

改进秃鹰搜索和 K 均值混合迭代的点云简化算法

牛宏侠^{1,2}, 李富丽^{2,3}

(1. 兰州交通大学自动化与电气工程学院, 730070, 兰州;

2. 兰州交通大学甘肃省高原交通信息工程及控制重点实验室, 730070, 兰州;

3. 兰州交通大学光电技术与智能控制教育部重点实验室, 730070, 兰州)

摘要: 针对激光雷达的固有特性和复杂环境易造成点云噪声和冗余点云,以及传统点云简化算法忽略了点云固有特征等问题,提出了一种基于改进秃鹰搜索和 K 均值聚类(KMC)混合迭代的点云简化算法(IBESEA)。首先,通过秃鹰搜索(BES)算法迭代阶段的竞争融合(CFBES),提高其收敛速度和优化精度;其次,通过 CFBES 和 KMC 算法的混合迭代,实现了点云数据的聚类;然后,在 k 近邻(k -NN)实现点云簇密度估计的基础上,结合香农熵实现点云信息量化;最后,删除信息量化值小于阈值的聚类簇,完成点云数据简化。使用 UCI 国际标准数据集和斯坦福点云数据集分别对 CFBES-KMC 算法的聚类效果及点云的简化效果进行验证,结果表明:与改进飞蛾扑火的 K 均值交叉迭代、 K -means++、模糊 C 均值聚类算法的聚类效果相比,CFBES-KMC 算法的聚类准确率分别提高了 1.02%、12.31%、14.72%;在斯坦福点云数据集上,IBESEA 算法在有效滤除冗余点云的基础上保留了原本点云的细节和形状特征,不失为一种高效的点云简化算法。

关键词: 秃鹰搜索算法;竞争融合; K 均值聚类混合迭代;香农熵;点云简化

中图分类号: TP391 **文献标志码:** A

DOI: 10.7652/xjtuxb202402018 **文章编号:** 0253-987X(2024)02-0172-12

Point Cloud Simplification Algorithm Based on Improved Bald Eagle Search and K -Means Clustering Iteration

NIU Hongxia^{1,2}, LI Fuli^{2,3}

(1. School of Automation and Electrical Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China;

2. Key Laboratory of Plateau Traffic Information Engineering and Control of Gansu Province, Lanzhou Jiaotong University,

Lanzhou 730070, China; 3. Key Lab of Opt-Electronic Technology and Intelligent Control of Ministry of Education, Lanzhou Jiaotong University, Lanzhou 730070, China)

Abstract: Aiming at the inherent characteristics of lidar and the point cloud noise and redundant point cloud that are easily caused in complex environment, as well as the problem that the inherent features of point cloud are often ignored in conventional point cloud simplification algorithms, a point cloud simplification algorithm based on improved bald eagle search and K -means clustering (KMC) hybrid iteration (IBESEA) is proposed. Firstly, the convergence is accelerated and optimization accuracy is upgraded through the competitive fusion in the BES iteration stage (CFBES). Secondly, the point cloud data is clustered by the hybrid iteration of CFBES and KMC

algorithm. Then, based on the implementation of point cloud density estimation with k -nearest neighbors (k -NN), point cloud information quantization is achieved combined Shannon entropy. Finally, the cluster with the information quantization value less than the threshold is deleted to complete the simplification of point cloud data. UCI international standard dataset and Stanford point cloud dataset are used to verify the clustering effect and point cloud simplification effect of CFBES-KMC. The results show that compared with hybrid iterative K -means clustering with improved moth-flame optimization, K -means++ and fuzzy C -means clustering algorithm, the clustering accuracy of CFBES-KMC is improved by 1.02%, 12.31%, 14.72% respectively. On the Stanford point cloud dataset, the IBESSA algorithm effectively filters out redundant point clouds while preserving the details and shape features of the original point cloud, which means it is an efficient point cloud simplification algorithm.

Keywords: bald eagle search algorithm; competitive fusion; K -means clustering hybrid iteration; Shannon entropy; point cloud simplification

随着 3D 目标检测技术的发展,点云的深度学习受到众多学者关注^[1]。然而,点云数据无序和离散的特点对 3D 检测任务造成了极大的困扰,因此,三维点云简化算法的研究对点云的深度学习显得尤为重要。经典的点云简化算法主要包括包围盒法^[2]、随机采样法^[3]、基于特征的点云简化算法^[4]等。Pauly 等^[5]通过计算空间二值化,提出了一种基于点样本分层分解的简化方法,但由于空间性质的原因,很难控制采样表面分布点的质量。Xuan 等^[6]引入信息熵和向量角,提出了一种渐进点云简化方法,该方法的基础是利用法向角的信息熵来寻找重要的点,并通过删除不相关的点来进行简化操作;然而,不相关点的选择具有极大的不确定性。Mahdaoui 等^[7]提出了一种基于香农熵和 K 均值聚类 (K -means clustering, KMC) 的方法来简化三维点云,由于聚类时受点云空间形状的限制,因此聚类差异较大;同时,由于聚类中心点选取的随机性以及聚类优化路径的单一性,导致聚类结果不稳定。群智能算法在对聚类中心进行优化方面具有显著优势^[8-9],结合聚类算法和信息估计为点云简化提供了一种新的思路。

秃鹰搜索算法(bald eagle search, BES)是 2020 年提出的一种新型优化算法^[10],由于可扩展性强,在大规模全局优化问题中能够有效地跳出局部最优,与其它群智能算法相比具有更强的全局搜索能力。BES 算法通过生物行为构建数学模型,主要分为选择、搜索和俯冲 3 个阶段,与 KMC 算法结合后能够实现点云的高精度聚类。由此,本文提出了一种基于改进秃鹰搜索和 KMC 混合迭代的点云简化算法(improved bald eagle search and KMC hybrid

iteration simplification algorithm, IBESSA)。首先,对 BES 算法的优化和更新迭代方式进行改进,通过 BES 算法迭代阶段的竞争融合(competitive fusion bald eagle search, CFBES)加速收敛;然后,通过 CFBES 和 KMC 算法的混合迭代,实现点云数据的聚类;最后,引入点云信息熵概念,结合点云聚类结果实现了点云的简化。

1 CFBES 算法

1.1 BES 算法

BES 算法通过模拟秃鹰捕食鲑鱼的过程,在搜索空间内对解进行优化。BES 算法主要分为选择、搜索和俯冲 3 个阶段。

(1)选择阶段:秃鹰通过观察区域内猎物的多少来确定合适的搜索区域,为下一阶段搜索猎物做好准备。该阶段秃鹰位置 $P_{i, new}$ 可用数学模型描述为

$$P_{i, new} = P_{best} + \alpha \eta (P_{mean} - P_i) \tag{1}$$

式中: P_{best} 为当前秃鹰确定的最佳搜索位置; α 为控制位置变化的参数, $\alpha \in (1.5, 2)$; η 为随机数, $\eta \in (0, 1)$; P_{mean} 为秃鹰根据所有搜索结果计算出的平均分布位置; P_i 为第 i 只秃鹰的位置。

