

# 基于散乱点云的列车关键部件高效滤波平滑算法

曾妮, 李金龙\*, 高晓蓉, 张渝, 罗林

西南交通大学物理科学与技术学院, 四川 成都 610031

**摘要** 针对三维激光扫描设备实际采集的列车关键部件点云模型通常存在噪声点和采集误差等问题, 提出一种高效的点云小尺度噪声滤波平滑算法。首先利用K-D树构建点与点之间的几何关系, 设置中心点并利用K近邻算法查询邻域信息; 然后假设输入点云与滤波后输出的点云之间存在线性关系, 从而利用两片点云构造代价函数; 最后通过求解代价函数最小值确定线性参数, 从而得到平滑后的点云模型。代价函数构建过程中, 根据点与其K近邻内点的欧氏距离的方差与整体欧氏距离方差的比值大小自适应地调节权重, 达到动态调整线性模型参数的目的。实验结果表明, 所提算法能够快速修正小尺度噪声、平滑边界轮廓、提高质量, 为点云识别、重建等后期任务奠定基础。

**关键词** 图像处理; 点云; 去噪; 列车关键部件; 自适应加权引导滤波

中图分类号 TP391

文献标志码 A

DOI: 10.3788/LOP221987

## Efficient Filtering and Smoothing Algorithm For Train Key Components Based on Scattered Point Clouds

Zeng Ni, Li Jinlong\*, Gao Xiaorong, Zhang Yu, Luo Lin

School of Physical Science and Technology, Southwest Jiaotong University, Chengdu 610031, Sichuan, China

**Abstract** An efficient point cloud small-scale noise filtering and smoothing algorithm is proposed to address the issues of noise points and acquisition errors in the point cloud model of the key components of the train acquired by 3D laser scanning equipment. First, a K-D tree was used to construct the geometric relationship between points, set the center point, and query the neighborhood information using the K nearest neighbor algorithm. Then, it was assumed that there was a linear relationship between the input point cloud and the filtered output point cloud. Finally, the linear parameters were determined by solving the cost function's minimum value, and the smoothed point cloud model was obtained. To adjust the linear model's parameters dynamically, the weight was adaptively adjusted based on the ratio of the Euclidean distance variance of the point and its K nearest neighbor to the overall Euclidean distance variance during the construction of the cost function. The experimental results reveal that the proposed algorithm can quickly correct small-scale noise, smooth the boundary contour, improve the quality, and thereby lay the groundwork for future tasks such as point cloud recognition and reconstruction.

**Key words** image processing; point cloud; denoise; train key component; adaptive weighted guided filtering

## 1 引言

轨道交通的快速发展对列车行驶安全性提出更高挑战<sup>[1-2]</sup>, 列车关键部件检测则是保障行车安全的重要环节<sup>[3]</sup>。目前常用的列车关键部件检测方法主要是人工检测法, 这种方法易漏检、误检, 并且效率低、不能数字化管理<sup>[4]</sup>。利用二维图像进行检测也存在着深度信息缺乏、过度曝光图像无法使用等缺点<sup>[5-6]</sup>。与之相

比, 三维点云有着较大优势, 利用信息量更丰富的点云进行安全性检测具有较高的实用价值<sup>[7-8]</sup>。然而, 实际采集的列车关键部件原始点云普遍存在噪声点和边界模糊等问题, 这会严重影响点云识别、重建任务的精度, 因此需要对其进行去噪平滑处理<sup>[9]</sup>。噪声点云可以归纳为两类<sup>[10]</sup>: 1) 大尺度噪声, 即离主体点云较远的点, 数量较少, 通常采用半径滤波、统计滤波等方式即可去除; 2) 小尺度噪声, 即靠近主体点云的点, 通常是

收稿日期: 2022-07-04; 修回日期: 2022-08-05; 录用日期: 2022-09-05; 网络首发日期: 2022-09-15

基金项目: 国家自然科学基金国际合作项目(61960206010)

通信作者: jinlong\_lee@126.com

由设备本身引起的测量误差,数量较多,这些噪声点与主体点不易辨别,但可以通过矫正点的位置达到平滑边界的目的。

近年来,提出大量平滑算法用于小尺度噪声处理。基于邻域的滤波算法——双边滤波<sup>[11]</sup>,最早在二维图像领域取得优异的降噪和保边表现。文献[12]将其推广至三维点云,提出利用位置信息和法线信息的点云双边滤波算法(BF),基本思想是将主体点云附近的噪声点沿着法向量方向移动从而调整位置,该算法有较大的计算开销并存在梯度反转问题。文献[13]提出基于主成分分析的双边滤波算法,利用当前点和邻域点的灰度定义加权函数,但要求点云含有颜色亮度信息,不适用于实际采集的列车点云。基于双边滤波的点云去噪算法过程简单,但对尖锐边缘处噪声矫正不准确<sup>[14]</sup>。基于投影的点云滤波算法的经典之作是移动最小二乘(MLS)算法和局部最优投影(LOP)算法。文献[15]提出的MLS被文献[16]用于处理点云噪声问题,MLS迭代地将噪声点投影到估计的平面上以重建光滑表面,但计算量大,不能很好处理离群点。文献[17]采用基于LOP的操作,但在点云分布不均匀时局部最优的投影倾向无法收敛。MLS和LOP也不太适用于处理实际采集的列车关键部件点云模型。文献[18]在引导图像滤波<sup>[19]</sup>的基础上提出点云引导滤波(GF),与点云双边滤波相比效率更高、保边效果更好,不会发生梯度反转现象。但点云引导滤波针对整片点云采用相同的平滑参数,未能充分考虑点云细节,导致细节突出、边缘明显的区域出现模糊,平坦区域又被过度矫正。

