

为此,研究人员提出了诸多相关的优化方案。文献[7-8]提出基于硬件方面的体绘制算法,并在传统光线投射算法和三维纹理映射算法的基础上使用硬件设备提高渲染性能。文献[9]针对海量点云数据索引时间过长且内存占用过大的问题,提出使用八叉树(Octre, OCT)重新组织数据并分块保存,但八叉树最小叶节点的确定较困难。文献[10]针对大量的地理空间数据,使用并行框架构建R树及其变体,以减少索引更新的时间,但R树本身结构导致存在部分数据节点空间的重叠浪费问题。文献[11]考虑构造R树的最小外接矩形框的影响,采用动态K-means聚类算法,在R树的基础上对结构进行优化,从而提高算法的检索速度。文献[12]将数据按照时空周期划分,结合动态K值的聚类算法构建希尔伯特R树(Hilbert R-Tree, HRT)结构,通过聚类和空间填充曲线在一定程度上降低节点之间的覆盖交叠,能很好地支持海量时空特征数据的索引。

除了从数据结构的角度解决快速可视化的问题,还有一些方案从预测视点的运动轨迹角度出发,结合调度加载算法解决当数据规模较大时出现的显示卡顿问题。文献[13]基于拉格朗日插值算法预测视点的运动轨迹,通过设置不同的插值步长决定最终预测的正确率和渲染效果,但其预测误差会随着阶次增加而增大。文献[14]在视点运动的预测上应用堆叠的长短时记忆(Long Short Term Memory, LSTM)网络,相比传统的估算方法预测精度有所提升。文献[15-16]基于深度学习模型对相关时间轨迹数据进行预测,得到较好的预测效果。文献[17-18]基于可见性判断的角度,通过结合视锥体裁剪算法加速绘制大规模数据,提高交互性能。文献[19-20]除了在数据结构上直接改善索引效率,还结合了动态调度的思想来提高数据加载速度。

本文提出一种面向海量地震数据的快速三维可视化方法,在HRT结构的基础上使用深度聚类优化结构,改善因空间数据分布不均造成的节点重叠问题,以提高索引查询效率。通过时序卷积模型提前预测下一视点位置,动态划分可视区域,并提前将潜

在区域的数据加载到内存中,避免因集中加载引起画面卡顿。此外,在不影响图像质量的前提下提前剔除不必要的节点,避免系统资源浪费,从而加快渲染速度。

1 本文算法框架

本文可视化算法的总体框架结构如图1所示,主要分为3个模块,包括建立高效的数据索引结构(a)、视点运动轨迹的预测(b)以及体数据的调度加载(c)。

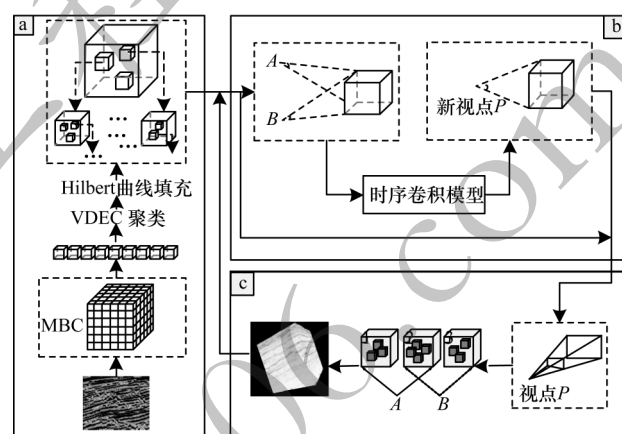


图1 本文算法的总体框架

Fig.1 General framework of algorithm in this paper

由图1可知,本文算法首先读取地震数据文件,将原地震数据格式映射到空间的三维结构中,并根据计算机内存划分最小边界立方体(Minimum Bounding Cube, MBC)数据块,在提取数据特征时利用变分深度嵌入聚类(Variational Deep Embedding Clustering, VDEC)完成聚类操作,将聚类之后的数据子块的中心点经过Hilbert曲线进行降维转换,然后根据节点生成时间和内存重组R树的结构。当数据进入下一模块时,确定当前视点位置坐标,此过程有两条分支:一是使用视锥体模型对当前视点位置对应的可见空间数据进行剔除并渲染显示;二是根据当前视点位置利用时序卷积网络模型预测未来视点的位置,通过比较视点动态划分模型,应用双层视锥体裁剪算法得到预测视点所涉及到的数据范围,对潜在数据进行加载渲染,并根据索引结构确定新的视点坐标,绘制下一帧图像。

2 深度聚类与数据结构融合算法

2.1 数据的索引结构

针对因数据规模的扩大导致的索引结构复杂、效率过低等问题,仅靠改进硬件远远不够,还需改进算法的软件,才能高效解决海量地震数据的快速三维可视化问题。

本文算法的索引模块采用HRT结构,其思想是采用Hilbert空间填充曲线,将无序数据按照特定方

式连接,令曲线穿过高维空间中的数据,对数据的位置坐标进行编码排序,将高维空间的数据拉直转换成一维空间的数据,然后用最小边界矩形将相邻元素框住,节点越往上,框住空间越大,从而形成HRT结构。

虽然HRT结构有相对较高的存储利用率,但由于其被应用在大型体数据时,若数据分布不均,则容易出现节点空间重叠问题,即空间上数据对象相近,但经过空间曲线转换后的数据值所属节点空间不相邻。对于大规模的数据集,可以通过合理的聚类算法将空间特征相似的数据聚集在一起,避免后续码值转换后数据错分的情况发生,以此来提高结构的存储利用率,实现快速索引。

2.2 变分深度嵌入聚类VDEC

为优化索引技术,本文使用变分深度嵌入聚类VDEC算法,相比传统的聚类算法而言,VDEC算法使用变分自编码器VAE^[21],采用无监督的算法反映输入数据分布的本质特征,并通过深度嵌入聚类(Deep Embedding Clustering, DEC)^[22]学习数据潜在隐变量空间的特征表示和聚类分配,以迭代优化目标,提升聚类性能。

