

# 基于 KD-Tree 加速的多线激光传感器数据融合方法 \*

李欣飞, 鄢然, 夏磊, 赵青, 张凯飞

(重庆理工大学机械工程学院 重庆 400054)

**摘要:**针对多个线激光传感器协同扫描测量中大规模点云数据融合效率低、拼接误差大、处理复杂度高等问题,故提出一种基于 KD-Tree 加速的多个线激光传感器数据融合方法,通过动态邻域搜索策略和自适应半径调整机制,实现点云数据的高效排序与并行平滑优化。首先,构建 KD-Tree 空间索引结构,创新性地设计动态邻域搜索策略,实现二维轮廓数据从无序到有序的快速重组,算法时间复杂度由传统方法的  $O(n^2 \log n)$  降至  $O(n \log n)$ ;其次,结合 OpenMP 多线程并行计算技术改进移动最小二乘算法,提出 K-MLS 并行平滑方法,算法的时间复杂度从  $O(n^2)$  优化至  $O(n \log n)$ ,显著提升大规模点云的处理效率。在火车车轮测量系统中验证表明,当点云规模达到 209 万时,排序算法耗时较传统方法提升 35.7 倍,平滑算法耗时较传统方法提升 84.5 倍。最后,对比分析了该方法在提升点云质量方面的实际效果,算法可有效填补部分扫描数据的空缺,在轮辋面测量的最大偏差从  $\pm 0.279$  mm 降低至  $\pm 0.085$  mm,三维点云配准的均方误差由 0.323 mm 优化至 0.106 mm。实验数据表明,所提方法在保持亚毫米级精度的同时,显著提升了百万级点云数据的处理效率,有效解决了多传感器数据融合中的拼接误差、重叠区密度不均等问题,验证了算法在工业在线测量场景中的有效性与鲁棒性。

**关键词:** KD-Tree; 线激光传感器; 数据融合; 点云处理; 三维测量

中图分类号: TH74 文献标识码: A 国家标准学科分类代码: 460.4

## Data fusion method for multi-line laser sensor based on KD-Tree acceleration

Li Xunfei, Yan Ran, Xia Lei, Zhao Qing, Zhang Kaifei

(College of Mechanical Engineering, Chongqing University of Technology, Chongqing 400054, China)

**Abstract:** To address the challenges of low fusion efficiency, significant stitching errors, and high processing complexity in large-scale point cloud data fusion during collaborative scanning with multiple line laser sensors, this paper proposes a KD-Tree-accelerated data fusion method for multi-line laser sensor systems. The method leverages a dynamic neighborhood search strategy and an adaptive radius adjustment mechanism to enable efficient ordering and parallel smoothing optimization of point cloud data. First, a KD-Tree spatial indexing structure is constructed, and an innovative dynamic neighborhood search strategy is designed to rapidly reorganize disordered 2D contour data into ordered sequences, reducing the algorithm's time complexity from the traditional  $O(n^2 \log n)$  to  $O(n \log n)$ . Second, by integrating OpenMP multi-threaded parallel computing with an improved Moving Least Squares algorithm, a K-MLS parallel smoothing method is proposed, optimizing the time complexity from  $O(n^2)$  to  $O(n \log n)$ , significantly enhancing the processing efficiency for large-scale point clouds. The proposed method was validated in a train wheel measurement system. When processing 2.09 million points, the sorting algorithm achieved a 35.7-fold speedup compared to traditional approaches, while the smoothing algorithm exhibited an 84.5-fold performance improvement. Comparative analysis further demonstrates the method's effectiveness in improving point cloud quality: it successfully fills data gaps, reduces the maximum deviation in wheel tread measurement from  $\pm 0.279$  mm to  $\pm 0.085$  mm, and lowers the mean square error of 3D point cloud registration from 0.323 mm to 0.106 mm. Experimental results confirm that the proposed method maintains sub-millimeter accuracy while significantly boosting processing efficiency for million-scale point clouds. It effectively addresses issues such as stitching errors and uneven point density in overlapping regions during multi-sensor data fusion, proving its robustness and applicability in industrial online measurement scenarios.

**Keywords:** KD-Tree; line laser sensor; data fusion; point cloud processing; 3D measurement

## 0 引言

在大型回转体工件的高精度三维测量领域,多个线激光传感器协同扫描技术因其高效率和非接触式测量的优势被广泛应用<sup>[1,6]</sup>。采用多个线激光传感器共同测量待测工件的一个断面轮廓,然后利用转台旋转进行逐帧扫描,可以完整地获取到工件三维几何信息<sup>[7-8]</sup>。各个线激光传感器采集的轮廓数据分别处于不同坐标系,通过外部标定矩阵可以统一到同一个坐标系下<sup>[9]</sup>,生成一个完整的断面轮廓。然而,多个传感器数据融合中存在受装配及坐标系差异导致的拼接误差<sup>[10]</sup>;扫描区域之间存在重合,重叠区域点云数据的无序性和密度不均引发的后续处理难题<sup>[11]</sup>,不利于三维点云处理与重建。因此,需要对原始扫描数据进行重构处理,主要包含滤波、排序、平滑等步骤<sup>[12]</sup>。在进行最近邻搜索等基本问题时,算法会随着点云规模的增大而效率低下,无法满足实际工程需求。以火车车轮为例,通常直径范围在 600~1 300 mm,重量在 500~1 000 kg,是典型的轴对称回转体工件,其断面轮廓具有闭合连续、曲率变化大等特征,需要采用多线激光传感器协同扫描的方案,传统方法难以满足其在线测量时对效率与精度的双重需求。国内外学者针对此类问题开展了大量研究,例如,马伯瑞等<sup>[13]</sup>提出了一种基于局部平均的轮廓点云精简方法,用于线激光技术在钢轨磨耗测量中的降噪处理。王曦等<sup>[14]</sup>提出了一种权重局部最优投影算法,在点云精简的同时保留点云的关键特征。张立斌等<sup>[15]</sup>基于 K 维树 (K-dimensional tree, KD-Tree) 建立点云空间的关系,采用邻域平均法实现对点云数据进行降噪,借助最小二乘法对点云数据进行精简。Fuangkhon<sup>[16]</sup>在数据缩减算法中引入 KD-Tree 代替欧几里得距离函数,显著减少大规模数据集缩减的运行时间。Chatterjee 等<sup>[17]</sup>提出了 LockFree-KD-tree, 实现并行可线性化的最近邻搜索方法。樊爽等<sup>[18]</sup>提出了逐点移动最小二乘 (pointwise moving least square, PMLS) 拟合方法,对感兴趣点局部采用极坐标计算提高移动最小二乘 (moving least square, MLS) 的计算效率和稳定性,用于对数字图像得到的应变场进行拟合。