基于此,本文通过分析铁路现场实际采集的列车关键部件点云的特点,提出一种点云自适应加权引导滤波(AWGF)算法。与传统点云引导滤波算法不同,所提算法认为点云中曲率变化大的区域比平坦区域包含更重要的信息,同时距离主体点云稍远的噪声点应给予更大的矫正权重。因此可以通过目标点与邻域点欧氏距离方差的平均值大小,给定邻域点云不同的权重因子自适应地调节滤波平滑参数,改变代价函数,从而调整输出点云模型。实验结果表明,所提算法相比经典的点云双边滤波和点云引导滤波具有较好的表现力,能够高效、准确地完成列车部件点云平滑保边任务。

## 2 基本原理

### 2.1 点云自适应加权引导滤波流程

点云自适应加权引导滤波算法整体流程主要包括点云数据输入及读取、拓扑关系构建、边缘感知权重计算、线性模型参数计算、点云线性变换等过程,具体流程如图1所示。

### 2.2 拓扑关系构建及邻域点集查找

实际采集的列车关键部件点云是无序点云,分布不均匀,点与点之间不存在拓扑信息,因此要对其进行拓扑关系构建。本实验采用K-dimensional tree(K-D

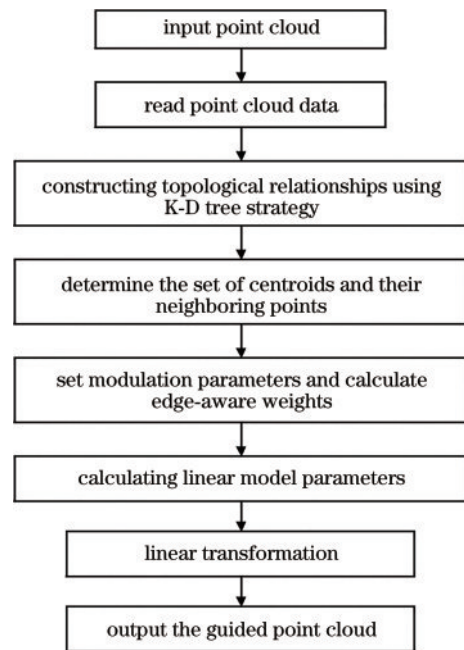


图1 点云自适应加权引导滤波

Fig. 1 Point cloud adaptive weighted guided filtering

tree)构建点云拓扑关系<sup>[20]</sup>,主要步骤如下:首先根据输入的点云数据建立根节点并选择方差值最大的特征为算法的分割特征;再计算分割特征下对应的中位数,将其作为分割点,遍历所有数据;将特征小于中位数对应的点划分到根节点下面的左子节点下,将特征大于中位数对应的点划分到根节点下面的右子节点下,不断迭代划分,直到所有数据都被建立于节点上面,如图2所示。

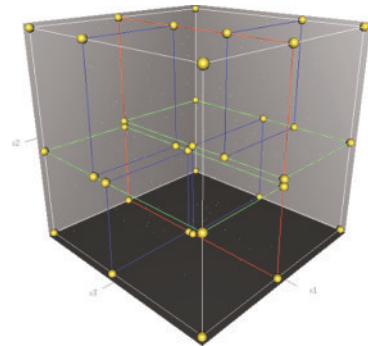


图2 K-D树空间划分图

Fig. 2 Diagram of K-D tree space division

拓扑关系构建完成后需要进行邻域搜索。本实验采用K近邻(KNN)搜索方法,即寻找距离查询点最近的 $k$ 个点构成邻域信息。KNN算法的具体步骤如下:对于点云 $Q = \{p_1, p_2, \dots, p_n\}$ ,首先遍历每一个采样点 $p_i$ ,计算出 $Q$ 中其他点 $p_{ij}$ 到 $p_i$ 的欧几里得距离 $d_{ij}$ ;再根据 $d_{ij}$ 从小到大排序,最后选择与 $p_i$ 距离最近的 $k$ 个点作为该采样点的邻域点集 $N(p_i)$ 。图3为KNN算法示意图。

### 2.3 点云自适应加权引导滤波

点云引导滤波的核心思想为滤波后输出的点云和对应的输入点云之间具有一个基于邻域的线性关系,

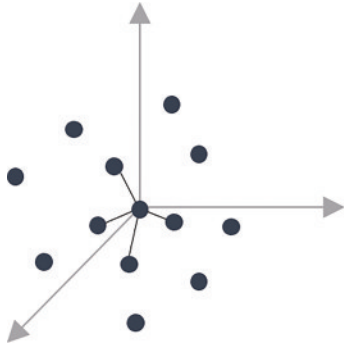


图 3 KNN算法示意图

Fig. 3 Schematic of KNN algorithm

即每一个点  $\mathbf{p}_i$  的线性变换参数和其邻域点集  $N(\mathbf{p}_i)$  相关<sup>[20]</sup>。所提点云自适应加权引导滤波则在点云引导滤波基础上考虑局部细节特征,给予点云局部信息不同的变换权重。首先采用 K-D tree 构建点云拓扑关系,确定每个采样点  $\mathbf{p}_i$  邻域搜索范围的参数  $k$  并查询邻域  $N(\mathbf{p}_i)$ ,用  $\mathbf{p}_{ij}$  表示  $\mathbf{p}_i$  在  $N(\mathbf{p}_i)$  中的第  $j$  个邻域点。在  $N(\mathbf{p}_i)$  中,每一个点都会经历一次线性变换。点云自适应加权引导滤波的线性变换模型可描述为

$$\mathbf{q}_{ij} = \mathbf{a}_i \mathbf{p}_{ij} + \mathbf{b}_i, \quad (1)$$

式中:  $\mathbf{q}_{ij}$  为滤波降噪平滑后的点云;  $\mathbf{a}_i$  和  $\mathbf{b}_i$  为受到预设邻域限制的线性模型参数,通过最小化代价函数得到。代价函数定义如下:

$$J(\mathbf{a}, \mathbf{b}) = \sum_{\mathbf{p}_j \in N(\mathbf{p}_i)} [(\mathbf{a}_i \mathbf{p}_{ij} + \mathbf{b}_i - \mathbf{p}_{ij})^2 + \epsilon \mathbf{a}_i^2], \quad (2)$$