(2)搜索阶段:秃鹰在先前选定的搜索区域内进行螺旋运动,不断改变搜索的角度和速度以寻找最佳俯冲捕获位置,并同时向最佳位置移动。其运动行为可描述如下

$$\theta(i) = \alpha \pi \eta \tag{2}$$

$$r(i) = \theta(i) + R\eta \tag{3}$$

$$x(i) = r(i) \sin(\theta(i)) / \max(|r(i) \sin(\theta(i))|) \tag{4}$$

$$y(i) = r(i) \cos(\theta(i)) / \max(|r(i) \cos(\theta(i))|) \tag{5}$$

$$\mathbf{P}_{i,\text{new}} = \mathbf{P}_i + x(i)(\mathbf{P}_i - \mathbf{P}_{\text{mean}}) + y(i)(\mathbf{P}_i - \mathbf{P}_{i+1}) \quad (6)$$

式中: $\theta(i)$ 和 $r(i)$ 分别为螺旋方程的极角和极径; a 和 R 为控制秃鹰飞行轨迹的常数, 影响秃鹰搜索周期的长短, 其中 $a \in (5, 10)$, $R \in (0.5, 2)$; η 为 $(0, 1)$ 内的随机数; $x(i)$ 和 $y(i)$ 分别为极坐标中秃鹰个体的位置, 取值均为 $(-1, 1)$; $\mathbf{P}_{i+1}$ 为第 i 只秃鹰下一次准备更新的位置。

(3) 俯冲阶段: 秃鹰以上一阶段搜索到的最佳位置为中心, 向猎物方向的最佳点移动并对其进行捕获。仍用极坐标方式来描述秃鹰的运动状态, 如下所示

$$\theta(i) = a\pi\eta; r(i) = \theta(i) \quad (7)$$

$$x_1(i) = r(i)\sinh(\theta(i))/\max(|r(i)\sinh(\theta(i))|) \quad (8)$$

$$y_1(i) = r(i)\cosh(\theta(i))/\max(|r(i)\cosh(\theta(i))|) \quad (9)$$

式中: $x_1(i)$ 、 $y_1(i)$ 为俯冲阶段秃鹰个体在极坐标系中的位置。

秃鹰在俯冲过程中的位置更新公式可表示为

$$\mathbf{P}_{i,\text{new}} = \eta\mathbf{P}_{\text{best}} + x_1(i)(\mathbf{P}_i - c_1\mathbf{P}_{\text{mean}}) + y_1(i)(\mathbf{P}_i - c_2\mathbf{P}_{\text{best}}) \quad (10)$$

式中: c_1 、 c_2 分别为秃鹰向最佳位置和中心位置运动强度的大小, 取值范围为 $(1, 2)$ 。

1.2 改进 BES 算法

1.2.1 基于混沌映射的秃鹰初始化

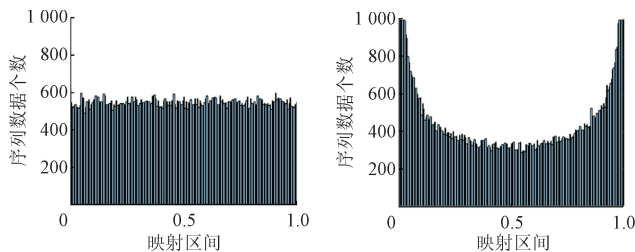
由于秃鹰种群初始分布是随机的, 生成的种群不够均匀, 因此严重影响到种群的收敛速度和最终解的精度。蒋宇飞等^[11]在蜉蝣算法中引入 Sine 混沌映射用于蜉蝣种群的初始化, 使种群能够均匀分布在解空间中, 从而提高了初始种群的质量, 优化了随机初始化产生的缺陷, 证明了混沌映射方法对元启发式算法的有效性。

本文利用混沌映射遍历性和随机性的特性, 在传统 Tent 混沌映射中加入调节机制 v/N , 得到新的映射机制, 如下所示

$$X_{i,j+1} = \begin{cases} \beta X_{i,j} + v/N, & X_{i,j} < 0.5 \\ \beta(1 - X_{i,j}) + v/N, & X_{i,j} \geq 0.5 \end{cases} \quad (11)$$

式中: $X_{i,j+1}$ 为映射得到的混沌值, i 为种群大小, $i = 1, 2, \dots, N$, j 为混沌序号, $j = 1, 2, \dots, d$; $X_{i,j}$ 为种群大小为 i 、混沌序号为 j 的混沌值; v 为随机数, $v \in [0, 1]$; β 为混沌参数, $\beta \in [0, 2]$ 。图 1 给出了混沌映射序列分布的直方图对比, 其中横坐标

表示映射区间, 纵坐标表示序列中落在相应取值区间内的数据个数。图 1(a) 为改进型 Tent 映射在区间 $[0, 1]$ 的分布图, 此时 $\beta = 1.2$; 图 1(b) 为经典的 Logistic 序列直方图。



(a) 改进型 Tent 序列直方图 (b) 经典 Logistic 序列直方图

图 1 混沌映射分布直方图对比

Fig. 1 Comparing histograms of chaotic map distribution

通过对比两种序列直方图, 发现改进后的 Tent 混沌序列能够使生成的初始种群较为均匀, 而均匀分布的初始位置使得种群更容易从局部最优解中逃脱, 从而寻找全局最优解。因此, 使用改进型 Tent 映射生成随机序列, 有利于前期的全局搜索以及种群的随机初始化。

通过不断迭代, 得到一系列混沌值 $X_{i,j}$, 再将其映射到种群搜索空间, 映射规则如下

$$\mathbf{P}_{i,j} = \mathbf{L}_{j,\min} + X_{i,j}(\mathbf{U}_{j,\max} - \mathbf{L}_{j,\min}) \quad (12)$$

式中: $\mathbf{L}_{j,\min}$ 、 $\mathbf{U}_{j,\max}$ 分别为秃鹰位置 $\mathbf{P}_{i,j}$ 的下边界和上边界。

1.2.2 竞争淘汰机制

秃鹰搜索算法中, 位置更新主要靠个体间信息的相互交流, 由于个体差异较大, 导致同一区域内的秃鹰通过螺旋运动进行食物搜索时, 需要多次迭代才能捕获猎物, 同时, 性能较弱的个体不能较快地达到最优区域, 致使秃鹰在进行捕获猎物时寻优效率过慢, 且在精度要求越高的场合, 算法效率就越低。为进一步提高 BES 算法的开发能力, 本文借鉴捕猎的习性, 在秃鹰搜索阶段设计了竞争淘汰机制对秃鹰种群进行筛选, 其基本过程主要分为个体筛选和盘旋寻优两个阶段。第一阶段以秃鹰个体的适应度函数作为筛选标准, 根据种群中子代个体的适应度判断个体的优化性能, 淘汰适应度较差的个体, 从而获得新的秃鹰种群。第二阶段对第一阶段得到的新秃鹰种群进行增殖, 以中心辐射法模拟秃鹰低空盘旋搜寻路线, 获得新的秃鹰增殖种群, 然后再次对新的种群进行适应度排序, 按照初始种群数保留新的优秀个体, 淘汰质量较差的个体, 以保证种群数量不变。竞争淘汰机制具体过程如下。

首先,根据适应度设定阈值,通过优化性能差异进行个体筛选操作,保留适应度高于设定阈值的个体,淘汰适应度差的个体,同时更新秃鹰种群的数量,具体淘汰机制如下

$$p_{i+1} = \begin{cases} p_{i+1}, & r < F_{\text{SortIndex}(i)} \\ 0, & \text{其他} \end{cases} \quad (13)$$