变分自编码器VAE主要通过无监督算法将隐变量空间的数据特征重构输出。VAE的网络结构包括编码器和解码器。包含地震特征的数据集 $X=\{x_i\}_{i=1}^N$ 经过隐变量分布 $p(z)$ 生成隐变量空间表示 z ,再通过 $p(x|z)$ 重构生成 \hat{x} 。其中编码器和解码器的权重和偏差模型分别表示为 ϕ 、 θ 。根据贝叶斯公式可以得到 $p(z|x)$,其表达式如式(1)所示:

$$p(z|x) = \frac{p(z)p(x|z)}{p(x)} \quad (1)$$

由于后验分布计算困难,因此通过引入近似后验分布 $q_\phi(z|x)$ 逼近真实的后验分布 $p(z|x)$ 。KL散度定义为:

$$D_{KL}(q_\phi(z|x)||p(z|x)) = E[\log_a(q_\phi(z|x) - p(z|x))] = \log_a p(x) + E[\log_a q_\phi(z|x) - \log_a p(x|z) - \log_a p(z)] \Rightarrow \log_a p(x) \geq E[\log_a p_\theta(x|z) - D_{KL}(q_\phi(z|x)||p(z|x))] \quad (2)$$

采用证据下界ELBO为变分自编码器的损失函数,其表达式如下:

$$L_{ELBO} = E_{x \sim q_\phi(z|x)}[\log_a p_\theta(x|z)] - D_{KL}(q_\phi(z|x)||p(z)) \quad (3)$$

设 z 服从正态分布:

$$p(z) \sim N(z; 0, I) \quad (4)$$

变分自编码网络VAE在隐藏层输出的参数有2个维度,即均值 μ 和方差 σ^2 :

$$q_\phi(z|x) \sim N(z; \mu, \sigma^2 I) \quad (5)$$

从近似后验分布 $q_\phi(z|x)$ 重采样得到 z :

$$z = \mu + \sigma \times \varepsilon, \varepsilon \sim N(0, 1) \quad (6)$$

将采样得到的 z 输入至生成模型 $p_\theta(x|z)$ 中,生成新的 \hat{x} ,取 $q_\phi(z|x)$ 与先验分布 $p(z)$ 之间的KL散度作为编码模型的损失:

$$L_{q_\phi} = K_{KL}(N(z; \mu, \sigma^2 I)||N(z; 0, I)) = \frac{1}{2} \sum_{j=1}^J (1 - \log_a(\sigma^2) + (\mu_j^2) + (\sigma_j^2)) \quad (7)$$

对于生成模型,定义解码器的重构误差为其损失:

$$L_{p_\theta} = \|x_i - \hat{x}\|^2 \quad (8)$$

取经过训练后的自编码器编码部分作为学习到的信息特征 z ,比较其与聚类中心的相似度,其中相似度及目标分布的定义如下:

$$q_{ij} = \frac{(1 + \|z_i - c_j\|^2)^{-1}}{\sum_j (1 + \|z_i - c_j\|^2)^{-1}} \quad (9)$$

$$p_{ij} = \frac{q_{ij}^2 / \sum_i q_{ij}}{\sum_j q_{ij}^2 / \sum_i q_{ij}} \quad (10)$$

其中: j 为隐变量空间的维度。

另外定义目标变量 P 与相似度变量 Q 的损失函数的表达式如下:

$$L_c = K_{KL}(P||Q) = \sum_i \sum_j p_{ij} \lg \frac{p_{ij}}{q_{ij}} \quad (11)$$

根据特征表示信息特征 z 与聚类中心之间的相似度表达式(9)得到数据 x 的聚类标签,如式(12)所示:

$$S_i = \arg \max_j q_{ij} \quad (12)$$

用梯度下降法迭代优化总体的目标函数,表达式如式(13)所示:

$$L = L_{p_\theta} + L_{q_\phi} + \alpha L_c \quad (13)$$

其中: α 为权重参数。

2.3 VDEC-HRT算法结构

变分深度嵌入聚类融合HRT树(下文简称为VDEC-HRT)的具体算法如下:

算法1 VDEC-HRT算法

输入 数据集 X ,停止迭代阈值 δ

输出 编码、解码器权重 ϕ 、 θ ,新的聚类中心 c

1.等量划分目标数据的最小边界立方体

2.利用特征数据集 X 和式(7)、式(8)预训练VAE

3.K-means算法动态初始化聚类中心

4.根据式(5)计算隐变量空间 z

5.根据式(9)和式(10),更新目标分布

6.根据式(12)更新聚类中心的原始标记

7.for L in $1, 2, \dots, T$

8.采用梯度下降法优化式(13)

9.if $\frac{O(S_{old} \neq S)}{N} < \delta$

10.return ϕ 、 θ 、 c

11.end

12.end

13.for i in $1, 2, \dots, n$

14.使用曲线 $m = \sum_{i=1}^n 8^{i-1} \times k_i$ 填充MBC块

15. 转换MBC中心点center $C_p = \left(\sum_{i=1}^n \frac{x_i}{n}, \sum_{i=1}^n \frac{y_i}{n}, \sum_{i=1}^n \frac{z_i}{n} \right)$ 对

应Hilbert值

16.end

17.MBC中心码值重新排序 $L_{MBC}(m_1, m_2, \dots, m_n)$

18.if $SIZE_{current\ cluster} \leq MAX_{node-HRT}$

19.当前类别的所有要素视为当前层节点

20.else

21.根据节点生成时间,从底至上构建索引结构

3 视点运动轨迹的预测

少量数据可以直接加载到内存显示,但当数据规模增大时,则需要将数据分块动态地加载和释放,在动态掉队加载之前根据视点运动轨迹,提前将数据加载到内存等待渲染,以避免大规模数据集集中载入引起渲染画面延迟和卡顿。