式中:  $\epsilon$  为调控参数,主要用来控制算法的平滑程度。代价函数反映滤波输出的点云数据与滤波输入的引导三维点云模型之间的差异。通过对式(2)中  $\mathbf{a}_i$  和  $\mathbf{b}_i$  求偏导,可以求得代价函数的极值对应定义域的数值:

$$\frac{\partial J(\mathbf{a}, \mathbf{b})}{\partial \mathbf{a}_i} = \sum_{\mathbf{p}_j \in N(\mathbf{p}_i)} 2[(\mathbf{a}_i \mathbf{p}_{ij} + \mathbf{b}_i - \mathbf{p}_{ij}) \mathbf{p}_{ij} + \epsilon \mathbf{a}_i], \quad (3)$$

$$\frac{\partial J(\mathbf{a}, \mathbf{b})}{\partial \mathbf{b}_i} = \sum_{\mathbf{p}_j \in N(\mathbf{p}_i)} [2(\mathbf{a}_i \mathbf{p}_{ij} + \mathbf{b}_i - \mathbf{p}_{ij})]. \quad (4)$$

当代价函数为极小值时,两个偏导数都为零,可计算出线性模型的参数  $\mathbf{a}_i$  和  $\mathbf{b}_i$ ,参数解的表达式为

$$\mathbf{a}_i = \frac{\left( \frac{1}{|N(\mathbf{p}_i)|} \sum_{\mathbf{p}_j \in N(\mathbf{p}_i)} \mathbf{p}_{ij} \mathbf{p}_{ij} - \bar{\mathbf{p}} \bar{\mathbf{p}} \right)}{\left[ \left( \frac{1}{|N(\mathbf{p}_i)|} \sum_{\mathbf{p}_j \in N(\mathbf{p}_i)} \mathbf{p}_{ij} \mathbf{p}_{ij} - \bar{\mathbf{p}} \bar{\mathbf{p}} \right) + \epsilon \right]}, \quad (5)$$

$$\mathbf{b}_i = \bar{\mathbf{p}} - \mathbf{a}_i \bar{\mathbf{p}}, \quad (6)$$

$$\bar{\mathbf{p}} = \frac{1}{|N(\mathbf{p}_i)|} \sum_{\mathbf{p}_j \in N(\mathbf{p}_i)} \mathbf{p}_{ij}. \quad (7)$$

采用输入点云作为引导点云,这样不仅简单有效,而且还具有可以让降噪处理后三维点云数据更好地保持原始点云数据几何特征的优势。为了更好地处理边缘,设定边缘感知权重  $\omega_{(\mathbf{p}'_j)}$ :

$$\omega_{(\mathbf{p}'_j)} = \frac{1}{|N(\mathbf{p}_i)|} \sum_{\mathbf{p}_j \in N(\mathbf{p}_i)} \frac{\sigma_i^2(\mathbf{p}'_{ij}) + \alpha}{\sigma_i^2(\mathbf{p}_{ij}) + \alpha}, \quad (8)$$

式中:  $\sigma_i^2(\mathbf{p}'_{ij})$  为  $\mathbf{p}_i$  在  $N(\mathbf{p}_i)$  中与周围  $j$  个邻域点距离的方差;  $\sigma_i^2(\mathbf{p}_{ij})$  为  $N(\mathbf{p}_i)$  中所有点与其邻域点距离的方差;  $\alpha$  为常数。在实验过程中,选定点  $\mathbf{p}_i$  并采用 K 近邻算法确定邻域,邻域确定后将  $\mathbf{p}_i$  与邻域点进行距离方差计算。当  $\mathbf{p}_i$  与邻域点距离方差越大时,边缘感知权重  $\omega_{(\mathbf{p}'_j)}$  越大,而代价函数  $J(\mathbf{a}, \mathbf{b})$  则相应减小,参数解  $\mathbf{a}_i$ 、 $\mathbf{b}_i$  随之调节。使用边缘感知权重向边缘处的点分配比在平坦区域点更大的权重,相应地在边缘处正则化系数较小,可以很好保留边缘,调节距离稍远噪声点的位置。因此,代价函数改写为

$$J(\mathbf{a}, \mathbf{b}) = \sum_{\mathbf{p}_j \in N(\mathbf{p}_i)} [(\mathbf{a}_i \mathbf{p}_{ij} + \mathbf{b}_i - \mathbf{p}_{ij})^2 + \frac{\epsilon}{\omega_{(\mathbf{p}'_j)}} \mathbf{a}_i^2], \quad (9)$$

参数解相应变换为

$$\mathbf{a}_i = \frac{\left( \frac{1}{|N(\mathbf{p}_i)|} \sum_{\mathbf{p}_j \in N(\mathbf{p}_i)} \mathbf{p}_{ij} \mathbf{p}_{ij} - \bar{\mathbf{p}} \bar{\mathbf{p}} \right)}{\left[ \left( \frac{1}{|N(\mathbf{p}_i)|} \sum_{\mathbf{p}_j \in N(\mathbf{p}_i)} \mathbf{p}_{ij} \mathbf{p}_{ij} - \bar{\mathbf{p}} \bar{\mathbf{p}} \right) + \frac{\epsilon}{\omega_{(\mathbf{p}'_j)}} \right]}, \quad (10)$$