综上,针对多线激光旋转扫描测量场景下的点云重构问题,提出一种基于 KD-Tree 加速的二维轮廓数据重构方法,利用 KD-Tree 空间索引的动态邻域搜索策略,结合自适应半径调整机制,实现点云数据的快速排序与平滑优化,显著提升多传感器数据融合效率,从而提高二维轮廓数据的质量和准确性。

本研究主要贡献主要包括:提出基于 KD-Tree 的动态邻域搜索算法,将点云排序的时间复杂度从  $O(n^2 \log n)$  降至  $O(n \log n)$ , 在扫描过程中实时高效率解决重叠点云的无序问题;设计 K-MLS 并行平滑算法,融

合 KD-Tree 邻域检索与 OpenMP 多线程技术,优化传统 MLS 算法的计算效率,在扫描过程实时高效率对每一帧数据进行融合处理;构建火车车轮测量系统,通过工业级实验验证了算法在实际场景中的有效性和鲁棒性。

## 1 多传感器数据融合方法

如图 1 所示,为一个基于线激光旋转扫描的火车车轮三维测量系统,系统采用了 4 个线激光传感器同步扫描车轮的一个断面轮廓,通过旋转车轮实现 360° 全周测量。

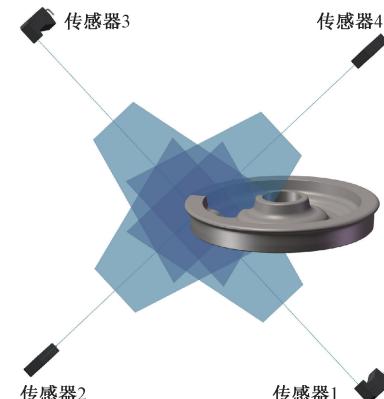


图 1 线激光火车车轮扫描测量系统

Fig. 1 Line laser train wheel scanning measurement system

如图 2 所示,为多线激光火车车轮测量系统测量的一个断面轮廓示意图。每个线激光传感器数据分别用不同灰度颜色表示,经过外部标定矩阵的变换后可以拼合为一个完整的断面轮廓。

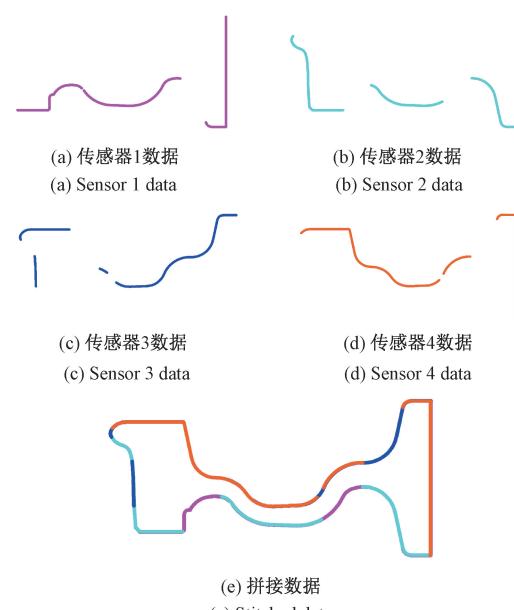


图 2 线激光传感器扫描数据示意图

Fig. 2 Schematic diagram of the scan data of the line laser sensor

在拼合的完整轮廓数据中,可以看出大部分的区域存在传感器的重叠,重叠的区域与轮廓曲线的形状有关,不同类型车轮的重叠区域存在不确定性。原始轮廓数据中点的顺序是分别按照单个传感器的数据进行独立排列的,这使得扫描的轮廓数据为无序点云,无法进行点云轮廓线规则化、轮廓曲线拟合等优化计算,交叠区点密度显著高于非重叠区,影响曲面重建精度。在大尺寸工件在线测量的场景下,点云规模大同时测量速度要求高,传统方法在解决此类问题时,循环结构计算方式会导致时间复杂度呈平方增长,出现效率低等问题。特别是多个线激光测量系统中,点云数据是逐帧扫描出来的,实时完成每一帧的数据融合,能够极大提升时间效率并提高点云数据的质量。

### 1.1 KD-Tree 结构与构建

KD-Tree 是一种基于空间划分的高效索引结构,广泛应用于最近邻搜索、范围查询等任务。核心思想是通过递归地将空间划分为互不重叠的超矩形区域,使得每个树节点对应一个子空间,并存储该子空间内的数据点或划分超平面。在二维轮廓数据排序与平滑任务中,KD-Tree 可以通过降低搜索复杂度显著提升算法效率。

给定一个点云数据集  $P = \{p_1, p_2, \dots, p_n\} \subset \mathbb{R}^2$ , 构建 KD-Tree 的步骤为:

1) 进行分割维度选择,从当前维度集合中选择方差最大的维度为分割轴,以最大化划分后的空间差异性。设当前待分割点集为  $P'$ , 分割轴  $d$  为:

$$d = \operatorname{argmax}_{k \in \{x, y\}} \operatorname{Var}(P'_k) \quad (1)$$

其中,  $\operatorname{Var}(P'_k)$  表示点集在维度  $k$  上的方差。

2) 沿分割轴  $d$  选择中位数点  $p_{med}^{(d)}$  作为分割点,将点集划分为左右子树,确保树结构的平衡性。分割点满足:

$$|\{p_j \in P' \mid p_j^{(d)} \leq p_{med}^{(d)}\}| = \frac{|P'|}{2} \quad (2)$$

3) 对左右子点集重复 1)~2),直到叶子节点包含单个点或满足预设终止条件。

### 1.2 二维数据排序

现有的点云轮廓排序方法例如,基于中心点旋转角的极坐标排序只能适用于简单的轮廓形状,不能适用于存在凹凸的形状<sup>[19]</sup>;旅行商问题求解的路径目标是总路程的最小值<sup>[20]</sup>,而不是扫描的实际轮廓曲线;迭代选择欧氏距离最近点排序<sup>[21]</sup>,高频次的全排序操作导致时间复杂度呈超线性增长。为此,提出了一种基于 K 维树空间索引加速近邻搜索的轮廓数据点云排序方法,通过构建拼合点云的 KD-Tree,动态调整搜索半径,寻找未处理的最近邻点,利用空间划分的方法将搜索的时间复杂度由指数级降为对数级,可以大大提升多线激光传感器数据拼合排序的效率。