本文视点预测模块根据持续变化的视点位置和视角预测未来视点的预期位置,并采用时间序列卷积网络(Temporal Convolutional Network, TCN)^[23]进行辅助。TCN网络相比RNN等网络,具有并行性高、梯度稳定、感受野灵敏等特点。

本文数据集中的每个点代表不同时刻连续运动轨迹的视点坐标, T_c 时刻的坐标为 $p_c(u_c, v_c, w_c)$, 其中 u, v, w 代表不同方向的维度。将当前时刻以及之前时刻的坐标序列 $X=(p_1, p_2, \dots, p_c)$ 作为模型的输入, 令卷积核 $F=(f_1, f_2, \dots, f_k)$, 其中 k 为卷积核的大小。将当前以及历史时刻的坐标点构成的坐标序列 $X=(p_1, p_2, \dots, p_c)$ 输入如图2所示的因果卷积结构, 该结构为单向的时间约束结构, 即当前时刻 c 的卷积操作仅在历史时刻 $c-1$ 及之前的信息上。 p_c 处的因果卷积输出为:

$$F(p_c) = \sum_{k=1}^K f_k p_{c-K+k} \quad (14)$$

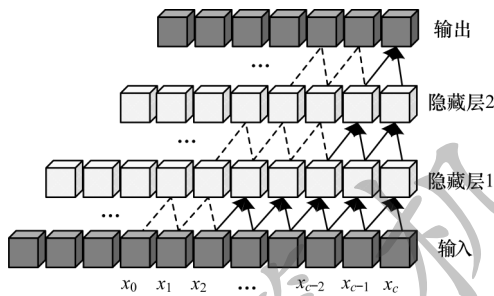


图2 因果卷积结构

Fig.2 Structure of causal convolution

之后将 p_c 处的因果卷积输出 $F(p_c)$ 输入如图3所示的膨胀卷积结构中, 其中 d 为空洞系数。

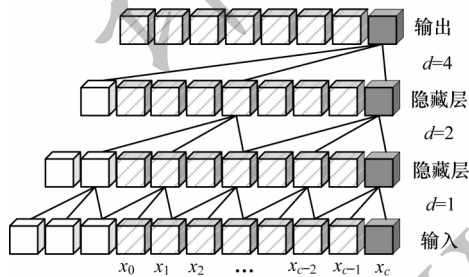


图3 膨胀卷积结构

Fig.3 Structure of expansion convolution

经过膨胀结构的输出公式为:

$$F_d(p_c) = \sum_{k=1}^K f_k p_{c-(K-k)d} \quad (15)$$

使用正则化规范隐藏层的输入, 并在正则化之后引入 Relu 函数:

$$F_o(F_d) = \max(0, F_d) \quad (16)$$

在经过图2和图3的结构之后, 引入 Dropout 函数简化网络, 从而避免过拟合:

$$F'_o \sim \text{Dropout}(F_o(F_d)) \quad (17)$$

为获得长时间序列信息, 且由于加上膨胀卷积后, 网络模型加深会影响其稳定性, 因此本文在网络结构中加入恒等映射, 以增加网络稳定度, 最终输出结果为:

$$O = F'_o(p_c) + p_c \quad (18)$$

将每2个因果膨胀卷积基础层、归一化线性单元块与恒等映射一起作为一个残差模块, 令网络模型以跨层方式传递信息。整体时序卷积网络的结构如图4所示。

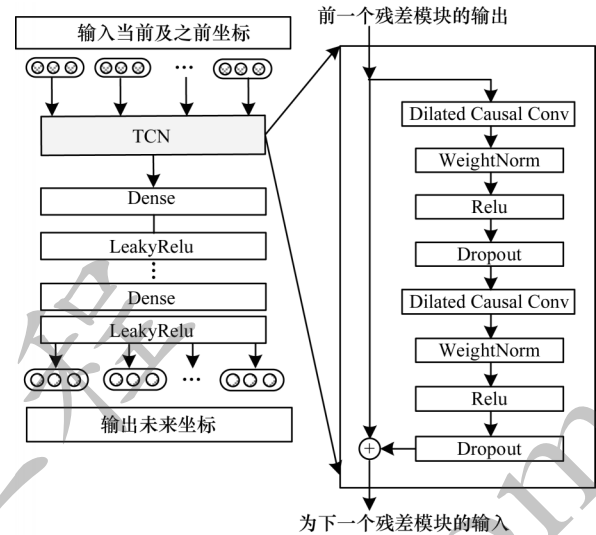


图4 时序卷积网络的结构

Fig.4 Structure of sequential convolutional network

最后输出结果的表达式如式(19)所示:

$$O_n = \text{Activation}(F'_o(p_c) + p_c) \quad (19)$$

时序卷积网络模型可以选择是否由残差模块堆叠起深度网络, 且前一个残差模块的输出可以作为下一个残差模块的输入, 以此加深网络层数, 提取本质特征。输出特征在最后使用全连接层, 并添加 LeakyRelu 修正线性单元, 输出预测视点的坐标位置。

4 体数据的调度加载

随着数据量的增大, 在加载过程中需绘制的对象数量也不断增加。可见性判断方法在不影响渲染质量的前提下, 在加载大规模数据时能够减少系统负荷, 加快图像渲染的速度。其中, 视锥体裁剪方法是可见性判断方法之一, 其操作开销相对较小, 且易于修改, 能够快速准确地加载图形对象。

将原数据等量划分成最小边界立方体子块, 之后用 VDEC-HRT 算法对原本的地震数据结构进行重组, 这部分的数据调度加载利用视锥体对每个 MBC 子块进行空间位置关系判断, 并在绘制阶段将不在范围内的节点裁剪, 从而缩短结构中对象遍历的时间, 提高渲染速度。

本文采用的双层视锥体裁剪算法的具体流程如下:

步骤1 采用粗裁剪算法, 将视锥体简化为圆锥体, 判断圆锥体和空间对象的位置关系, 减少整体判断次数以提高裁剪效率。粗裁剪算法流程如下:

1) 将视锥体六面体简化为一个简易圆锥, 其剖面图如图5所示, 其中 A 为某一 MBC 块的中心坐标, C 为视轴上某点, AC 垂直于圆锥面于点 B, d 为 MBC 块中心到任意顶点的距离。从视点位置作最小圆锥包含视锥体, 使用简易圆锥检查其与最小边界立方体 MBC 之间的位置关系。

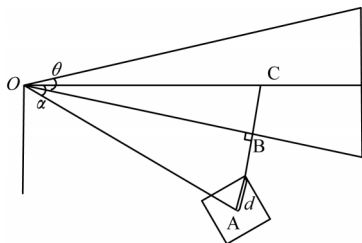


图5 MBC块与简易圆锥的位置剖面图

Fig.5 Position profile of MBC block and simple cone

2) 如果某一 MBC 块的中心 $A(x, y, z) \notin \text{cone}$ 不在圆锥内部, 且 $AB \geq d$, 则判断 MBC 块在圆锥的外部, 对该 MBC 块进行裁剪, 算法结束, 否则继续。

3) 如果 $\left(d + z \sin \frac{\theta}{2}\right)^2 \leq (x^2 + y^2) \cos^2 \left(\frac{\theta}{2}\right)$, 则判断 MBC 块在圆锥外部, 裁剪该 MBC 块包含的空间对象, 否则判断 MBC 块与圆锥相交, 可继续使用精细裁剪算法对其进一步裁剪, 算法结束。

步骤2 精细裁剪算法在粗裁剪的基础上进一步使用标准视锥体截棱锥对空间对象进行精细裁剪, 提高裁剪的精确性。

使用标准视锥体进行裁剪时, 先使用数据的 MBC 块对视锥体的 6 个平面 (即图 6 中六面体的上、下、左、右、近、远平面) 依次判断空间位置。

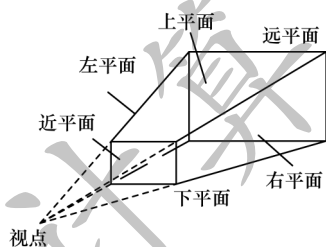


图6 视锥体示意图

Fig.6 Schematic diagram of visual cone

精细裁剪算法的总体思想: 当 MBC 块位于视锥体某一平面方程的外部时, 即为不可见, 则裁掉该物体对象; 当 MBC 块在上述过程并未被裁剪, 则说明其与视锥体的位置关系为包含或相交, 具体来说为下面 2 种情况:

1) 位置关系为包含。当 MBC 块处于 6 个平面内, 说明空间对象位于视点区域内, 则将空间对象直接送入 GPU 绘制管线中并进行渲染显示。

2) 位置关系为相交。当截锥体某一平面的内侧包含 MBC 的一部分, 且该平面外侧也包含一部分, 说明空间对象与视锥体是相交关系, 则继续对下层物体进行判断, 依次遍历所有的空间对象。

具体来说, 视点位置位于世界坐标系的零点, 首先将视锥体模型沿 Z 轴正向放置, 使用设定的投影矩阵转换顶点, 进而得到截锥体对应的 6 个平面方程。然后找出 MBC 块的顶点 n 点和 p 点, p 点是距离平面最近的顶点, n 点是最远对角线的顶点 (距离 p 最近的顶点): 设 2 个顶点的坐标分别为 $p(x_{\min}, y_{\min}, z_{\min})$, $n(x_{\max}, y_{\max}, z_{\max})$, 视锥体的 6 个平面方程为 $ax + by + cz + d = 0$, 将 n, p 分别带入 6 个平面方程中, 若 $ax_{\min} + by_{\min} + cz_{\min} + d > 0$, 则可以判断该 MBC 块离平面最近的

顶点在平面外侧, 则此 MBC 块在平面外侧, 对该 MBC 块进行裁剪。同理, 若 $p(x_{\min}, y_{\min}, z_{\min})$ 在平面内侧, $n(x_{\max}, y_{\max}, z_{\max})$ 在平面外侧, 说明 MBC 块与视锥体相交。如果 p 点和 n 点都在视锥体平面方程的内侧, 则说明 MBC 块在平面内侧, 直接将该 MBC 块的数据送入绘制管道渲染显示。

将视点移动过程看作如图 7 所示的视点动态模型, 移动过程将整个数据对象范围划分为当前可见区域、潜在预加载区域以及释放卸载区域。在经过时序卷积模型预测得到下一视点坐标之后, 通过比较视点划分的区域, 结合视锥体模型进行卸载和加载渲染。具体来说, 当视点处于 A 位置时, 图 7 中的 1、2 部分即为当前可见区域, 对其进行渲染显示, 潜在预加载区域则是划分为预测下一视点 B 的可见区域, 即图 7 的 2、3 部分。当视点移动到 B 时需要提前将该区域的数据标记为可见数据, 并载入到内存等待渲染, 便于用户在持续浏览时直接对数据对象进行读取显示, 避免交互时出现画面卡滞的现象。另外当视点由 A 运动到 B 时, 将区域 1 标记为释放卸载域以减少系统负荷。

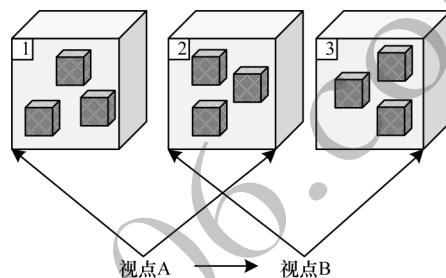


图7 视点动态模型示意图

Fig.7 Schematic diagram of viewpoint dynamic model

体数据调度加载的流程如图 8 所示。首先, 根据当前视点坐标并结合视点动态划分模型, 通过约束条件构建视锥体, 获取当前视点范围内必须绘制的数据块, 将当前可见的数据块送入 GPU 渲染管线。其次, 若视点持续进行匀速规则运动, 则通过算法预测下一视点坐标, 根据预测的坐标并结合视点动态区域模型获取潜在预加载区域的数据块, 通过视锥体裁剪潜在区域范围内的必要绘制数据, 并将其载入内存等待显示。最后, 释放卸载区域的数据, 开始下一帧的绘制。