$$\mathbf{b}_i = \bar{\mathbf{p}} - \mathbf{a}_i \bar{\mathbf{p}}. \quad (11)$$

## 2.4 评价指标

为了客观评价点云滤波算法的平滑效果,采用定量匹配度指标<sup>[21]</sup>,分别为点云最大偏移距离 ( $D_{\max}$ )、点云平均偏移距离 ( $D_{\text{mean}}$ )、点云平均偏移量标准差 ( $D_{\text{std}}$ )。其中:  $D_{\max}$  指经过滤波矫正后的点云到标准点云模型中对应的点之间距离的最大值;  $D_{\text{mean}}$  指经过滤波矫正后的点云中的点到标准点云模型中对应的点之间距离的平均值;  $D_{\text{std}}$  指经过滤波矫正后的点云到标准点云模型中对应的点之间距离平均值的标准差。  $D_{\max}$ 、 $D_{\text{mean}}$ 、 $D_{\text{std}}$  的计算公式分别为

$$D_{\max} = \max(\|\mathbf{q}_i - \mathbf{p}_i\|), \quad (12)$$

$$D_{\text{mean}} = \frac{1}{n} \sum_{i=1}^n (\|\mathbf{q}_i - \mathbf{p}_i\|), \quad (13)$$

$$D_{\text{std}} = \frac{1}{n} \sum_{i=1}^n (\|\mathbf{q}_i - \mathbf{p}_i\| - D_{\text{mean}})^2. \quad (14)$$

3 个值越小,说明位置矫正后的模型与原始模型的匹配程度越高,对应的矫正效果越好。同时采用滤波时间来评价算法计算效率,滤波时间越短,计算复杂度越低,更为高效,实用性更强。

## 3 实验及结果分析

### 3.1 实验配置

实验环境如下:计算机处理器为 AMD Ryzen 7 4800H with Radeon Graphics,软件环境为 64 位 Windows 10 系统,编程平台为 Microsoft Visual Studio 2019 结合 PCL 点云库,编程语言为 C++。点云模型为标准列车

轮对点云、标准列车转向架点云以及四川主导科技有限公司提供的实际采集的列车关键部件点云模型。

### 3.2 定量评价实验

为了定量评价算法计算效率和滤波性能,采用标准点云模型进行参数对比实验。但现场采集的列车关键部件点云由于测量误差的存在会使位于同一面上的点出现偏移,因此本实验充分利用高斯噪声扰动点坐标而不增加点的特性。主要操作方法为对标准点云模型施加高斯噪声,使点云模型中的点发生坐标偏移来模拟现场采集的点云。首先,对轮对点云和转向架点云施加

均值为 1、方差为 20 的高斯噪声。轮对点云点数为 50000,尺寸如下: $X=2247.36$  mm, $Y=999.147$  mm, $Z=999.604$  mm。施加噪声后,噪声点云与标准点云的最大偏移距离为 92.6869 mm,平均偏移距离为 16.7004 mm,平均偏移距离标准差为 10.501 mm。转向架点云点数为 500000,尺寸如下: $X=3555.55$  mm, $Y=2981.85$  mm, $Z=1166.34$  mm。施加噪声后,噪声点云与标准点云的最大偏移距离为 91.1029 mm,平均偏移距离为 14.7018 mm,平均偏移距离标准差为 10.0926 mm。实验效果如图 4 和图 5 所示。

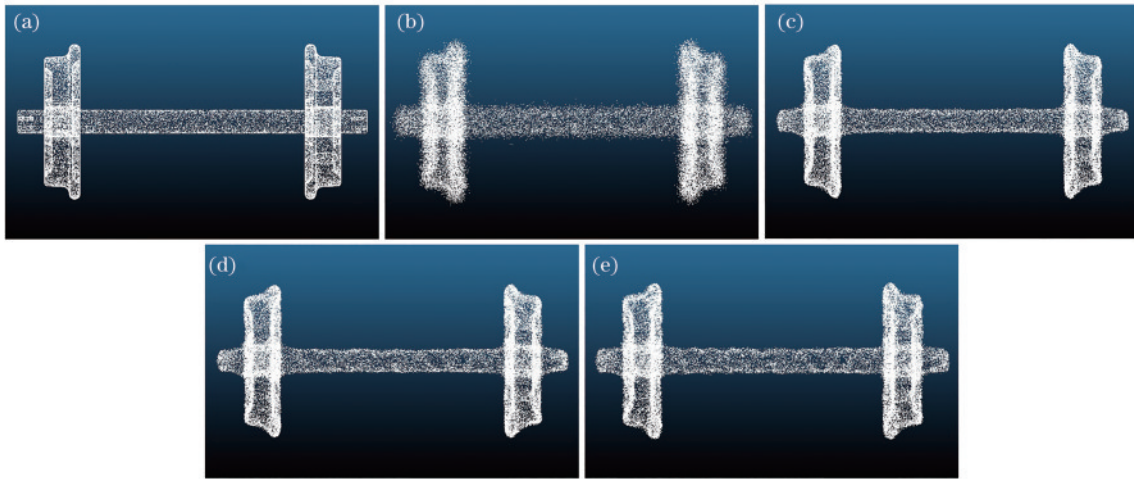


图 4 列车轮对点云模型滤波前后对比图。(a)标准点云;(b)施加高斯噪声后;(c)双边滤波;(d)引导滤波;(e)自适应加权引导滤波  
Fig. 4 Comparison of train wheel pair point clouds before and after filtering. (a) Standard point cloud; (b) after applying Gaussian noise; (c) bilateral filtering; (d) guided filtering; (e) adaptive weighted guided filtering