给定系统扫描的一个无序轮廓点云  $P = \{p_1, p_2, \dots,$

$p_n\} \subset \mathbb{R}^2$ , 目标生成有序点云序列  $S = (p_{s_1}, p_{s_2}, \dots, p_{s_n})$ , 对点云数据进行重组排序,相邻点  $p_{s_i}$  与  $p_{s_{i+1}}$  满足空间邻域性。点  $p_i$  的最近  $k$  个邻居点集合,由 KD-Tree 查询的结果为:

$$N_k(p_i) = \operatorname{argmin}_{p_j \in Q} \|p_i - p_j\|^2 \quad (3)$$

式中:  $Q$  为点云  $P$  的子集,大小为  $k$ ;  $\|p_i - p_j\|$  表示点  $p_i$  与  $p_j$  的欧式距离;  $\operatorname{argmin}$  表示返回目标函数最小的  $Q$ ,即距离平方和最小的  $k$  个点。

从初始点  $p_{s_i} = p_i$  开始,迭代选择未访问的最近邻点的索引

$$s_{i+1} = \operatorname{argmin}_{j \notin \{s_1, \dots, s_i\}} \|p_{s_i} - p_j\| \quad (4)$$

式中:  $s_i$  表示已经排序的点云序列的索引,索引  $j$  未在已排序的点云序列中出现。

在点云排序过程中,搜索范围  $k$  的取值是算法的核心参数。若取值过小,在搜索过程中可能因点云的密度不均导致算法停滞,例如某些区域的点密度较低,  $k$  可能无法找到未访问的点,导致路径断裂。若直接设置为全局搜索,即  $k = n$ ,单次查询的时间复杂度为  $O(n)$ ,效率低下。为此,提出了一种动态调整搜索范围的策略,仅在必要时扩大  $k$ ,大部分情况下使用较小的  $k$  完成搜索。在排序过程中,若最近邻点已经被访问过,则扩展其搜索范围,直至找到未访问的点或遍历完成。该策略通过局部搜索与动态扩展的协同机制,在稀疏区域避免路径断裂,同时维持整体效率。搜索范围  $k$  按照以下规则进行动态调整

$$k^{(t+1)} = \begin{cases} k^{(t)}, & j \in N_{k^{(t)}}(p_{s_i}) \wedge j \notin S_t \\ \min(2k^{(t)}, n), & \text{其他} \end{cases} \quad (5)$$

式中:  $t$  为迭代次数,表示动态调整搜索范围的步骤;  $S_t = \{s_1, s_2, \dots, s_t\}$ , 为前  $t$  步已选点的集合。搜索范围  $k$  按照几何级数增长,但不超过点云总数  $n$ 。

基于全排序的贪心算法进行计算时,设点云规模为  $n$ ,每次迭代处理剩余的  $m$  个点时的时间复杂度为  $O(m \log m)$ ,则总的时间复杂度为:

$$T_c(n) = \sum_{m=1}^{n-1} m \log m \approx \int_1^n x \log x dx = O(n^2 \log n) \quad (6)$$

利用 KD-Tree 空间索引进行加速近邻搜索计算时,构建点云全局 KD 树的时间复杂度为  $O(n \log n)$ 。设第  $i$  次迭代中,动态调整搜索范围的次数为  $m_i$ 。每次调整后的搜索范围为  $k_j (j = 1, 2, \dots, m_i)$ ,则单次迭代的时间复杂度为:

$$T_i = \sum_{j=1}^{m_i} (O(k_j) + O(\log n)) \quad (7)$$

式中:  $O(k_j)$  为处理  $k_j$  个候选点的时间;  $O(\log n)$  为 KD-Tree 的最近邻查询时间。

初始化时构建点云全局 KD 树的时间复杂度为  $O(n \log n)$ , 则总的时间复杂度为:

$$T_k = O(n \log n) + \sum_{i=1}^{n-1} \sum_{j=1}^{m_i} (O(k_j) + O(\log n)) \quad (8)$$

考虑一般情况下, 待排序的点云存在局部稀疏的区域, 需要动态扩展搜索范围  $k$ 。平均扩展次数  $m$  与输入点云的规模  $n$  无关, 其时间复杂度为  $O(1)$ 。 $k_j = 2^j$  为几何级数增长, 则此时 KD-Tree 加速算法的时间复杂度为:

$$T_k = O(n \log n) \quad (9)$$

如表 1 所示, 对比了基于 KD-Tree 的动态近邻搜索策略与传统迭代最近邻算法的性能差异。

表 1 排序算法对比

Table 1 Comparison of sorting algorithms

特征	迭代最近点算法	KD-Tree 加速算法
时间复杂度	$O(n^2 \log n)$	$O(n \log n)$
空间复杂度	$O(n)$	$O(n)$
动态搜索机制	固定搜索范围	几何级数自适应扩展
使用场景	小规模点云	大规模点云

在时间复杂度方面, 通过将全局搜索转化为层次化局部搜索, 在空间划分特性与动态调整策略的协同作用下, 计算复杂度从  $O(n^2 \log n)$  降低至  $O(n \log n)$ 。空间复杂度方面, 尽管需要维护 KD-Tree 索引结构, 但其存储开销仍保持在  $O(n)$  量级。算法的核心创新体现在动态搜索机制设计上, 通过如式(5)所示定义的几何级数扩展规则, 实现了搜索半径的自适应调节。这种机制在保证稀疏区域连通性的同时, 避免了全局搜索带来的计算浪费, 对点云密度变化具有更强的适应性。

### 1.3 K-MLS 平滑

点云平滑是进行三维重建以及尺寸测量的关键预处理步骤, 可以补偿大量程线激光传感器线性度较差的问题, 降低噪声, 有效提高点云数据的质量。点云平滑处理中的移动最小二乘算法存在处理速度慢、效率低<sup>[22]</sup>, 无法满足工业在线实时测量需要等问题。为此, 提出一种基于 KD-Tree 与 OpenMP 并行化的改进型 MLS 点云平滑算法(KD-tree moving least square, K-MLS), 其核心思想是利用 KD-Tree 空间索引结构降低传统 MLS 算法的全局遍历复杂度, 结合半径搜索策略实现局部邻域的快速定位, 利用高斯权重函数补偿线激光传感器非线性误差, 实现大规模点云的高效处理。算法包含 3 个主要阶段: 空间索引构建、邻域自适应搜索和加权最小二乘优化。相较于传统 MLS 算法  $O(n^2)$  的时间复杂度, KD-Tree 加速算法将复杂度降低至  $O(n \log n)$  量级。