式中: p_{i+1} 为根据筛选原则保留下来的适应度较优的个体; r 为 $(0,1)$ 间服从均匀分布的随机数; $F_{\text{SortIndex}(i)}$ 为当前种群中第 i 个秃鹰个体对应的优化性能,表达式可写为

$$F_{\text{SortIndex}(i)} = 1 - \sigma \lg \left(\frac{f_{\max} - f(i)}{f_{\max} - f_{\min}} + 1 \right) \quad (14)$$

式中: $\text{SortIndex}(i)$ 为个体 p_i 排序后的索引值; σ 介于 0 和 1 之间,旨在任意方向形成搜索向量,提高找到最优个体的可能性; $f_{\max}$ 、 $f_{\min}$ 分别为当前种群中最优适应度和最差适应度; $f(i)$ 为第 i 个秃鹰个体的适应度。

通过个体筛选策略,适应度低于阈值的个体会被淘汰,性能较优的个体会被保留,算法的寻优能力和搜索精度得到了提高。

以中心辐射法模拟上一阶段得到的秃鹰种群低空盘旋搜寻路线,进而获得新的秃鹰增殖种群,再次对新的种群进行适应度排序,按照初始种群数保留新的种群个体,以保证种群数量不变。秃鹰个体在搜索空间仍然服从均匀分布,用 E_i 表示通过中心辐射法得到的秃鹰数量, O_i 表示以最优位置进行辐射增值的秃鹰种群半径,表达式可写为

$$E_i = \lambda \frac{f_{\max} f(i) + b}{f_{\max} \sum_{i=1}^N f(i) + b} \quad (15)$$

$$O_i = \zeta \frac{f(i) - f_{\min} + b}{\sum_{i=1}^N (f(i) - f_{\min}) + b} \quad (16)$$

式中: λ 为辐射系数,用以决定种群中产生有效个体的能力, λ 越大种群产生有效个体的数量越多; b 为增值步长; N 为种群数量; ζ 为调节常数。

竞争淘汰机制可以通过淘汰适应度低的个体,使优质个体更容易地在搜索空间中找到更好的解,并避免过早收敛于局部极值。同时,在竞争淘汰过程中,采用中心辐射增值法会不断产生新的种群,并与原有个体进行竞争,从而保持种群多样性,避免种群陷入全局最优。通过设置适当的增值步长、辐射系数等参数,能够进一步增强算法的全局搜索能力。

2 CFBES 与 KMC 算法的混合迭代

KMC 算法^[12]是一种无监督的机器学习算法,

其将数据点划分为 K 个簇,广泛应用于数据分析和深度学习领域。KMC 算法的具体流程如下。

(1) 随机选取 K 个对象作为初始聚类中心,按照距离最近原则将数据分为 K 组。

(2) 计算每个聚类簇各个维度的平均值,并将其作为新聚类中心用以更新聚类簇。

(3) 重复过程(2),直到新的聚类中心不再变化或达到最大迭代次数,输出聚类中心。

KMC 算法采用迭代方式求取各个维度的均值,但通过此方法确定聚类中心的局限性在于易陷入局部最优,导致优化精度过低。本文通过采用 CFBES 代替 KMC 的优化路径,实现 CFBES 与 KMC 的混合迭代,扩大了寻优范围,从而提升了聚类的效果。采用同一簇内聚类中心和样本点之间距离平方和,定义 CFBES 算法的适应度函数为

$$f(\mu_j) = \sum_{i=1}^m \|\mu_j - x\|_2^2, \quad 1 \leq j \leq K \quad (17)$$

式中: μ_j 为第 j 个聚类中心; x 为样本点; m 为该簇中所有样本点的个数; K 为聚类个数。通过适应度函数可知,当前聚类效果的好坏受该类样本数及每个类别中所有样本点到该簇聚类中心距离的影响。

聚类数目的设定会影响到最终的聚类结果,不同点云数据由于空间分布差异大,无法预先统一聚类数目。为适应不同数据空间特征,引入肘部法则(elbow method, EM)来确定点云数据的最佳聚类数目。EM 法则的适用条件为:需要确定的聚类数目位于一个合理的范围内,且数据的聚类结构比较清晰,聚类簇之间的聚类有明显差异。EM 法则对于点云数据同样适用,其基本思想为:随着聚类数目的增加,聚类效果会不断提高,但当聚类数目达到某一阈值时,聚类效果的提升会逐渐减缓,这个阈值即为最优聚类数目。EM 法则的实现原理为:利用误差平方和 S_{SSE} 画出 K - S_{SSE} 曲线,通过观察图像,找到图像中出现“拐点”的位置,这个拐点对应的聚类数目即为最优聚类数目。 S_{SSE} 的表达式可写为

$$S_{\text{SSE}} = \sum_{i=1}^K \sum_{x \in C_i} |x - \mu_i| \quad (18)$$

式中: C_i 为第 i 个簇; x 为样本点; μ_i 为 C_i 的质心。

图 2 给出了 EM 法则的取值示意图。其中,图 2(a) 表示输入点云数据散点图;图 2(b) 表示根据输入的数据进行聚类得到不同的簇;同时计算误差平方和确定最佳簇数,最后得到的聚类结果如图 2(c)。由