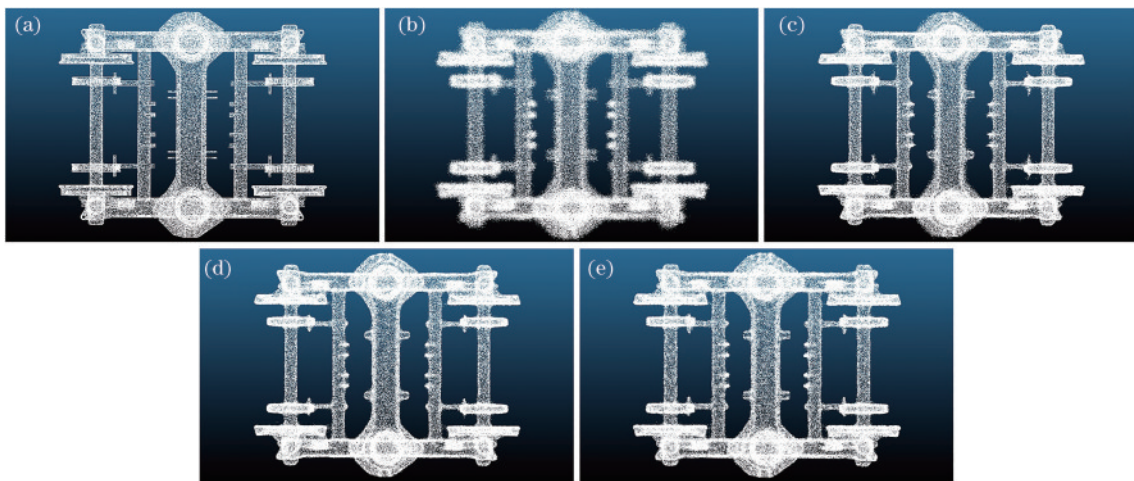


图 5 列车转向架点云模型滤波前后对比图。(a)标准点云;(b)施加高斯噪声后;(c)双边滤波;(d)引导滤波;(e)自适应加权引导滤波

Fig. 5 Comparison of train bogie point clouds before and after filtering. (a) Standard point cloud; (b) after applying Gaussian noise; (c) bilateral filtering; (d) guided filtering; (e) adaptive weighted guided filtering

标准列车轮对点云部件边界清晰光滑、分布均匀,如图 4(a)所示。对其施加高斯噪声后,点的位置发生移动,导致边界上本处于同一平面的点出现不同程度的偏移,如图 4(b)所示。图 5(a)为列车转向架点云标准模型,边界同样清晰光滑,施加噪声后,边界变得模

糊,如图 5(b)所示。这种情况和实际采集的铁路部件点云由于测量误差导致边界不清晰的现象相似。因此,对施加噪声后的标准列车部件点云模型进行去噪平滑处理,以定量对比所提算法和其他算法的性能。图 4(c)~(e)分别为对含高斯噪声的列车轮对点云采

用双边滤波、点云引导滤波、点云自适应加权引导滤波进行滤波去噪的结果,图 5(c)~(e)分别为对含高斯噪声的列车转向架点云采用双边滤波、点云引导滤波、点云自适应加权引导滤波进行滤波去噪的结果。两个模型的去噪结果均表明:经过点云双边滤波处理后,点云边界不够清晰,仍有散乱点分布在主体点云周边;而点云引导滤波和自适应加权引导滤波后的点云边界更加清晰,分布在主体点云周边的散乱点更少;在点云边缘处,点云自适应加权引导滤波处理效果更好、更为光滑。

仅仅通过肉眼主观评价滤波效果具有一定的局限性,为更客观评价算法效果,将添加高斯噪声后的模型、点云双边滤波后的模型、点云引导滤波后的模型、点云自适应加权引导后的模型在相同的评价指标下,分别与标准列车轮对模型和标准列车转向架模型进行对比,结果如表 1、表 2 所示。

表 1 列车轮对点云模型滤波结果

Table 1 Train wheel pair point cloud filtering results

Train wheel pair	Time /ms	$D_{max}$	$D_{mean}$	$D_{std}$
Gaussian noise		92.69	16.70	10.50
BF	1419.20	59.28	10.75	6.78
GF	560.48	56.45	10.62	6.28
AWGF	715.10	56.79	10.39	6.14

表 2 列车转向架点云模型滤波结果

Table 2 Train bogie point cloud filtering results

Train bogie	Time /ms	$D_{max}$	$D_{mean}$	$D_{std}$
Gaussian noise		91.10	14.70	10.09
BF	35081.10	84.33	9.70	6.28
GF	15886.60	56.11	9.62	5.90
AWGF	16072.50	55.10	9.29	5.74

表 1 和表 2 数据表明,从计算复杂度而言,点云双边滤波耗时太长,并且滤波后的模型与标准模型差距

最大,不具备竞争力。点云引导滤波摒弃计算点云法向量,引入线性模型的方式极大地降低了计算复杂度,节约了大量的计算成本,并且取得了较好的滤波表现。点云自适应加权引导滤波由于权重感知系数的引入,带来了一定的计算开销,导致滤波耗时相比于简单的点云引导滤波略长,但也使得滤波效果得到一定的提升。与点云双边滤波相比,所提算法耗时不足其一半,但滤波效果却具有显著的优势。

综合对比滤波效果图并结合实验数据可得,点云自适应加权引导滤波对于噪声点的矫正效果最好,滤波后的点云模型相对于标准模型的最大偏移距离、平均偏移距离和平均偏移距离标准差均最小,在运算性能和滤波性能两个方面达到平衡。

### 3.3 列车关键部件点云滤波

为进一步比较所提算法与点云双边滤波算法、点云引导滤波算法的性能,在现场采集的 6 个列车关键部件点云模型上进行对比实验。列车关键部件点对应点数如表 3 所示。

表 3 列车部件点云数目

Table 3 Number of point clouds of train components

Train key component	Number of points
1	218452
2	332895
3	523009
4	843918
5	1029027
6	1834880