给定一个点云数据集  $P = \{p_i\}_{i=1}^n \in \mathbb{R}^2$ , 传统的 MLS 方法通过对每个点  $p_i$  的邻域进行加权最小二乘拟合实现平滑。对于任意点  $p_i$ , MLS 方法进行平滑后的位置  $q_i$  可表示为:

$$q_i = \frac{\sum_{j=1}^n \omega(\|p_i - p_j\|) \cdot p_j}{\sum_{j=1}^n \omega(\|p_i - p_j\|)} \quad (10)$$

其中, 权重函数  $\omega(d)$  采用高斯核函数:

$$\omega(d) = e^{-\frac{d^2}{2\sigma^2}} \quad (11)$$

式中:  $\sigma$  为平滑系数, 用来控制邻域影响范围;  $d = \|p_i - p_j\|$  表示点间距,  $p_j$  是满足半径约束的局部邻域点。

传统 MLS 方法实现时需要计算所有点对之间的欧式距离, 导致时间复杂度为  $O(n^2)$ 。当处理大规模点云时, 同样存在计算效率急剧下降的问题。为降低计算复杂度, 引入了 KD-Tree 结构, 采用快速近似最近邻库(fast library for approximate nearest neighbors, FLANN)进行构建, 时间复杂度为  $O(n \log n)$ , 对构建过程可形式化表示为:

$$T_{KDtree} = \text{BuildKDTree}(P) \quad (12)$$

其中, 每个数节点存储空间划分超平面和子节点指针, 叶节点包含点集子集。

针对每个查询点  $p_i$ , 以  $r = 3\sigma$  为搜索半径覆盖 99.7% 的高斯分布区域, 执行范围搜索, 即:

$$N_r(p_i) = \{p_j \in P \mid \|p_i - p_j\| \leq r\} \quad (13)$$

其中, 搜索半径  $r = \sigma$ , 与平滑系数直接关联。

通过加权平均得到平滑后的点坐标为:

$$q_i = \frac{\sum_{p_j \in N_r(p_i)} \omega_{ij} p_j}{\sum_{p_j \in N_r(p_i)} \omega_{ij}} \quad (14)$$

式中:  $\omega_{ij}$  为  $p_i$  对每个邻域点  $p_j \in N_r(p_i)$  的高斯权重, 距离  $p_i$  越近的邻域点  $p_j$ , 其权重越大, 对平滑结果的贡献越明显。

采用 OpenMP<sup>[23]</sup> 实现多线程并行处理, 根据 CPU 的线程数  $T$ , 可将点云划分为  $T$  个独立任务单元。并行化计算过程描述为:

$$\begin{cases} \text{Parallel For } p_i \in P: \\ N_r(p_i) \leftarrow \text{RadiusSearch}(T_{KDtree}, r) \\ q_i \leftarrow \sum_{p_j \in N_r(p_i)} \omega_{ij} p_j / \sum_{p_j \in N_r(p_i)} \omega_{ij} \end{cases} \quad (15)$$

如表 2 所示, 提出的基于 KD-Tree 加速的移动最小二乘方法(K-MLS)与传统 MLS 算法在计算范式上存在显著差异, 主要体现在 4 个核心特征维度。

在邻域搜索机制方面, 传统 MLS 采用全局遍历策略, 通过计算当前点与点云全体样本的几何关系进行加权优化; 而 K-MLS 基于空间索引的局部邻域约束进行

表 2 平滑算法对比

Table 2 Comparison of smoothing algorithms

特征	传统 MLS	K-MLS
邻域范围	全体点云 ( $j = 1, \dots, n$ )	局部邻域 $p_j \in N_r(p_i)$
时间复杂度	$O(n^2)$	$O(n \log n)$
空间复杂度	$O(n)$	$O(kn)$
实现方式	双重循环遍历	KD-Tree 搜索+并行计算

计算,有效降低了冗余计算量。在时间复杂度方面,传统方法因双重循环嵌套在处理大量级点云时面临严重局限性;而 K-MLS 通过 KD-Tree 的层次化空间划分,在搜索过程中使用剪枝策略进行加速,将平滑算法的时间复杂度优化至  $O(n \log n)$  量级。空间复杂度方面,传统方法仅需考虑输入存储的空间复杂度  $O(n)$ ,K-MLS 方法需要存储树结构空间以及每个点的  $k$  个邻域点的索引与距离需占用  $O(kn)$  空间,但这一代价在现代计算设备的存储能力下可忽略不计。实现方式上,传统算法采用串行双重循环遍历结构,难以发挥多核处理器性能;K-MLS 则深度融合了 KD-Tree 的快速邻域检索与 OpenMP 多线程并行计算技术,通过指令级优化实现任务级并行化,显著提升了算法在分布式计算环境下的执行效率。

## 2 实验验证

### 2.1 实验环境

在开发的多线激光火车车轮三维测量系统中应用了如前文所述的基于 KD-Tree 加速的多线激光传感器数据融合方法进行实测验证,并与使用传统方法的版本进行对比。如图 3 所示,实验过程使用自主开发的火车车轮尺寸测量系统软件 (BSNWVision V1.0.5) 及其配套设备,测量装置中采用的线激光传感器为 4 个基恩士 LJ-X8900。

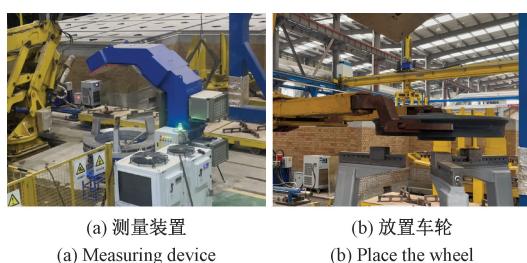


图 3 实验环境

Fig. 3 Experimental environment

使用测量系统扫描 BA319 车轮,采集数据集,每个数据集包含 360 帧断面轮廓,点云点数约为 2 000 000 个。算法运行的硬件环境 CPU 为 AMD Ryzen 7 4800H @ 2.9 GHz,