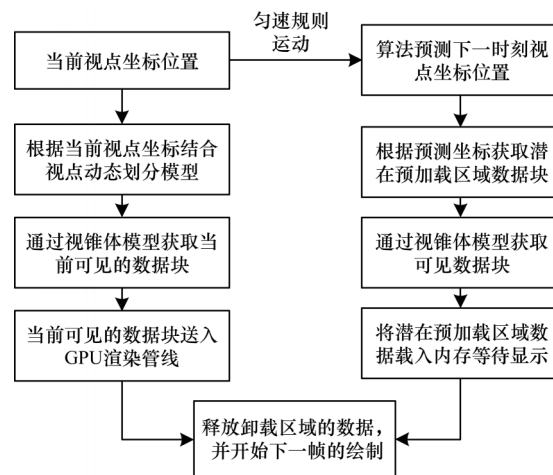


图8 体数据调度加载流程

Fig.8 Procedure of volume data scheduling loading

5 实验结果与分析

5.1 实验平台及设置

实验中用到的硬件设备为 Dell G5-5500,处理器为 Intel Core i7-10750H,芯片类型为 GeForce RTX 2060,CPU 主频 2.60 GHz,内存 16 GB。应用程序使用 Visual Studio 2015+PyCharm2019+ QT5.7 进行代码编写。

实验所用的数据为我国某油田部分的地震工区数据,并以 SEG-Y 格式进行存储。数据分为 3 组,数据集 A 为 469.7 MB,数据集 B(体数据分布较均匀)为 3 348.5 MB,数据集 C(数据量较大且较分散)为 14 643.4 MB,地震数据信息中包含埋藏深度、范围、厚度、伸展趋势等信息。

5.2 实验结果

本文分别针对索引结构、视点预测以及体数据调度加载模块进行对比评估,以证明本文方法的有效性。

5.2.1 数据结构的索引效率

为评估 VDEC 模型聚类的有效性,本文在 MNIST 手写数据集^[24]上对其进行验证,首先采用 K-means 算法初始化簇中心,使用编码器预训练,其中优化器采用 Adam 优化器得到模型最佳参数,将学习率设置 0.002,每训练 10 个 epoch 更新一次参数,输出维度设置为 10,训练 200 次,并根据 encode 编码层返回的均值和方差构建隐藏特征空间 z 。用 TSNE 降维方法将采样的小部分 z 映射到二维空间,并进行可视化,其潜在空间中的数据聚集分布如图 9 所示,可以看出数据的聚类效果较好,这说明潜在特征表示适合聚类。

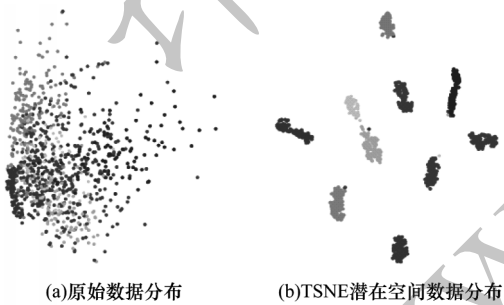


图9 VDEC 潜在空间的数据聚类分布
Fig.9 Data clustering distribution of VDEC potential space

此外,本文选择聚类准确率(Accuracy, ACC)、标准互化信息(Normalized Mutual Information, NMI)以及调整兰德指数(Adjusted Rand Index, ARI)作为聚类指标^[25],这 3 类指标的值均是越靠近 1,表明聚类效果越好。在 MNIST 数据集上用不同算法测试聚类效果,结果如表 1 所示。根据实验结果,本文算法在 3 个评价指标上依次比 K-means 算法提高了 44.79%、44.44%、59.78%,相比 AE 和 DEC 算法,其聚类效果也有一定提升。这说明本文算法相对于传统聚类算法的效果更好,能够为 HRT 树的建树过程奠定良好的结构基础,避免数据节点在建树过程中出现空间重叠问题,进而提高索引的效率。

表 1 不同算法在 MNIST 数据集上的结果对比
Table 1 Comparison of results of different algorithms under MNIST dataset

算法	ACC	NMI	ARI
AE ^[21] 算法	0.79	0.76	0.68
DEC ^[22] 算法	0.84	0.82	0.73
K-means ^[26] 算法	0.53	0.50	0.37
VDEC 算法	0.96	0.90	0.92

本文通过与其他索引结构算法的查询效率进行对比,评估本文算法整体的数据结构索引效果。具体来说,为证明数据量大小的差异对索引结果的影响,对 A 组数据分别取 2%、4%、6% 的数据量比率,同理 B、C 两组也取 2%、4%、6% 比率的数据,并比较 OCT、HRT、基于 K 均值算法的 HRT(KHRT)、变分深度嵌入聚类下的 HRT(Variational Deep Embedding Clustering-HRT, VDEC-HRT)等结构查询数据块算法的查询时间,结果如图 10 所示。