图 6~11 展示了 6 个实际列车部件滤波前以及分别用点云双边滤波、点云引导滤波和点云自适应加权引导滤波去噪后的对比图。

通过实际列车关键部件滤波前后对比图可以看

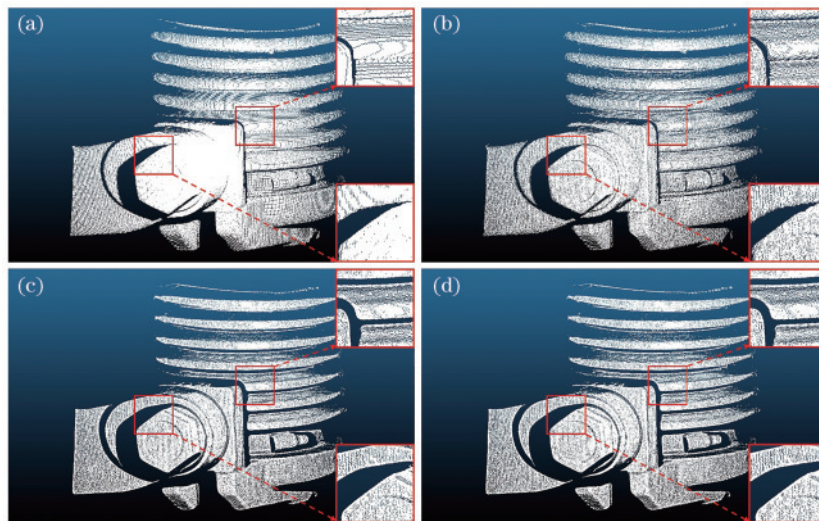


图 6 列车部件 1 对比图。(a)原始点云;(b)双边滤波;(c)引导滤波;(d)自适应加权引导滤波

Fig. 6 Comparison of train component 1. (a) Original point cloud; (b) bilateral filtering; (c) guided filtering; (d) adaptive weighted guided filtering

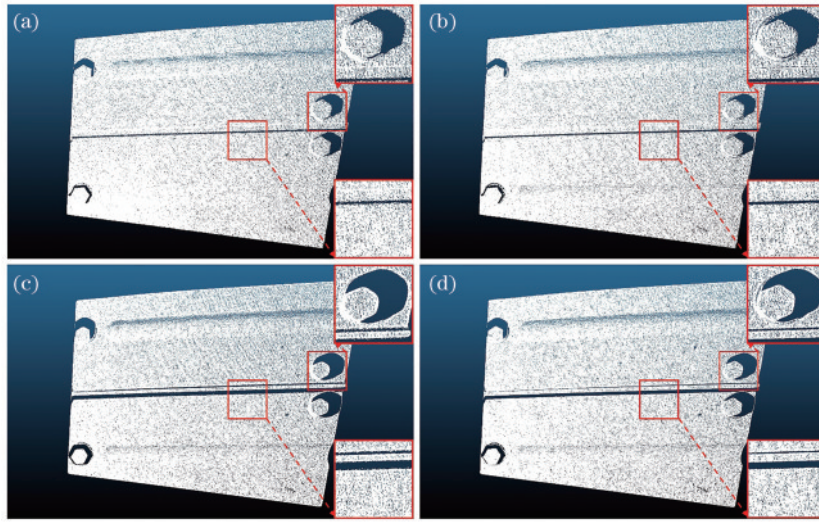


图 7 列车部件 2 对比图。(a)原始点云;(b)双边滤波;(c)引导滤波;(d)自适应加权引导滤波

Fig. 7 Comparison of train component 2. (a) Original point cloud; (b) bilateral filtering; (c) guided filtering; (d) adaptive weighted guided filtering

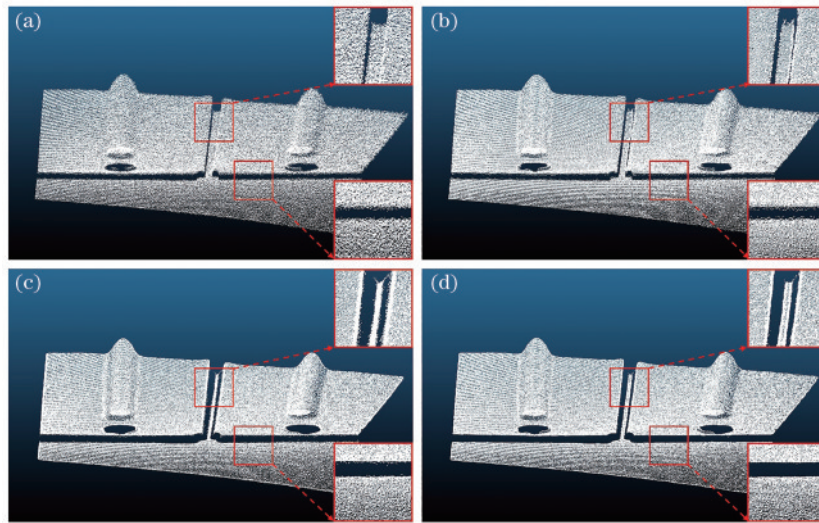


图 8 列车部件 3 对比图。(a)原始点云;(b)双边滤波;(c)引导滤波;(d)自适应加权引导滤波

Fig. 8 Comparison of train component 3. (a) Original point cloud; (b) bilateral filtering; (c) guided filtering; (d) adaptive weighted guided filtering