GPU 为 NVIDIA GeForce RTX 2060 Max-Q 6 GB, 内存为 32 GB, 3 200 MT/s。软件的开发基于 C++ 语言, 采用了模块化的设计方式。首先, 使用该数据集分别对二维轮廓数据排序算法和 K-MLS 平滑算法的效率进行了测试。算法的时间效率利用高精度计数器进行测量, 采用 C++ 标准库中的 `<chrono>` 高分辨率时钟, 分别于算法执行起始点与终止点记录时间戳, 最终耗时通过计算两个时间点的差值获得, 测量精度为毫秒级。为保证测量有效性, 每个测试用例均进行 5 次重复实验并取平均值。进一步地, 从扫描后拼接的二维断面轮廓和三维点云数据两个角度, 对比了应用该方法前后构建的数据质量, 验证了算法在补偿线激光传感器线性度、提升点云质量等方面的提升效果。

### 2.2 排序算法性能分析

采集的每个车轮数据集中包含 4 个线激光传感器单独采集的数据, 为进一步验证算法在不同点云数量下的运行效率, 对单个线激光传感器采集的原始数据进行了不同参数的均匀下采样滤波改变点云的密度, 测试不同情况下的算法运行效果。测试结果如表 3 所示, 统计的运行时间为该算法处理一个完整车轮数据的时间, 即算法对 360 帧轮廓数据计算的总时间。

表 3 排序算法对比

Table 3 Comparison of sorting algorithms

序号	点云数量	运行时间/ms	
		迭代最近点算法	KD-Tree 加速算法
1	301 613	2 658	502
2	365 549	4 158	635
3	438 506	6 122	877
4	523 817	9 280	1 051
5	617 739	12 872	1 315
6	719 158	18 736	1 547
7	951 103	34 246	2 133
8	1 140 621	50 821	2 718
9	1 752 591	123 384	4 116
10	2 098 818	181 731	5 093

实验数据表明, 基于 KD-Tree 加速的排序算法在各类点云规模下均展现出显著的效率优势。当处理最小规模点云 301 613 点时, KD-Tree 加速算法耗时仅 502 ms, 相比传统迭代最近点方法实现了 5.3 倍的加速效果。随着点云规模增长至 2 098 818 点, 加速比提升至 35.7 倍。在点云规模达到百万级时, 传统方法耗时已超过 50 s 而 KD-Tree 加速方法仍保持在秒级处理水平, 效率上的量级差异在热态车轮检测等需要在线自动化实时的工业场

景中具有重要的实际意义。如图 4、5 所示,从算法运行时间的曲线来看, KD-Tree 加速方法的运行时间增长趋势明显趋缓,其运行时间与点云数量间呈现近似  $O(n \log n)$  的复杂度特征,而传统方法则表现出接近明显的二次增长特性,这与两种算法的时间复杂度理论预期相吻合。

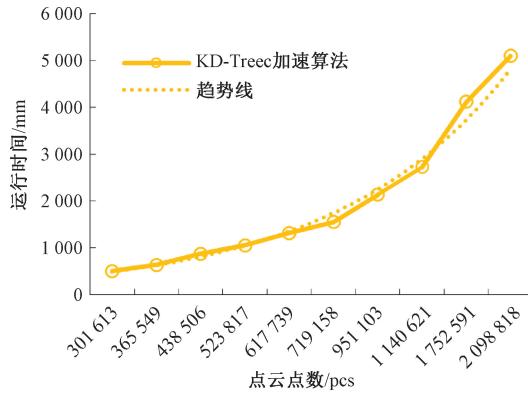


图 4 KD-Tree 加速算法耗时

Fig. 4 KD-Tree acceleration algorithm runtime

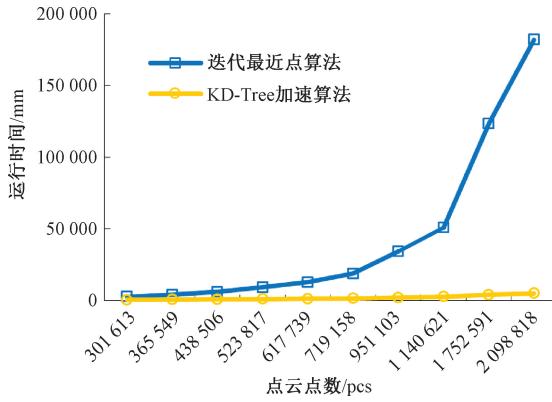


图 5 KD-Tree 加速算法与迭代最近点算法耗时对比

Fig. 5 Comparison of the running time of the KD-Tree acceleration algorithm and the iterative nearest point algorithm

### 2.3 平滑算法性能分析

采用与排序算法同样的测试方法,对平滑算法进行效率测试。对相同的数据集在算法参数相同的情况下分别使用传统 MLS 方法和 K-MLS 方法运行程序,结果如表 4 所示。

在点云平滑处理方面, K-MLS 算法展现出更加优异的性能特征。从数据可见,当点云规模在 30 万~70 万点时, K-MLS 算法处理耗时稳定在 3 500 ms 以内,而传统 MLS 方法处理时间从 6 429 ms 快速攀升至 45 488 ms。点云的空间分布特性可能触发算法的邻域自适应搜索机制,使得 K-MLS 算法耗时会出现下降现象。如图 6、7 所示,从对比曲线可以看出传统 MLS 方法的耗时呈现出

表 4 平滑算法对比  
Table 4 Comparison of smoothing algorithms

序号	点云数量	运行时间/ms	
		MLS	K-MLS
1	301 613	6 429	3 162
2	365 549	11 194	3 501
3	438 506	13 979	3 379
4	523 817	23 525	2 851
5	617 739	31 360	3 041
6	719 158	45 488	2 793
7	951 103	67 660	2 880
8	1 140 621	114 074	3 216
9	1 752 591	211 037	4 107
10	2 098 818	349 853	4 140

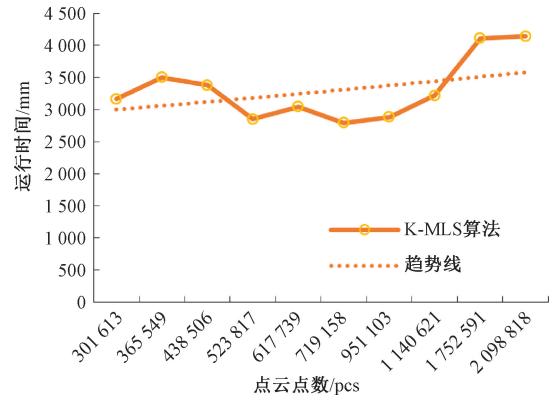


图 6 K-MLS 算法耗时

Fig. 6 K-MLS algorithm runtime

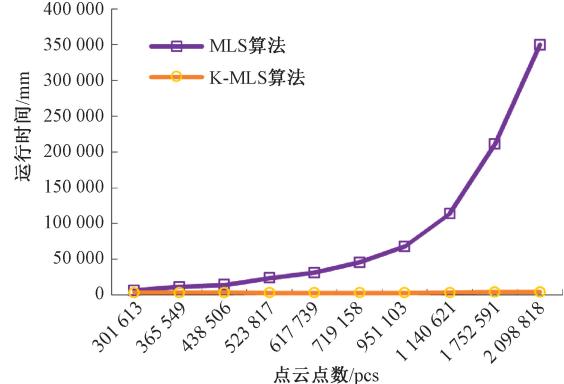


图 7 K-MLS 算法与 MLS 算法运行耗时对比

Fig. 7 Comparison of the running time of the K-MLS algorithm and the MLS algorithm