图可知,当聚类数目为 4 时, S_{SSE} 的斜率变化最大,由此可知,EM 法则取值合理。

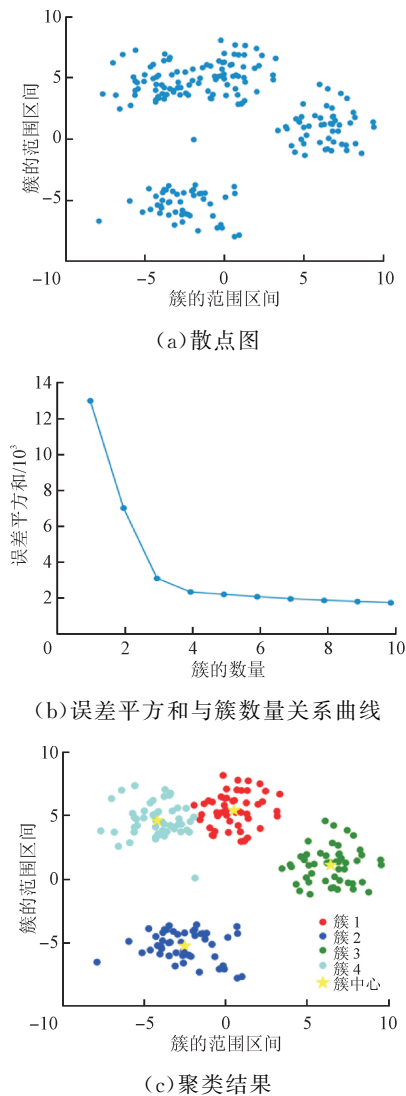


图 2 EM 法则取值示意图

Fig. 2 Schematic diagram of EM values

CFBES-KMC 算法流程如图 3 所示。算法的基本步骤描述如下。

步骤 1 输入标准点云数据集,计算误差平方和 S_{SSE} ,确定初始聚类数目 K 。

步骤 2 利用混沌映射对秃鹰搜索算法进行初始化。

步骤 3 根据秃鹰搜索策略对种群进行更新。

步骤 4 利用改进的秃鹰搜索算法对种群进行寻优操作,得到新的聚类中心。若新得到的聚类中心适应度优于上次迭代的聚类中心,用新的聚类中心代替历史聚类中心。

步骤 5 判断当前迭代是否达到结束条件,若达到终止条件,输出寻优结果,结束程序。若未到达,则跳到步骤 3,继续执行。

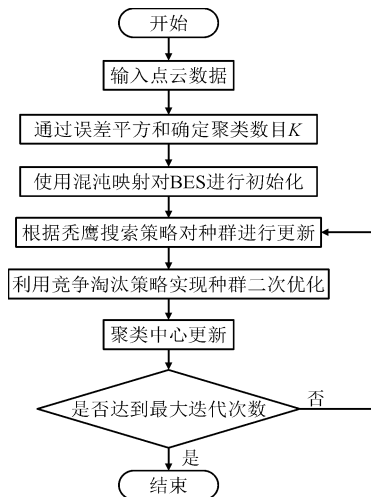


图 3 CFBES-KMC 算法流程

Fig. 3 CFBES-KMC algorithm flow char

3 点云信息量化

在三维点云简化任务中,需要通过信息熵对局部点云包含信息进行量化,通过计算信息熵评估点云中点的分布情况,涉及到密度评估函数^[13-15],密度评估函数一般包括参数法和非参数法两种。由于点云数据的无序性,采用参数法进行密度估计存在一定的难度,因此,本文采用 k 近邻 (k -nearest neighbors, k -NN) 估计去完成点云信息熵的计算。

3.1 基于高斯核函数的 k 近邻密度估计

高斯核函数^[16]是一种径向基核函数,常用于处理非线性数据,其基本原理是将数据映射到一个无限维的特征空间中,使数据变得线性可分,且具有很好的非线性分类能力。因此,本文结合高斯核函数提出了一种点云 k -NN 密度估计方法。

点云 k -NN 密度估计通过计算样本空间中每个点云的 k 近邻距离来估计该点处的概率密度,估计量的水平由 k 定义,它是最近邻的整数,与样本 N 成比例。样本对象 x 与其余点之间的距离关系可表示如下

$$Z_1(x) < \cdots < Z_{k-1}(x) < Z_k(x) < \cdots < Z_N(x) \quad (19)$$

式中: Z_k 为样本对象 x 到第 k 近邻点的距离。

d 维 k -NN 估计定义如下

$$Q_{k-NN}(x) = N^{-1} Z_k(x)^{-d} \sum_{i=1}^N G[Z_k(x)^{-1}(x - x_i)] \quad (20)$$

式中: $G(u)$ 为高斯核函数,可写为

$$G(u) = (2\pi)^{-\langle d/2 \rangle} \exp\left(-\frac{1}{2}u^T u\right) \quad (21)$$

将式(21)代入式(20),可得

$$Q_{k\text{-NN}}(x) = N^{-1} Z_k(x)^{-d} (2\pi)^{-\langle d/2 \rangle} \cdot \sum_{i=1}^N \exp\left[-\frac{1}{2} (Z_k(x)^{-1} (x - x_i))^2\right] \quad (22)$$

进一步推导可得

$$Q_{k\text{-NN}}(x) = N^{-1} C_d^{-1} Z_k(x)^{-1} \cdot \sum_{i=1}^N \exp\left[-\frac{1}{2} (Z_k(x)^{-1} (x - x_i))^2\right] \quad (23)$$

式中: C_d 为 d 维空间中单位球体的体积。

由上可见, k -NN 密度估计方法的优点是不需要对概率密度函数进行假设,且对密度函数的局部变化具有较好的适应性。

3.2 信息熵定义

香农熵^[17]是由美国科学家克劳德香农于1948年提出的一个数学函数,其直观地对应于信息源所包含或传递的信息量。在点云簇的 k -NN 密度估计完成后,需要根据得到的局部密度值来计算点云的信息熵。点云的信息熵表示点云中点分布的不确定度,可以采用香农熵来计算,公式如下

$$H(X) = - \sum_{i=1}^n Q(X = x_i) \log_2 Q(X = x_i) \quad (24)$$

式中: n 为点云中的点数; $Q(X = x_i)$ 为 x_i 的密度估计值。

通过式(24),可以计算每个点云簇中各个点的贡献信息熵,并将其累加得到整个点云簇的信息熵。对于孤立的噪声点云而言,其信息熵通常为0。因此,香农熵用来表示每个点云中的信息量大小,数值越大包含的信息量越多。如果一个点云的信息熵为0,则意味着该点云的取值是确定的,不包含任何信息。基于对信息量的估计,可以准确地评估该点云簇是否包含主要特征,从而确定是否需要保留该簇。

3.3 简化误差评估

为了评价本文所提简化方法的准确性,采用平均欧式距离(mean euclidean distance, MED)^[18]指

标来评估全局误差,采用 Hausdorff 距离^[19]测量简化后两个点集之间的局部误差。

平均欧式距离能够直观地反映简化后点云和原始点云之间的差异,其原理是计算简化后点云与原始点云之间的欧式距离,并取其平均值来评估简化误差的大小。设两个点集分别为 $X = x_1, x_2, \dots, x_m$ 和 $Y = y_1, y_2, \dots, y_n$, 则平均欧氏距离可写为

$$\text{dist}(X, Y) = \frac{1}{m} \sum_{i=1}^m \min_{j=1}^n |x_i - y_j| + \frac{1}{n} \sum_{j=1}^n \min_{i=1}^m |x_i - y_j| \quad (25)$$

Hausdorff 距离常用于评估简化后点云与原始点云之间的局部误差,其基本思想是找到一个点在两个集合之间的最短距离,然后取最大值。给定点集 A 和点集 B ,它们之间的 Hausdorff 距离可定义为

$$H(A, B) = \max\left\{\sup_{a \in A} \inf_{b \in B} d(a, b), \sup_{b \in B} \inf_{a \in A} d(a, b)\right\} \quad (26)$$

式中: $d(a, b)$ 为点 a 和点 b 之间的距离; $\inf$ 为下确界操作; $\sup$ 为上确界操作。式(26)可解释为:对于 A 中每个点 a , 计算与 B 中最近点 b 的距离 $d(a, b)$, 取所有距离中的最大值;对于 B 中的每个点 b , 计算与 A 中最近点 a 的距离 $d(a, b)$, 取所有距离中的最大值;将两个最大值中较大的一个作为 Hausdorff 距离,此值越小,说明简化后的点云越接近原始点云。

3.4 基于 CFBES-KMC 算法和信息熵的三维点云简化

三维点云简化的目的是选择相关且具有代表性的三维点,去除冗余的数据点。本文将改进的 CFBES-KMC 算法用于简化密集点云,如图4所示。首先使用 CFBES-KMC 算法将点云细分为若干个簇,然后计算每个簇的信息熵,根据所得熵对簇进行降序排列,从而去除信息熵低的簇。这种简化方法既保留了鲜明的轮廓特征,又在简化点集中保留了细节特征,适用于简化非均匀分布的点集。