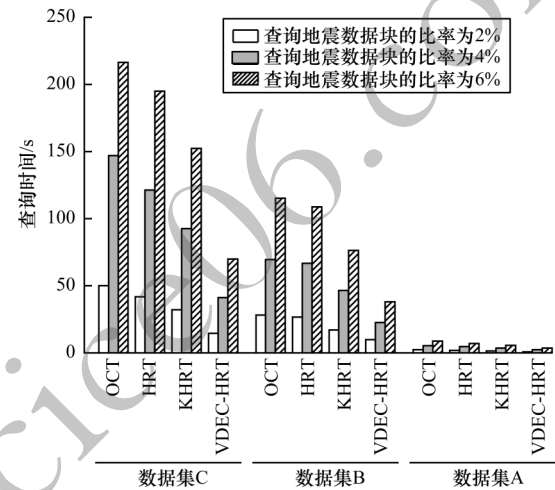


图 10 不同结构查询数据块算法的查询时间对比
Fig.10 Comparison of query time of different structure query data block algorithms

由图 10 可以看到,对于 A 组数据而言,本文算法(VDEC-HRT)相比 OCT、HRT 和 KHRT 算法的查询时间有所减少,但差距不明显。对于 B 组数据,与 OCT 算法相比,本文 VDEC-HRT 索引数据块的时间减少了 65.05%~66.96%,与 HRT 算法相比,本文算法索引时间减少了 63.10%~66.01%,与 KHRT 相比减少了 42.27%~51.74%。对于查询 C 组数据子块,与 OCT、HRT 和 KHRT 算法对比,本文算法的时间分别减少了 67.69%~72.22%, 64.14%~66.37%, 54.10%~55.89%。所以即使对于规模较大的数据,本文算法能通过改进聚类使数据节点空间减少重叠,从而有效减少频繁访问的次数,提高查询索引的效率。

5.2.2 对视点预测的评估

为评估预测算法的准确率,本文与拉格朗日插值、堆叠 LSTM 算法进行对比,分别在时长 1 min、5 min、

10 min 下预测数据块的准确率,结果如表 2 所示。由表 2 可知,本文算法的精度相比拉格朗日插值、堆叠 LSTM 算法分别提高了 12.08%~22.70%,2.24%~4.34%。虽然随着时长和数据集的增大,预测的正确率有所下降,但本文算法相对于另外 2 种算法的预测正确率仍然有一定提升。

表 2 不同算法在不同时长下预测数据块的准确率对比

Table 2 Comparison of accuracy of algorithm prediction data blocks in different time periods %				
数据集	算法	准确率		
		时长为 1 min	时长为 5 min	时长为 10 min
数据集 A	拉格朗日插值算法	82.53	75.45	70.81
	堆叠 LSTM 算法	92.37	88.96	87.57
	本文算法	94.61	92.35	90.74
数据集 B	拉格朗日插值算法	80.49	72.18	66.93
	堆叠 LSTM 算法	90.24	87.41	85.71
	本文算法	92.83	90.62	88.30
数据集 C	拉格朗日插值算法	71.33	65.97	61.66
	堆叠 LSTM 算法	87.55	84.67	80.02
	本文算法	90.67	87.28	84.36

此外,本文取 35 frame/s 左右的平均帧率预测采样部分位置坐标的 X 和 Y 分量,对比使用 TCN 和 LSTM 网络预测位置坐标的效果,TCN 网络的参数设置 nb_filters=64,kernel_size=3,残差块中使用 relu 激活函数,时间步长选取 20。迭代 20 次后得到的拟合结果如图 11 所示。在本文坐标数据集中,TCN、LSTM 网络的预测精度分别为 97.68%、96.92%。由图 11 可以看出,TCN 网络在 X、Y 坐标分量上的预测精度比 LSTM 网络稍好,说明其在本文预测视点的任务中是有效的,有利于后续数据调度加载的准确性。

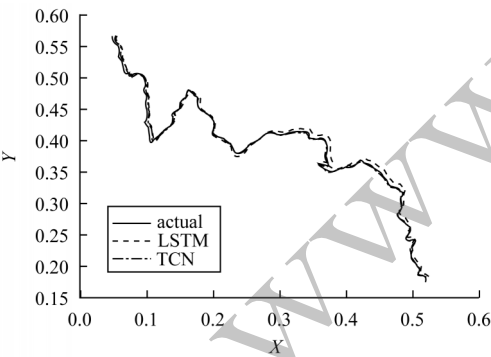


图 11 不同网络的拟合结果对比曲线
Fig.11 Comparison curve of fitting results of different networks

为测试步长对预测结果的影响,在平均帧率不变的情况下,分别设置步长为 10、20、30,并随机改变摄像机的位置和方位,使视点保持匀速运动持续 10 min,同时为了避免不同平均帧率实验的干扰,本文选择 24、48 和 96 frame/s 的平均帧率进行非同步长计算和对比预测错误率,结果如图 12 所示。

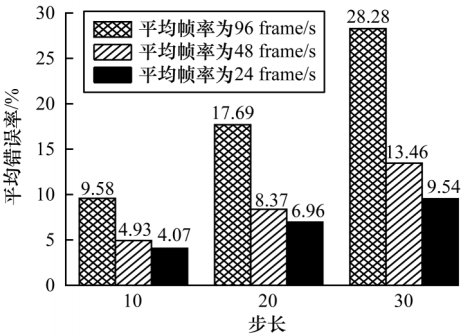


图 12 不同步长对预测准确率的影响
Fig.12 Influence of asynchronous length on prediction accuracy

由图 12 可知,误差会随着步长增加而增大,但步长越小,预测精度相对越低,因此可以通过调节步长参数来达到提高预测准确性的目的。

5.2.3 对数据加载渲染的评估

为测试本文算法的稳定性,选择 60 帧采样点作为 X 轴分量,分别在 A、B、C 数据集上比较无预测算法(即在本文算法基础上去除视点运动轨迹预测模块的算法)与本文算法的帧率,结果如图 13 所示。由图 13 可知,在 A 组数据集上,即使数据量较小,帧率也能保持在 50 frame/s 左右。在 B、C 组数据集上采用本文算法的帧率,相比无预测算法来说,即使在数据规模较大时,帧率仍然相对较高且较稳定,这意味着渲染效果更加稳定平滑,能够避免交互时产生波动。