指数增长特性,K-MLS 算法的时间增长呈现出缓慢的近似线性特征,当处理最大规模点云时,K-MLS 算法耗时仅

4 140 ms, 相较传统 MLS 方法实现了 84.5 倍的加速比。实验数据验证了算法的工程可行性, 不仅降低了硬件配置需求, 更为在线质量检测提供了可靠的技术保障。

## 2.4 点云质量分析

在多线激光测量系统中, 点云数据构建过程为每个传感器原始数据拼接为一个完整轮廓, 然后逐帧扫描整个工件。原始数据的质量由线激光传感器硬件参数决定, 逐帧扫描的过程由机械系统的运动组件决定, 因此, 在排除外界干扰情况下, 多线激光传感器的数据融合算法是点云数据构建过程中对点云质量产生影响最大的环节。

如图 8 所示, 为扫描 BA319 车轮时每个线激光传感器的一帧实测数据, 使用外部标定矩阵变换到统一的坐标系后生成的完整轮廓。与前文所述的扫描数据示意图情况一致, 轮廓数据存在着若干重叠交叉区域, 如图 9 所示, 图 9(a)~(d) 分别对应图 8 中的①~④, 部分区域存在畸变导致的数据重叠、分层、毛刺等现象, 以及受车轮轮廓曲率和线激光传感器安装角度限制导致的数据空缺。由于轮廓数据为无序点云, 难以识别空缺数据并进行有效插值或连线补全。车轮在采用辗轧工艺成型时, 需要根据理论与实际回转面轮廓的差异对比来调整工艺参数, 此类问题在影响测量精度的同时往往会导致误判现象。

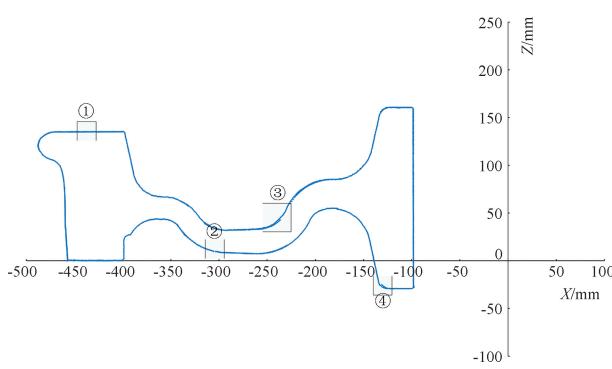
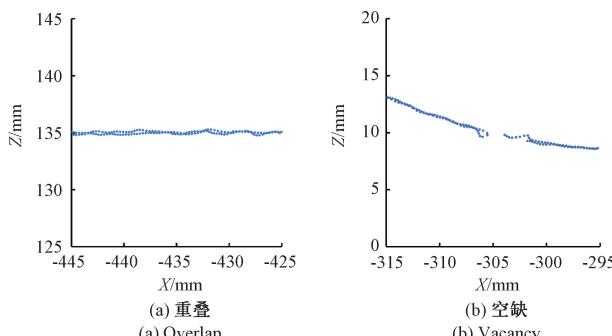


图 8 扫描的轮廓数据

Fig. 8 Scanned profile data



(a) 重叠

(a) Overlap

(b) 空缺

(b) Vacancy

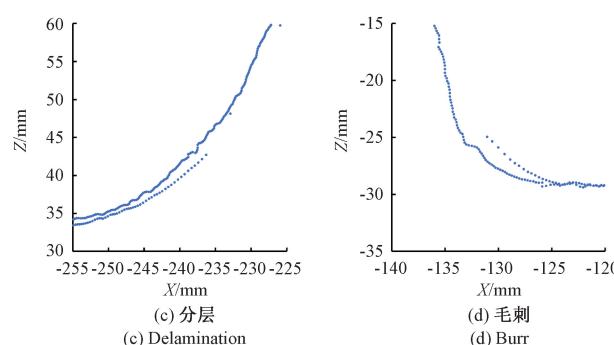


图 9 扫描数据的问题

Fig. 9 Problems with scanned data

如图 10 所示, 无序点云在直接进行连线时, 数据混乱, 无法有效补全空缺数据和拟合计算。如图 11 所示, 应用数据融合方法后, 可将扫描的实际回转面轮廓连接为一个闭合的完整曲线, 一定程度上弥补了扫描数据的空缺部位, 并在测量空缺部分的尺寸时可插值出关键点进行选取。