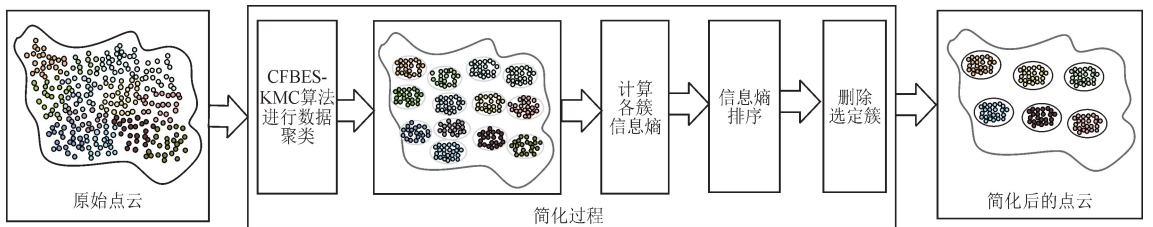


图4 主程序流程图

Fig. 4 Flow chart of the main program

4 实验与结果分析

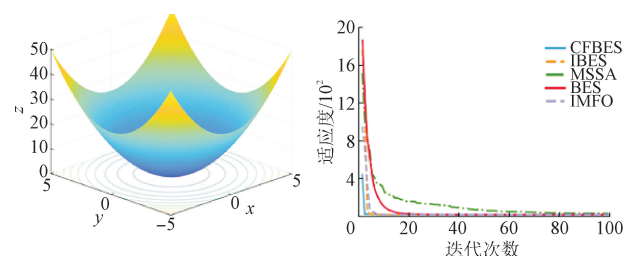
实验硬件平台为 Intel Core i7-12700KF 处理器, NVIDIA GeForce RTX 3060 Ti 显卡 $\times 2$, 32 GB 内存的计算机, 操作系统为 Windows 11, 软件实现语言为 python。

4.1 CFBES 算法性能测试

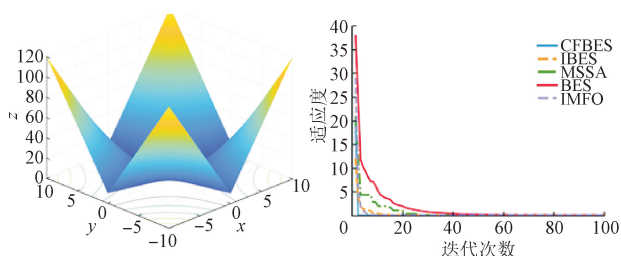
为验证 CFBES 算法的优化性能, 将 CFBES 与 BES^[9]、改进秃鹰搜索 (IBES)^[20]、非均匀变异麻雀搜索 (MSSA)^[21]、改进飞蛾扑火 (IMFO)^[22] 4 种优化算法进行对比, 采用单峰测试函数 $F_1 \sim F_4$ 测试其收敛效果, 多峰测试函数 $F_8 \sim F_{10}$ 及 F_{14} 测试其局部寻优及全局搜索能力。图 5 所示为各测试函数及

收敛曲线的可视化结果, 图中的 x, y, z 轴仅表示函数数值, 量纲为 1。设各种群大小均为 30, 迭代次数为 100, 维度为 30, 实验次数为 20。CFBES 算法中的辐射系数 λ 取 30, 调节常数 ζ 取 20, 增值步长 b 取 1.5。采用平均适应度、标准差来定量评价算法的优化性能。

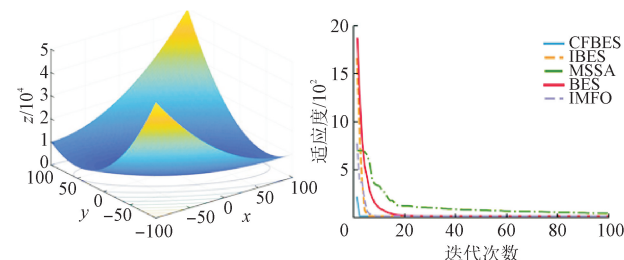
由图 5(a)~(d) 的收敛曲线可见, 对于单峰测试函数, CFBES 算法的收敛速度最快, 即同一时刻 CFBES 算法获得的适应度最小, 相比改进前的 BES 算法在搜索性能方面有较大的提升。由图 5(e)~(h) 的多峰测试函数迭代曲线可以看出, CFBES 算法能够较早脱离局部最优, 在全局寻优能力方面有着不错的表现。



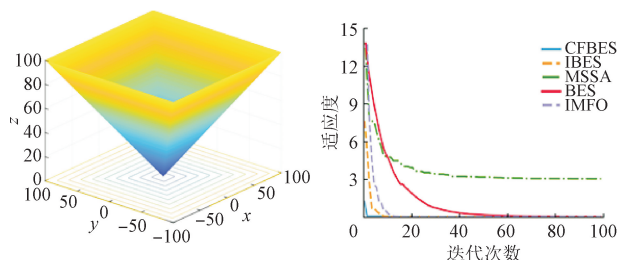
(a) F_1 函数及收敛曲线



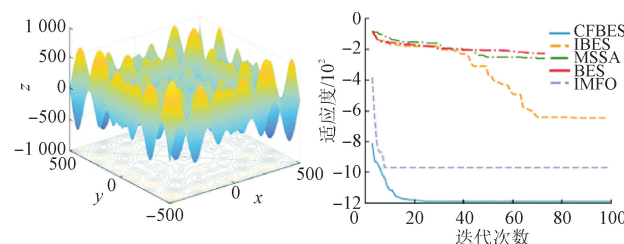
(b) F_2 函数及收敛曲线



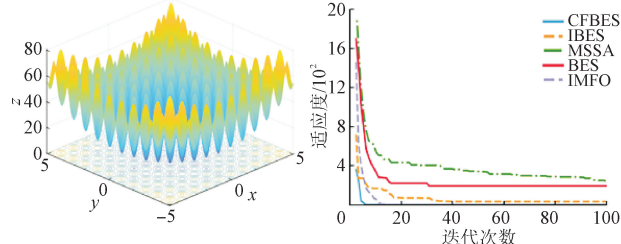
(c) F_3 函数及收敛曲线



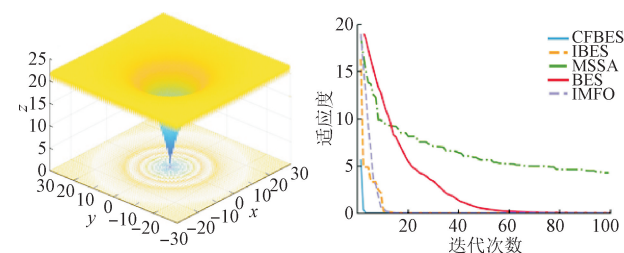
(d) F_4 函数及收敛曲线



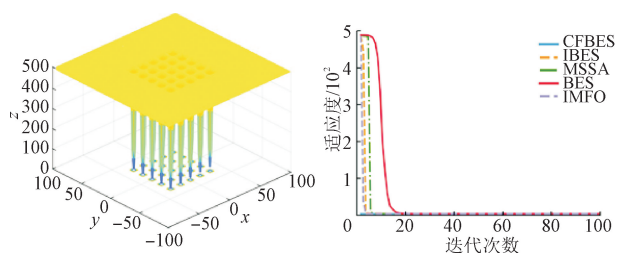
(e) F_8 函数及收敛曲线



(f) F_9 函数及收敛曲线



(g) F_{10} 函数及收敛曲线



(h) F_{14} 函数及收敛曲线

图 5 测试函数及收敛曲线可视化结果

Fig. 5 Search surfaces for different test function

表 1 给出了 CFBES 算法与其他优化算法的测试结果对比。由分析结果可知,在平均适应度方面,CFBES 算法在 8 个测试函数中寻优得到的平均适应度更接近

理论最优值,表明其收敛效果更佳,能够快速找到全局最优解。适应度标准差方面,CFBES 算法在选取的 8 个测试函数中寻优效果较其他算法而言更加稳定。

表 1 CFBES 算法与其他优化算法测试结果的对比
Table 1 Comparisons of test results of CFBES algorithm and other optimization algorithms