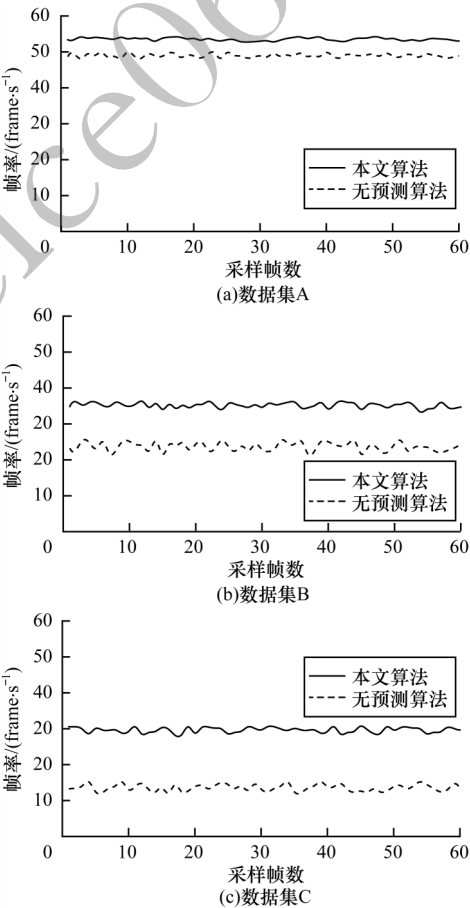


图 13 不同算法在 3 组数据集下的帧率对比
Fig.13 Comparison of frame rates of different algorithms under three sets of datasets

5.2.4 对整体性能的评估

为整体评估算法,将本文的VDEC-HRT、OCT、HRT、KHRT分别与基于TCN和双层视锥体裁剪的调度模型(Scheduling Model Based on TCN and Double-Layer Visual Cone Clipping,TDVSM)算法进行组合,并测试各算法组合的整体交互性能。以B组数据为例,每隔10片图像对地震数据进行切片,并记录各算法查询切片的时间,结果如图14所示。可以看出,与其他算法相比,本文算法查询切片的时间明显减少。

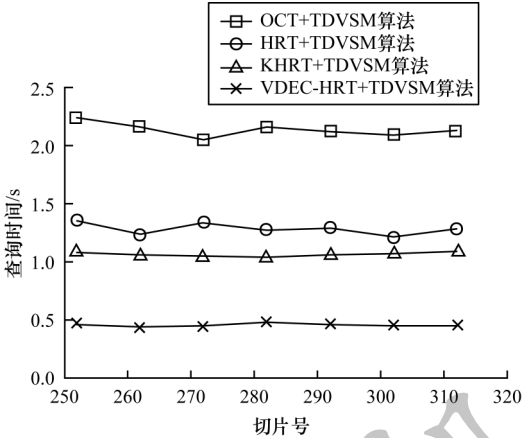


图14 不同算法查询切片的时间对比
Fig.14 Comparison of query slice time of different algorithms

最后,本文对系统的整体显示效果进行测试,系统平台设置有不同显示模式下的渲染效果,图15是体数据显示的蓝白红模式(彩色效果见《计算机工程》官网HTML版),图16是地震数据沿不同测线方向切片显示效果(彩色效果见《计算机工程》官网HTML版)。在运行观测实际的地震数据时,由图15和图16可以看出,本文系统平台能够高质量地描绘出地震体数据的内部结构和信息。

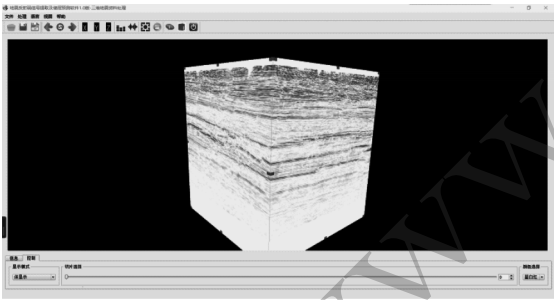


图15 地震体数据显示效果
Fig.15 Display effect of seismic body data

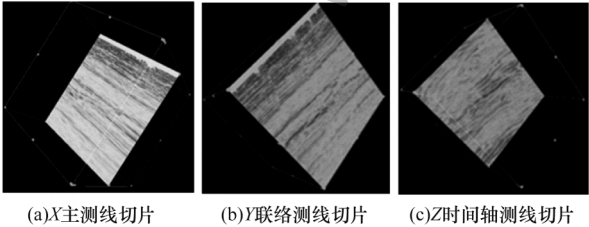


图16 地震数据沿不同测线方向切片的显示效果
Fig.16 Effect of slice display of seismic data along different survey lines

6 结束语

针对海量地震数据体绘制在集中载入时存在画面显示延迟和卡顿的问题,本文提出一种快速三维可视化算法。在HRT结构的基础上通过变分深度聚类改善节点数据因分布不均造成的空间重叠问题,提高数据索引的效率。使用时序卷积模型预测视点的运动轨迹,并与双层视锥体裁剪算法相结合,以更精确地将数据提前载入内存等待渲染,避免画面显示延迟,提高渲染速度。实验结果表明,本文算法相较于OCT、HRT、KHRT等算法的查询时间均有所减少,且渲染帧率波动较为稳定平滑,能够实现海量地震数据的快速三维可视化。下一步将使用其他类型的大规模数据集对本文算法进行测试,尝试采用wassterient距离代替KL散度,并在视点预测模块中引入注意力机制,从而提高图像质量。

参考文献

[1] VISUTSAK P. Marching cubes and histogram pyramids for 3D medical visualization[J]. Journal of Imaging, 2020, 6(9): 88-98.

[2] LI L F, QI M R. Design of three-dimensional visualization system for medical image based on magnetic resonance angiography data[J]. Matrix Science Medica, 2019, 3(2): 28-34.