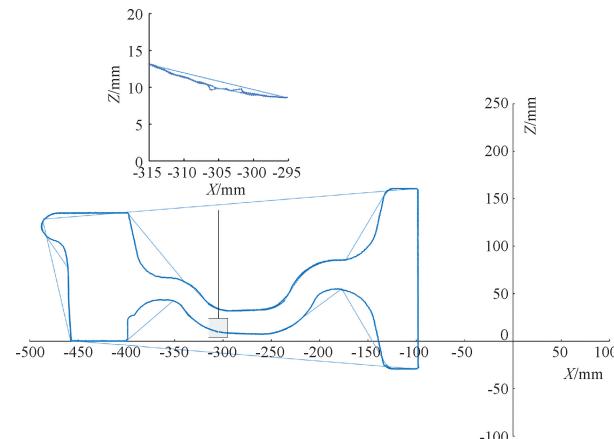


图 10 扫描数据连线图

Fig. 10 Line diagram of scan data

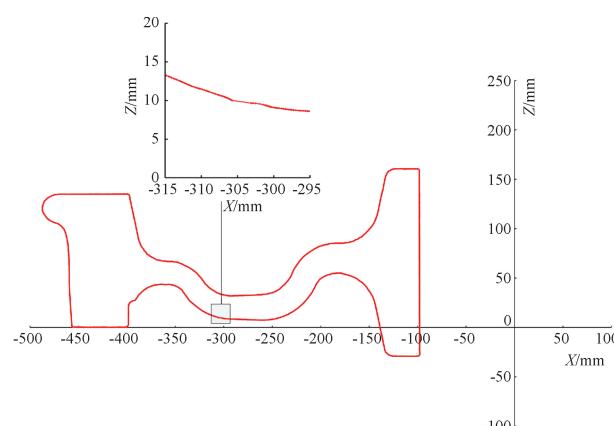


图 11 融合数据连线图

Fig. 11 Line diagram of the fused data

线激光传感器的线性度是在整个测量范围内相对于目标的实际距离与相对于目标的测量距离之间的差值,通常表示传感器的绝对距离测量能力。设计的线激光扫描系统中,传感器为倾角安装方式,扫描的轮廓数据在传感器视角下实际为不同距离的数据。因此,直接选取车轮内侧轮廓面处,分析扫描轮廓与实际轮廓的最大偏差,可反应出数据融合方法在补偿传感器线性度方面的效果。如图 12 所示,为应用该方法前后车轮内侧轮廓面扫描数据的对比图,传统方法的最大偏差为  $\pm 0.279$  mm,采用该方法后最大偏差降为  $\pm 0.085$  mm。

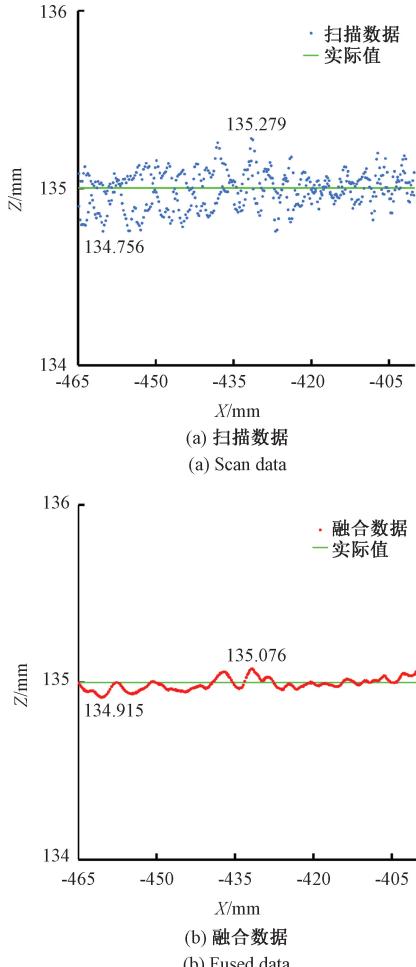
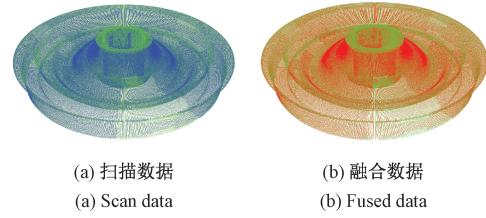


图 12 线性度对比

Fig. 12 Linearity comparison

为了对整体点云进行质量验证,需构建完整的三维点云数据。实验采用 HandySCAN BLACK<sup>TM</sup> 型手持式三维扫描仪,精度为 0.025 mm,采集该型号车轮,生成实际的三维点云数据。将线激光测量系统应用数据融合方法前后的构建出的三维点云分别与手持式三维扫描仪构建的三维点云进行迭代最近点 (iterative closest point, ICP) 配准。计算源点云配准后的每一个点到目标点云最近邻

点的距离平方,采用均方误差 (mean squared error, MSE) 作为评价指标来衡量其与目标点云的差异程度,其值越小表示配准精度越高,两个点云之间的几何差异越小。如图 13 所示,为点云配准结果。仅通过标定方式构建的点云数据的均方误差为 0.323 mm,采用数据融合方法后点云数据的均方误差为 0.106 mm。因此,应用提出的数据融合方法后得到的三维点云数据的配准结果更优,与实际工件的点云契合度更高,点云质量相较于仅通过标定方式构建的点云有显著提升。

图 13 点云配准  
Fig. 13 Point cloud registration

### 3 结 论

针对多线激光旋转扫描测量场景下的点云重构问题,提出基于 KD-Tree 加速的动态邻域搜索算法与 K-MLS 并行平滑算法,解决了大规模点云数据融合效率低、实时性差等难题。实验结果表明,数据融合方法在百万级点云处理中显著优于传统算法:排序算法的时间复杂度从  $O(n^2 \log n)$  降至  $O(n \log n)$ ,效率提升最高达 35.7 倍;K-MLS 算法通过 KD-Tree 邻域搜索与并行计算优化,平滑效率提升 84.5 倍,且耗时随数据规模增长呈近似线性趋势。实际工业测试验证了算法在火车车轮测量中的鲁棒性,为大型工件在线检测提供了高效可靠的技术方案。

该方法虽然以火车车轮作为研究对象,但其核心思想是对多个线激光传感器协同扫描的数据进行高效高质量的融合优化,满足高效率高质量的双重测量需求。而且,火车车轮是回转体中断面轮廓最为复杂的典型工件之一,算法设计中采用的动态邻域机制可自适应不同特征,因此技术框架同样适用于采用多个线激光传感器协同扫描一个完整轮廓,通过旋转或移动的方式实现三维测量的其他场景。特别是与火车车轮制造采用类似辗轧工艺成型的工件在线测量,算法可以实时获得成型的二维断面轮廓和三维整体数据。在后续的研究中,将进一步地探索改进型 KD-Tree 空间结构,并应用到更多的线激光扫描测量场景中。

### 参考文献

- [ 1 ] JIANG X, ZHANG X L, LU W D, et al. Research on gear profile deviation detection method based on linear