函数	性能参数	数值				
		IBES 算法	BES 算法	MSSA 算法	IMFO 算法	CFBES 算法
F_1	平均适应度	7.23×10^{-76}	1.30×10^{-23}	3.43×10^{-5}	4.32×10^{-46}	2.24×10^{-107}
	标准差	3.51×10^{-77}	5.68×10^{-24}	1.49×10^{-4}	6.72×10^{-47}	3.32×10^{-108}
F_2	平均适应度	4.31×10^{-38}	1.32×10^{-12}	1.18×10^{-23}	5.23×10^{-72}	4.41×10^{-121}
	标准差	6.53×10^{-39}	5.30×10^{-13}	3.11×10^{-24}	1.67×10^{-71}	5.83×10^{-122}
F_3	平均适应度	1.96×10^{-31}	1.10×10^{-23}	3.55×10^{-11}	5.61×10^{-44}	6.72×10^{-100}
	标准差	9.01×10^{-30}	4.80×10^{-24}	3.54×10^{-10}	4.83×10^{-43}	5.11×10^{-101}
F_4	平均适应度	2.81×10^{-45}	1.72	1.85×10^{-3}	9.72×10^{-36}	3.71×10^{-144}
	标准差	4.38×10^{-46}	5.23	5.57×10^{-3}	3.44×10^{-35}	4.33×10^{-143}
F_8	平均适应度	3.51×10^2	3.37×10^2	1.21×10^{-6}	2.83×10^3	7.28×10^4
	标准差	1.25×10^3	1.41×10^2	1.91×10^{-7}	1.76×10^3	2.38×10^4
F_9	平均适应度	4.73×10^{-44}	3.36×10^{-11}	8.89×10^{-23}	2.73×10^{-36}	3.21×10^{-71}
	标准差	3.81×10^{-43}	1.46×10^{-12}	3.87×10^{-24}	6.44×10^{-37}	8.77×10^{-72}
F_{10}	平均适应度	2.22×10^{-23}	1.00	3.01×10^{-2}	1.84×10^{-33}	2.22×10^{-102}
	标准差	9.67×10^{-22}	4.35	0.13	8.05×10^{-34}	4.44×10^{-101}
F_{14}	平均适应度	0.39	1.24×10^2	0.39	0.17	0.90
	标准差	0.68	2.16×10^2	0.68	0.65	0.73

4.2 CFBES-KMC 算法总体效果评价

为了评价本文提出的 CFBES-KMC 算法的改进效果,将其与改进飞蛾扑火 K 均值交叉迭代(IMFO-KMC)算法^[22]、K-means++算法^[23]、模糊 C 均值(FCM)聚类算法^[24]应用到 UCI 数据集中,用以比较各算法的优劣性能。算法的参数设置如下:各类种群大小均为 30,最大迭代次数为 100,FCM 聚类算法中的权重指数取 2.1。标准数据集如表 2 所示。图 6 分别给出了 CFBES-KMC、IMFO-KMC、K-means++、FCM 4 种算法在 Iris、Wine、CMC 和 Seeds 数据集上的适应度收敛曲线。

表 2 标准数据集特征

Table 2 Standard data set characteristics

数据集名称	样本数	属性维数	类别数	各类别样本数
Iris	150	4	3	505 050
Wine	178	13	3	597 148
CMC	1 473	9	3	629 333 511
Seeds	210	7	3	707 070

由图 6 可见,在 Iris 和 Wine 数据集测试中,CFBES-KMC 算法均具有比 IMFO-KMC 算法更快的收敛速度;在 CMC 和 Seeds 数据集上,CFBES-KMC

算法收敛效果明显优于参与对比的其他 3 种算法。由此可得,本文所改进的 CFBES-KMC 算法在收敛速度方面优于传统的 K-means++和 FCM 算法。

同时,为更客观地评价 CFBES-KMC 算法的改进效果,分别采用 Y_{Acc} 、 T_{ARI} 、 B_{NMI} 3 个指标来衡量不同聚类算法的性能^[25]。其中, Y_{Acc} 表示聚类的准确率,即比较聚类结果和真实结果之间的一致性; T_{ARI} 衡量的是两个数据分布的吻合程度,值越大意味着计算结果与真实值越相似; B_{NMI} 为归一化互信息,用来表示两组数据之间的关联程度。表 3 列出了 4 种算法在 3 个数据集上测试得到的评价指标以及单次迭代所需时间。由表 3 可知,CFBES-KMC 算法在 Iris 数据集上测得的 Y_{Acc} 指标高于其他 3 种算法,表明 CFBES-KMC 算法在复杂寻优过程中具有更好的鲁棒性,其主要是由于在秃鹰搜索过程中加入了竞争淘汰机制;在 Wine 数据集上,CFBES-KMC 算法的 Y_{Acc} 相较于 IMFO-KMC、FCM、K-means++算法,分别提高了 0.21%、2.48%和 8.83%;在 CMC 数据集中,CFBES-KMC 算法的 T_{ARI} 指标相较于其他算法提升了 0.26%~1.94%;与此同时,CFBES-KMC 算法在 Seeds 数据集上测

得的 Y_{Acc} 、 T_{ARI} 和 B_{NMI} 也均优于其他 3 种算法。

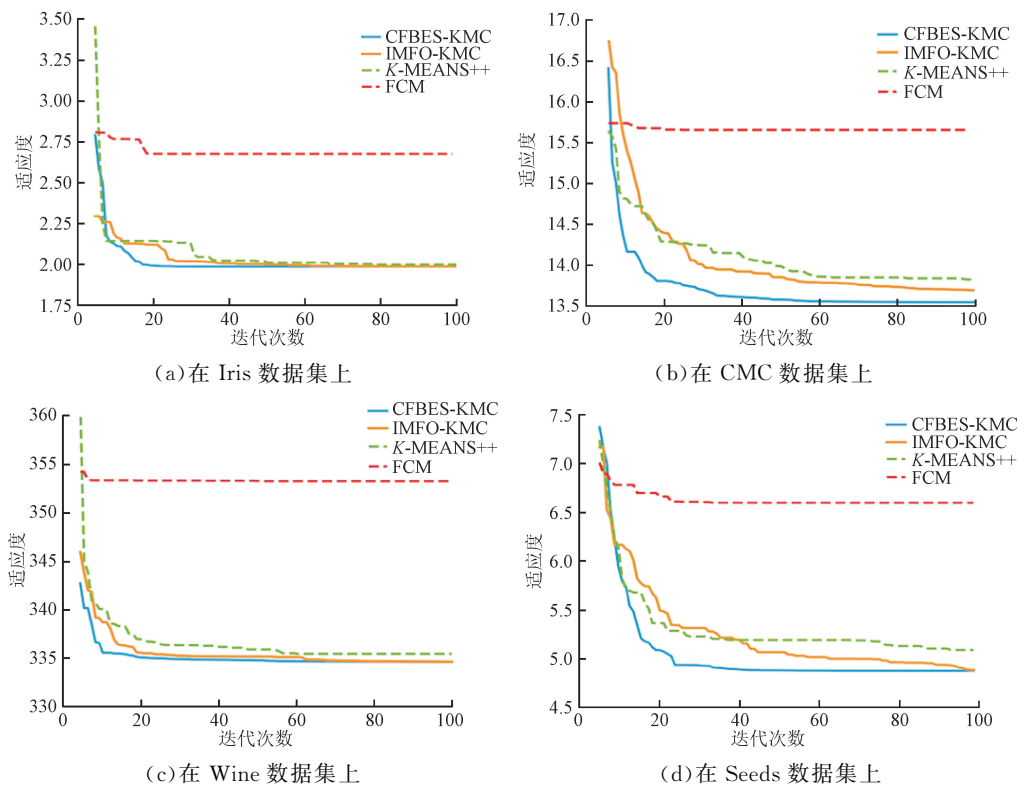


图 6 不同算法在 4 种数据集上的适应度对比
Fig. 6 Comparison of the fitness of different algorithms on 4 datasets