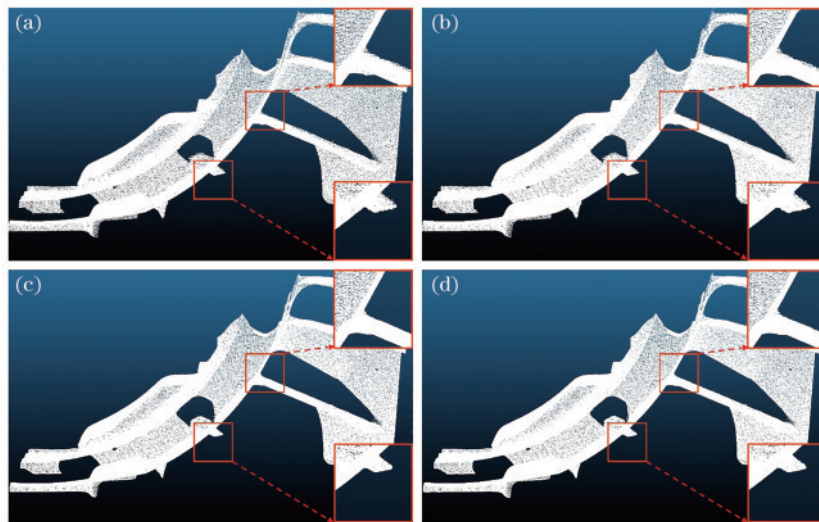


图 9 列车部件 4 对比图。(a)原始点云;(b)双边滤波;(c)引导滤波;(d)自适应加权引导滤波

Fig. 9 Comparison of train component 4. (a) Original point cloud; (b) bilateral filtering; (c) guided filtering; (d) adaptive weighted guided filtering

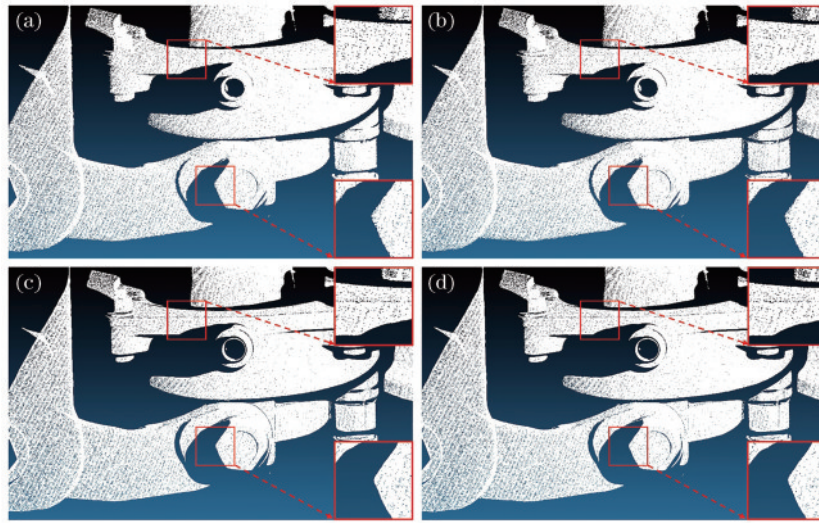


图 10 列车部件 5 对比图。(a)原始点云;(b)双边滤波;(c)引导滤波;(d)自适应加权引导滤波

Fig. 10 Comparison of train component 5. (a) Original point cloud; (b) bilateral filtering; (c) guided filtering; (d) adaptive weighted guided filtering

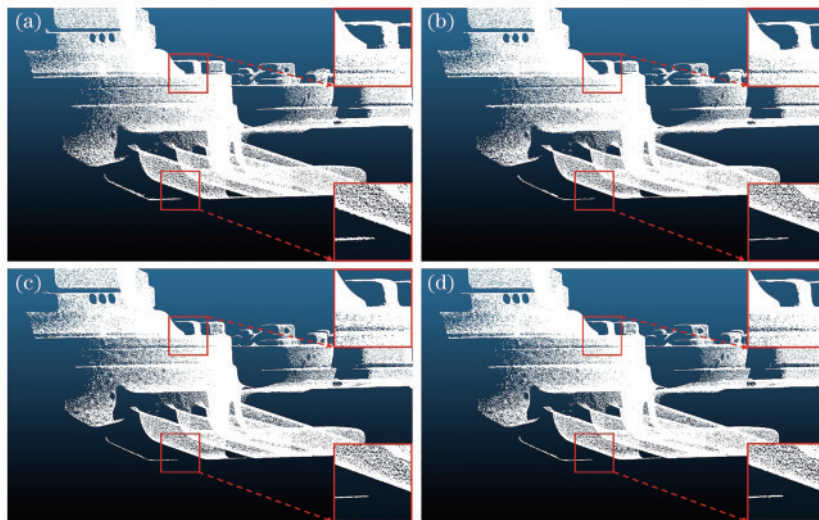


图 11 列车部件 6 对比图。(a)原始点云;(b)双边滤波;(c)引导滤波;(d)自适应加权引导滤波

Fig. 11 Comparison of train component 6. (a) Original point cloud; (b) bilateral filtering; (c) guided filtering; (d) adaptive weighted guided filtering

出,3种算法均能达到一定的去噪平滑效果。点云双边滤波对于边界轮廓处的平滑效果不及点云引导滤波和点云自适应加权引导滤波。在边缘处,点云自适应加权滤波的平滑保边效果比点云引导滤波更好。同时,滤波时间对比结果如表 4 所示,可以看出,

点云双边滤波在处理大规模点云时耗时较长,而点云自适应加权引导滤波与点云引导滤波算法时间相近。综上所述,所提算法能够取得计算效率和平滑效果的平衡,较好完成列车关键部件点云的去噪平滑任务。

表 4 不同数目点云部件滤波时间对比

Table 4 Comparison of filtering time for different number of point cloud components

unit: ms

Train component	1	2	3	4	5	6
BF	10086.2	15451.8	24106.4	32991.5	43347.6	78469.9
GF	4274.4	6579.5	10225.4	16610.5	19983.2	36216.9
AWGF	4376.1	6803.4	10667.7	17129.0	20345.7	37077.5