- structured light[J]. The International Journal of Advanced Manufacturing Technology, 2024, 131(9/10): 5353-5363.
- [2] JIN L, LI S Y, QIN G, et al. Outer surface defect detection of steel pipes with 3D vision based on multi-line structured lights[J]. Measurement Science and Technology, 2024, 35(6): 065203.
- [3] 黄喆,王浩森,胡成军,等.基于双线激光基准的盾尾间隙测量方法研究[J].仪器仪表学报,2023,44(7):85-92.
- HUANG ZH, WANG H S, HU CH J, et al. Research on the measurement method of shield tail clearance based on double line laser datum[J]. Chinese Journal of Scientific Instrument, 2023, 44(7): 85-92.
- [4] YANG G W, WANG Y ZH. Three-dimensional measurement of precise shaft parts based on line structured light and deep learning [J]. Measurement, 2022, 191: 110837.
- [5] YAN SH Z, XU D, HE W ZH, et al. Online measurement system of slab front-end bending in hot rough rolling based on line structured light vision [J]. Measurement Science and Technology, 2023, 34(9): 095201.
- [6] 刘少丽,杜浩浩,何楠,等.基于线结构光的大型构件表面尺寸加工质量检测方法[J].北京理工大学学报自然版,2024,44(7):701-711.
- LIU SH L, DU H H, HE N, et al. Surface dimension processing quality detection method for large components based on line structured light[J]. Transactions of Beijing Institute of Technology, 2024, 44(7): 701-711.
- [7] 石照耀,李美川,孙衍强,等.齿轮线激光三维测量仪的研制[J].仪器仪表学报,2024,45(4):95-103.
- SHI ZH Y, LI M CH, SUN Y Q, et al. Development of three-dimensional gear measuring instrument with line laser [J]. Chinese Journal of Scientific Instrument, 2024, 45(4): 95-103.
- [8] YE J, XIA G S, LIU F, et al. 3D reconstruction of line-structured light based on binocular vision calibration rotary axis[J]. Applied Optics, 2020, 59(27): 8272-8278.
- [9] 李津容,葛广言,冯晓冰,等.在机测量线激光传感器安装位姿的全局标定[J].仪器仪表学报,2022,43(9):158-165.
- LI J R, GE G Y, FENG X B, et al. Global calibration of laser sensor installation pose of on-machine measurement line[J]. Chinese Journal of Scientific Instrument, 2022, 43(9): 158-165.
- [10] 董洁,王宗平,欧登荧,等.基于线激光扫描的叶片三维型面重构方法[J].四川大学学报(自然科学版),2023,60(3):110-117.
- DONG J, WANG Z P, OU D Y, et al. Three-dimensional surface reconstruction method of blade based on line laser scanning[J]. Journal of Sichuan University (Natural Science Edition), 2023, 60(3): 110-117.
- [11] LANG H, PENG Y, ZOU ZH, et al. Multi-feature-filtering-based road curb extraction from unordered point clouds[J]. Sensors, 2024, 24(20): 6544.
- [12] ZHANG Y, SUN Y, WU L, et al. 3D point cloud semantic segmentation based PACConv and SE\_variant[J]. Instrumentation, 2023, 10(4): 27-38.
- [13] 马伯瑞,杨明来,曹振丰.基于局部平均的钢轨轮廓点云精简方法[J].电子测量与仪器学报,2025,39(2):205-212.
- MA B R, YANG M L, CAO ZH F. The rail profile point cloud simplification method based on local averaging[J]. Journal of Electronic Measurement and Instrumentation, 2025, 39(2): 205-212.
- [14] 王曦,王宗彦,张宇廷,等.基于FPFH的权重局部最优投影点云精简算法[J].电子测量技术,2022(23):119-124.
- WANG X, WANG Z Y, ZHANG Y T, et al. Weighted local optimal projection point cloud simplification algorithm based on FPFH[J]. Electronic Measurement Technology, 2022(23): 119-124.
- [15] 张立斌,吴岛,单洪颖,等.基于激光点云的车辆外廓尺寸动态测量方法[J].华南理工大学学报(自然科学版),2019,47(3):61-69.
- ZHANG L B, WU D, SHAN H Y, et al. Dynamic measurement method for vehicle contour dimensions based on laser point cloud[J]. Journal of South China University of Technology (Natural Science Edition), 2019, 47(3): 61-69.
- [16] FUANGKHON P. Boosting interclass boundary preservation (BIBP): A KD-tree enhanced data reduction algorithm[J]. International Journal of Information Technology, 2024, 16(7): 4069-4088.
- [17] CHATTERJEE B, WALULYA I, TSIGAS P. Concurrent linearizable nearest neighbour search in LockFree-KD-tree[J]. Theoretical Computer Science, 2021, 886: 27-48.
- [18] 樊爽,郭玉荣.基于逐点移动最小二乘拟合的数字图像相关测量[J].激光与光电子学进展,2023,60(6):240-250.
- FAN SH, GUO Y R. Digital image correlation measurement based on point-by-point moving least squares fitting[J]. Progress in Laser and Optoelectronics, 2023, 60(6): 240-250.

- [19] 刘书桂,毛晨丽,张海涛. 基于无序点云的叶片截面特征参数提取[J]. 航空动力学报,2016,31(9):2226-2232.  
LIU SH G, MAO CH L, ZHANG H T. Extraction of characteristic parameters of blade cross-section based on disordered point cloud [J]. Journal of Aerodynamics, 2016, 31(9): 2226-2232.
- [20] SK HOJAYFA R, MANAS KUMAR M. A hybridization of grey wolf optimizer and genetic algorithm for the traveling salesman problems[J]. Soft Computing, 2024, 28(23/24): 13127-13148.
- [21] 赵京东,杨凤华. 激光散乱点云K最近邻搜索算法[J]. 计算机应用,2016,36(10):2863-2869.  
ZHAO J D, YANG F H. K-nearest neighbor search algorithm for laser scattered point cloud [J]. Journal of Computer Applications, 2016, 36(10): 2863-2869.
- [22] FUJITA Y, IKUNO S, ITOH T, et al. Modified improved interpolating moving least squares method for meshless approaches[J]. IEEE Transactions on Magnetics, 2019, 55(6): 2900374.
- [23] DAGUM L, MENON R. OpenMP: An industry standard API for shared-memory programming[J]. IEEE Computational Science and Engineering, 1998, 5(1): 46-55.

### 作者简介



李欣飞,2023年于重庆理工大学获得学士学位,现为重庆理工大学硕士研究生,主要研究方向为数字化与智能制造。

E-mail:1780295190@qq.com

**Li Xinfei** received his B. Sc. degree from Chongqing University of Technology in 2023.

He is currently a master's student at Chongqing University of Technology, with a research focus on digitalization and intelligent manufacturing.



鄢然(通信作者),2010年于华中科技大学获得学士学位,2016年于华中科技大学获得博士学位。现为重庆理工大学副教授,主要研究方向为深度学习与计算机视觉、智能制造与工业机器人等。

E-mail:yanran@cqu.edu.cn

**Yan Ran** (Corresponding author) received his B. Sc. degree from Huazhong University of Science and Technology in 2010 and his Ph. D. degree from Huazhong University of Science and Technology in 2016. He is currently an associate professor at Chongqing University of Technology, with research interests in deep learning and computer vision, intelligent manufacturing and industrial robots.