表 3 不同算法在 4 种数据集上的性能指标对比

Table 3 Comparison of the performance indicators of different algorithms on 4 datasets					
数据集	算法	Y_{Acc}	T_{ARI}	B_{NMI}	单次迭代时间/s
Iris	IMFO-KMC	0.894 3	0.743 7	0.763 6	0.040 1
	K-means++	0.885 2	0.721 8	0.716 3	0.005 3
	FCM	0.893 1	0.729 6	0.759 8	0.006 4
	CFBES-KMC	0.894 5	0.745 2	0.756 3	0.038 2
Wine	IMFO-KMC	0.707 9	0.371 5	0.419 3	0.037 8
	K-means++	0.651 8	0.349 2	0.403 1	0.005 0
	FCM	0.692 2	0.360 2	0.405 2	0.009 9
	CFBES-KMC	0.709 4	0.371 2	0.418 2	0.034 1
CMC	IMFO-KMC	0.712 3	0.371 2	0.425 5	0.306 1
	K-means++	0.566 3	0.365 1	0.418 1	0.040 0
	FCM	0.700 2	0.368 7	0.420 0	0.063 3
	CFBES-KMC	0.712 3	0.372 2	0.423 7	0.285 2
Seeds	IMFO-KMC	0.895 4	0.716 5	0.703 3	0.210 7
	K-means++	0.892 4	0.712 9	0.693 1	0.007 5
	FCM	0.894 3	0.714 6	0.694 2	0.008 9
	CFBES-KMC	0.895 4	0.717 0	0.706 3	0.162 4

在 4 种测试数据集上, K -means++ 和 FCM 算法单次迭代所需的时间均少于 CFBES-KMC 算法, 而 CFBES-KMC 算法所消耗的时间相较于 IMFO-KMC 算法有不同程度的提升。其中, 在 Wine 和 CMC 数据集上, CFBES-KMC 算法的单次迭代时间相较于 IMFO-KMC 算法, 分别节省了 9.79%、6.83%。

由上述实验结果可知,KMC 算法思路简单,聚类精度低,随机选取聚类中心致使算法稳定性差;FCM 算法由于涉及到模糊控制,迭代曲线趋于平滑,但聚类精度不高;IMFO-KMC 算法相较于 CFBES-KMC 算法,需要花费更多的时间才能收敛到足够精度;而 CFBES-KMC 算法由于采取了竞争淘汰机制,使得聚类算法不易陷入局部最优,稳定性较强,相比于其他算法聚类准确率更高。

4.3 基于信息熵的 CFBES-KMC 聚类三维点云简化结果分析

实验采用常见的三维点云数据格式,选取兔、椅

子和马作为点云数据模型进行三维点云的简化,得到原始点云、聚类结果、简化后点云的可视化图像,如图 7~图 9 所示。由图可见,聚类结果保留了原始点云的结构特征,将模型主体结构划分为同一聚类簇,信息熵占比较大,避免了信息熵过低导致的结构性特征删除;而冗余的细节特征表现为离散的小型聚类簇,此类冗余点云数据由于信息熵过低,可在后期被过滤以达到删除非结构性特征的目的,从而实现三维点云的简化。表 4 给出了不同三维点云模型简化后的量化结果,统计数据为 20 次独立实验得到的平均值。

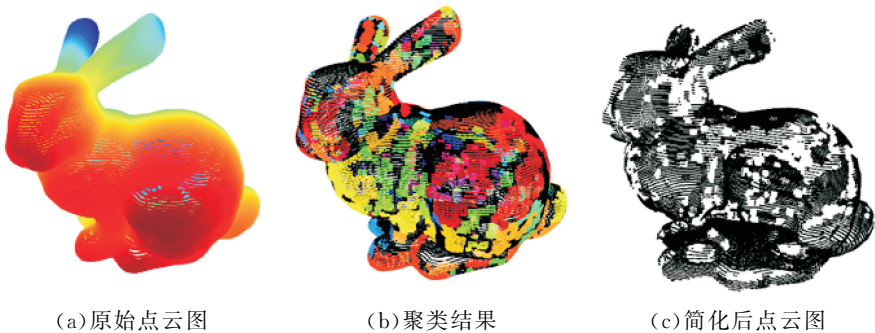


图 7 兔数据的点云简化效果
Fig. 7 Simplification effect of point cloud on rabbit dataset



图 8 椅子数据集的点云简化效果
Fig. 8 Simplification effect of point cloud on chair dataset



图 9 马数据集的点云简化效果
Fig. 9 Simplification effect of point cloud on horse dataset

表 4 不同三维点云模型简化结果
Table 4 Simplified results of different 3D point cloud models

模型	原始 点云数量	简化后 点云数量	原始 簇的数量	删除后 簇的数量	平均欧氏 距离	Hausdorff 距离	算法运行 时间/s
兔	35 947	21 347	807	97	0.018 1	0.002 7	0.006 3
椅子	49 960	32 748	206	18	0.009 5	0.007 1	0.008 1
马	48 485	29 716	318	54	0.022 6	0.005 3	0.007 2

由表 4 可知,改进后的简化算法在删除点数和简化表面之间的误差方面有着较好的表现,原始表面和使用改进后方法得到的简化表面之间的全局、局部误差均较小;在运行时间方面,改进后的点云简化算法的运行速度能够满足实时性要求,且在有效滤除冗余点云的基础上保留了原始模型的纹理特性和几何特征,实现了点云数据模型的有效简化。

5 结 论

本文通过混沌映射对秃鹰种群的初始化进行优化,并在搜索阶段加入竞争淘汰机制对秃鹰种群进行筛选,提高了 BES 算法的搜索精度和全局寻优能力,然后通过 CFBES-KMC 算法实现点云数据的聚类,在 k -NN 实现点云簇密度估计的基础上,结合香农熵实现点云信息量化,删除量化值小于阈值的聚类簇,从而完成了点云数据的简化。仿真结果表明:在标准测试函数上进行优化性能分析时,CFBES 算法的寻优性能优于参与比较的其它算法;与同类型的 IMFO-KMC、 K -means++、FCM 算法的聚类效果相比,CFBES-KMC 算法的聚类准确率分别提高了 1.02%、12.31%、14.72%;提出的 IBESSA 点云简化算法在有效滤除冗余点云的基础上保留了原本点云的细节和形状特征,不失为一种高效的点云简化算法。

参考文献:

[1] YUAN Hui, ZHANG Dexiang, WANG Weiwei, et al. A sampling-based 3D point cloud compression algorithm for immersive communication [J]. Mobile Networks and Applications, 2020, 25(5): 1863-1872.