## 4 结 论

针对列车关键部件点云小尺度噪声去除和平滑任务,根据不同区域曲率大小设计边缘感知权重,对传统的点云引导滤波算法进行优化,提出点云自适应加权引导滤波算法,并在多个点云模型上进行对比实验。实验结果表明,点云自适应加权引导滤波算法相比于点云双边滤波具有极大的速度优势和性能优势。与点云引导滤波算法相比,所提算法滤波效果更好,与标准点云模型更为符合,能够为后续铁路列车关键部件检测任务提供基础,具有一定的实用价值。但该算法在自适应调节权重的同时带来了一定的计算开销,相比引导点云滤波速度上不占优势。后期可以尝试对原始点云先进行下采样,完成引导点云工作后再上采样填充点,达到整体滤波平滑同时节约计算时间的目的。

### 参 考 文 献

- [1] 丁叁叁, 陈大伟, 刘加利. 中国高速列车研发与展望[J]. 力学学报, 2021, 53(1): 35-50.  
Ding S S, Chen D W, Liu J L. Research, development and prospect of China high-speed train[J]. Chinese Journal of Theoretical and Applied Mechanics, 2021, 53(1): 35-50.
- [2] 李维刚, 梅洋, 樊响, 等. 基于车载激光点云的铁路轨道检测[J]. 中国激光, 2022, 49(4): 0410002.  
Li W G, Mei Y, Fan X, et al. Railway track detection based on vehicle laser point cloud[J]. Chinese Journal of Lasers, 2022, 49(4): 0410002.
- [3] Yao L Y, Qiu J H, Gao S B, et al. Defect detection in high-speed railway overhead contact system: importance, challenges, and methods[C]//2021 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC), June 18-21, 2021, Chengdu, China. New York: IEEE Press, 2021: 76-80.
- [4] 王志学, 张渝, 王楠, 等. 基于卷积神经网络的防松铁丝断裂分类算法研究[J]. 铁道学报, 2022, 44(2): 27-33.  
Wang Z X, Zhang Y, Wang N, et al. Locking wire fracture classification algorithm based on convolutional neural networks[J]. Journal of the China Railway Society, 2022, 44(2): 27-33.
- [5] 赵冰, 代明睿, 李平, 等. 基于深度学习的铁路关键部件缺陷检测研究[J]. 铁道学报, 2019, 41(8): 67-73.  
Zhao B, Dai M R, Li P, et al. Research on defect detection of railway key components based on deep learning[J]. Journal of the China Railway Society, 2019, 41(8): 67-73.
- [6] Li C Q, Wei Z Z, Xing J. Online inspection system for the automatic detection of bolt defects on a freight train [J]. Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit, 2016, 230(4): 1213-1226.
- [7] 李维刚, 樊响, 梅洋, 等. 基于车载激光点云的道路标线提取方法[J]. 中国激光, 2022, 49(7): 0710001.  
Li W G, Fan X, Mei Y, et al. Road marking extraction method based on vehicle laser point cloud[J]. Chinese Journal of Lasers, 2022, 49(7): 0710001.
- [8] Lin S, Xu C, Chen L P, et al. LiDAR point cloud recognition of overhead catenary system with deep learning[J]. Sensors, 2020, 20(8): 2212.
- [9] Han X F, Jin J S, Wang M J, et al. A review of algorithms for filtering the 3D point cloud[J]. Signal Processing: Image Communication, 2017, 57: 103-112.
- [10] 袁华, 庞建铿, 莫建文. 基于噪声分类的双边滤波点云去噪算法[J]. 计算机应用, 2015, 35(8): 2305-2310.  
Yuan H, Pang J K, Mo J W. Denoising algorithm for bilateral filtered point cloud based on noise classification [J]. Journal of Computer Applications, 2015, 35(8): 2305-2310.
- [11] Tomasi C, Manduchi R. Bilateral filtering for gray and color images[C]//Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271), January 7, 1998, Bombay, India. New York: IEEE Press, 1998: 839-846.
- [12] Julie D, Carlo D F. The bilateral filter for point clouds [J]. Image Processing on Line, 2017, 7: 278-287.
- [13] Zhang F, Zhang C, Yang H M, et al. Point cloud denoising with principal component analysis and a novel bilateral filter[J]. Traitement Du Signal, 2019, 36(5): 393-398.
- [14] Zhou L, Sun G X, Li Y, et al. Point cloud denoising review: from classical to deep learning-based approaches [J]. Graphical Models, 2022, 121: 101140.
- [15] Levin D. The approximation power of moving least-squares[J]. Mathematics of Computation, 1998, 67(224): 1517-1531.
- [16] Alexa M, Behr J, Cohen-Or D, et al. Computing and rendering point set surfaces[J]. IEEE Transactions on Visualization and Computer Graphics, 2003, 9(1): 3-15.
- [17] Lipman Y, Cohen-Or D, Levin D, et al. Parameterization-free projection for geometry reconstruction[J]. ACM Transactions on Graphics, 2007, 26(3): 22.
- [18] Han X F, Jin J S, Wang M J, et al. Guided 3D point cloud filtering[J]. Multimedia Tools and Applications, 2018, 77(3): 17397-17411.
- [19] He K M, Sun J, Tang X O. Guided image filtering[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(6): 1397-1409.
- [20] 魏亚, 肖庸, 闫闯, 等. 基于点云预处理的路面三维重构数据优化[J]. 吉林大学学报(工学版), 2020, 50(3): 987-997.  
Wei Y, Xiao Y, Yan C, et al. Data optimization of pavement 3D reconstruction based on point preprocessing [J]. Journal of Jilin University (Engineering and Technology Edition), 2020, 50(3): 987-997.
- [21] Javaheri A, Brites C, Pereira F, et al. Subjective and objective quality evaluation of 3D point cloud denoising algorithms[C]//2017 IEEE International Conference on Multimedia & Expo Workshops, July 10-14, 2017, Hong Kong, China. New York: IEEE Press, 2017.