[3] HAN X, LIU J M, TAN B H, et al. Design and implementation of smart ocean visualization system based on extended reality technology[J]. Journal of Web Engineering, 2021, 20(2): 557-574.

[4] BI M L, XUE X G, LI C H, et al. Discussion on 3D visualization model of geological structure [J]. Journal of Physics: Conference Series, 2019, 1345(5): 14-21.

[5] 周文辉,石敏,朱登明,等. SEGY 格式地震数据的三维可视化[J]. 计算机应用与软件, 2019, 36(2): 78-84, 154. ZHOU W H, SHI M, ZHU D M, et al. Three-dimensional visualization of seismic data in segy format[J]. Computer Applications and Software, 2019, 36(2): 78-84, 154. (in Chinese)

[6] 王华维,何柳,曹轶,等. 大规模科学数据体绘制技术综述[J]. 国防科技大学学报, 2020, 42(2): 1-12. WANG H W, HE L, CAO Y, et al. A survey of the techniques of volume rendering for large-scale scientific data[J]. Journal of National University of Defense Technology, 2020, 42(2): 1-12. (in Chinese)

[7] AGRAWAL A, KUMAR P. GPU-based focus-driven multi-coordinates viewing system for large volume data visualisation[J]. International Journal of Computational Systems Engineering, 2018, 4(2/3): 86-92.

[8] XIE K, WU P, YANG S. GPU and CPU cooperation parallel visualisation for large seismic data[J]. Electronics Letters, 2010, 46(17): 1196-1203.

[9] 李彩林,陈文贺,胡善明,等. 采用八叉树和OSG分页结点海量点云三维可视化[J]. 计算机工程与应用, 2019, 55(21): 234-239. LI C L, CHEN W H, HU S M, et al. Massive point cloud visualization using octree and OSG paging nodes[J]. Computer Engineering and Applications, 2019, 55(21): 234-239. (in Chinese)

- [10] LIMKAR S V, JHA R K. A novel method for parallel indexing of real time geospatial big data generated by IoT devices[J]. *Future Generation Computer Systems*, 2019, 97: 433-452.
- [11] HAI Y, HAN B, SU W, et al. R-tree index construction of dynamic K-means algorithm[C]//*Proceedings of the 14th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery*. Washington D. C. , USA; IEEE Press, 2018: 831-836.
- [12] 侯海耀, 钱育蓉, 英昌甜, 等. 基于 Hilbert-R 树分级索引的时空查询算法[J]. *计算机应用*, 2018, 38(10): 2869-2874, 2885.
HOU H Y, QIAN Y R, YING C T, et al. Spatio-temporal query algorithm based on Hilbert-R tree hierarchical index[J]. *Journal of Computer Applications*, 2018, 38(10): 2869-2874, 2885. (in Chinese)
- [13] 王响, 雷小永, 戴树岭. 基于视点预测的大规模地形的实时渲染[J]. *系统仿真学报*, 2013, 25(6): 1202-1206.
WANG X, LEI X Y, DAI S L. Real time rendering of large scale terrain based on viewport prediction[J]. *Journal of System Simulation*, 2013, 25(6): 1202-1206. (in Chinese)
- [14] CHENG H, XIE K, WEN C, et al. Fast visualization of 3D massive data based on improved Hilbert R-tree and stacked LSTM models[J]. *IEEE Access*, 2021, 9: 16266-16278.
- [15] SUO Y, CHEN W, CLARAMUNT C, et al. A ship trajectory prediction framework based on a recurrent neural network[J]. *Sensors(Basel)*, 2020, 20(18): 5133-5143.
- [16] MIKHAILOV S, KASHEVNIK A. Car tourist trajectory prediction based on bidirectional LSTM neural network[J]. *Electronics*, 2021, 10(12): 1390-1403.
- [17] SERPA Y R, RODRIGUES M A F. A draw call-oriented approach for visibility of static and dynamic scenes with large number of triangles[J]. *The Visual Computer*, 2019, 35(4): 549-563.
- [18] JIANG Y Z, NIU P. View frustum culling algorithm for scene based on adaptive binary tree[C]//*Proceedings of the 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference*. Washington D. C. , USA; IEEE Press, 2018: 1159-1164.
- [19] WEN C, LI L, XIE K. Fast visualisation of massive data based on viewpoint motion model[J]. *Electronics Letters*, 2017, 53(15): 1038-1040.
- [20] 孙小虎, 韩文军, 潘娟, 等. 基于聚合与动态调度的变电站三维可视化研究[J]. *计算机工程*, 2019, 45(12): 300-307.
SUN X H, HAN W J, PAN J, et al. Research on three-dimensional visualization of substation based on aggregation and dynamic scheduling[J]. *Computer Engineering*, 2019, 45(12): 300-307. (in Chinese)
- [21] WEI R Q, GARCIA C, EL-SAYED A, et al. Variations in variational autoencoders-a comparative evaluation [J]. *IEEE Access*, 2020, 8: 153651-153670.
- [22] XIE J Y, GIRSHICK R, FARHADI A. Unsupervised deep embedding for clustering analysis[EB/OL]. [2021-10-01]. <https://arxiv.org/abs/1511.06335>.
- [23] BAI S J, KOLTER J Z, KOLTUN V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling[EB/OL]. [2021-10-01]. <https://arxiv.org/abs/1803.01271>.
- [24] YANN L C, CORINNA C, CHRISTOPHER J C. The mnist database of handwritten digits[EB/OL]. [2021-10-01]. <http://yann.lecun.com/exdb/mnist/>.
- [25] CAI D, HE X, HAN J. Document clustering using locality preserving indexing[C]//*Proceedings of IEEE Transactions on Knowledge and Data Engineering*. Washington D. C. , USA; IEEE Press, 2005: 1624-1637.
- [26] KHAN S S, AHMAD A. Cluster center initialization algorithm for K-means clustering[J]. *Pattern Recognition Letters*, 2004, 25(11): 1293-1302.