[2] 陈辉,黄晓铭,刘万泉. 基于动态网格 k 邻域搜索的激光点云精简算法 [J]. 控制与决策, 2020, 35(12): 2986-2992.

CHEN Hui, HUANG Xiaoming, LIU Wanquan. Laser point cloud simplification algorithm based on dynamic grid k -nearest neighbors searching [J]. Control and Decision, 2020, 35(12): 2986-2992.

[3] 王建强,樊彦国,李国胜,等. 基于多参数 k -means 聚

类的自适应点云精简 [J]. 激光与光电子学进展, 2021, 58(6): 167-175.

WANG Jianqiang, FAN Yanguo, LI Guosheng, et al. Adaptive point cloud reduction based on multi parameter k -means clustering [J]. Laser & Optoelectronics Progress, 2021, 58(6): 167-175.

[4] DING Xiaoying, LIN Weisi, CHEN Zhenzhong, et al. Point cloud saliency detection by local and global feature fusion [J]. IEEE Transactions on Image Processing, 2019, 28(11): 5379-5393.

[5] PAULY M, GROSS M, KOBBELT L P. Efficient simplification of point-sampled surfaces [C]//2002 Proceedings of the IEEE Visualization (VIS). Boston, MA, USA: IEEE, 2002: 163-170.

[6] XUAN Wei, HUA Xianghong, CHEN Xijiang, et al. A new progressive simplification method for point cloud using local entropy of normal angle [J]. Journal of the Indian Society of Remote Sensing, 2018, 46(4): 581-589.

[7] MAHDAOUI A, SBAI E H. 3D point cloud simplification based on k -nearest neighbor and clustering [J]. Advances in Multimedia, 2020, 2020: 8825205.

[8] MA Tong, SHEN Wei. Notice of removal: research on a hybrid k -means clustering algorithm based on improved genetic algorithm [C]//2018 International Computers, Signals and Systems Conference (ICOMSSC). Piscataway, NJ, USA: IEEE, 2018: 228-233.

[9] HUA Chun. A quantum-inspired particle swarm optimization k -means++ clustering algorithm [C]//2021 IEEE Symposium Series on Computational Intelligence (SSCI). Piscataway, NJ, USA: IEEE, 2021: 1-6.

[10] ALSATTAR H A, ZAIDAN A A, ZAIDAN B B. Novel meta-heuristic bald eagle search optimisation algorithm [J]. Artificial Intelligence Review, 2020, 53(3): 2237-2264.

[11] 蒋宇飞,许贤泽,徐逢秋,等. 多策略融合改进的自适应蜉蝣算法 [J/OL]. 北京航空航天大学学报, 2022: 1-14[2022-11-23]. <https://doi.org/10.13700/j.bh.1001-5965.2022.0492>.

JIANG Yufei, XU Xianze, XU Fengqiu, et al. Adap-

- tive ephemera algorithm with improved multi-strategy fusion [J/OL]. Journal of Beijing University of Aeronautics and Astronautics, 2022: 1-14 [2022-11-23]. <https://doi.org/10.13700/j. bh. 1001-5965. 2022. 0492>.
- [12] MAHDAOUI A, BOUAZI A, HSAINI A M, et al. Comparison of k -means and fuzzy c -means algorithms on simplification of 3D point cloud based on entropy estimation [J]. Advances in Science, Technology and Engineering Systems Journal, 2017, 2(5): 38-44.
- [13] DING Xinghao, HE Fujin, LIN Zhirui, et al. Crowd density estimation using fusion of multi-layer features [J]. IEEE Transactions on Intelligent Transportation Systems, 2021, 22(8): 4776-4787.
- [14] ZHOU Zhou, SI Gangquan, SUN Haodong, et al. A robust clustering algorithm based on the identification of core points and KNN kernel density estimation [J]. Expert Systems with Applications, 2022, 195: 116573.
- [15] GONDELACH D J, LINARES R. Real-time thermospheric density estimation via radar and GPS tracking data assimilation [J]. Space Weather, 2021, 19(4): e2020SW002620.
- [16] THARWAT A. Parameter investigation of support vector machine classifier with kernel functions [J]. Knowledge and Information Systems, 2019, 61(3): 1269-1302.
- [17] CINCOTTA P M, GIORDANO C M, ALVES SILVA R, et al. The Shannon entropy: an efficient indicator of dynamical stability [J]. Physica: D Nonlinear Phenomena, 2021, 417: 132816.
- [18] TOLEDO F J, GALIANO V, BLANES J M, et al. Photovoltaic single-diode model parametrization: an application to the calculus of the Euclidean distance to an $I-V$ curve [J/OL]. Mathematics and Computers in Simulation, 2023: 1-26 [2023-03-10]. <https://doi.org/10.1016/j. matcom. 2023. 01. 005>.
- [19] 杨清凤, 游志胜, 张先玉. 基于豪斯多夫距离的快速多人脸检测算法 [J]. 电子科技大学学报, 2004, 33(4): 407-409.
- YANG Qingsu, YOU Zhisheng, ZHANG Xianyu. Fast multi-face detection algorithm based on Hausdorff distance [J]. Journal of University of Electronic Science and Technology of China, 2004, 33(4): 407-409.
- [20] 郭云川, 张长胜, 段青娜, 等. 融合多策略的改进秃鹰搜索算法 [J]. 控制与决策, 2024, 39(1): 69-77.
- GUO Yunchuan, ZHANG Changsheng, DUAN Qingna, et al. Improved bald eagle search algorithm fused with multiple strategies [J]. Control and Decision, 2024, 39(1): 69-77.
- [21] 刘卫明, 崔瑜, 毛伊敏, 等. 基于 MapReduce 和 MSSA 的并行 k -means 算法 [J]. 计算机应用研究, 2022, 39(11): 3244-3251, 3257.
- LIU Weiming, CUI Yu, MAO Yimin, et al. Parallel k -means algorithm based on MapReduce and MSSA [J]. Application Research of Computers, 2022, 39(11): 3244-3251, 3257.
- [22] 黄鹤, 李昕芮, 吴琨, 等. 引入改进飞蛾扑火的 K 均值交叉迭代聚类算法 [J]. 西安交通大学学报, 2020, 54(9): 32-39.
- HUANG He, LI Xinrui, WU Kun, et al. Hybrid iterative K -means clustering with improved moth-flame optimization [J]. Journal of Xi'an Jiaotong University, 2020, 54(9): 32-39.
- [23] NIU Lede, XIONG Liran. Optimisation and application research of ant colony algorithm in vehicle routing problem [J]. International Journal of Computing Science and Mathematics, 2021, 13(2): 177-193.
- [24] TIAN Hui. Research on robot optimal path planning method based on improved ant colony algorithm [J]. International Journal of Computing Science and Mathematics, 2021, 13(1): 80-92.
- [25] YU Jin, YOU Xiaoming, LIU Sheng. Dynamic reproductive ant colony algorithm based on piecewise clustering [J]. Applied Intelligence, 2021, 51(12): 8680-8700.

(编辑 李慧敏 刘杨)