

# 逆向工程中点云孔洞轮廓线的提取

王春香, 周国勇\*, 刘流

内蒙古科技大学机械学院, 内蒙古 包头 014010

**摘要** 逆向工程中点云模型孔洞边界的检测是孔洞修补的前提, 完美的孔洞轮廓线有利于提高孔洞修补的质量。首先参照二维图像中边界的定义, 给出了三维点云模型中孔洞边界的定义; 之后通过分析待测点邻域协方差矩阵特征值之间的关系, 设计了一种边界点检测算子, 用于初步提取孔洞边界特征点; 然后采用改进的 Kruskal 最小生成树算法构建点云最小生成图, 并在最小生成图中完成孔洞边界点的排序与孔洞轮廓线的提取。实验结果表明: 所提边界检测算子步骤简单, 算法效率高; 基于计算机图论的孔洞轮廓线生成算法提取的孔洞轮廓线具有光滑、连续的优点, 并且在生成轮廓线时完成了孔洞聚类。

**关键词** 检测; 点云; 协方差矩阵; 逆向工程; 边界检测

**中图分类号** TP391 **文献标志码** A

**doi:** 10.3788/LOP57.242801

## Extracting Contour Lines of Point Cloud in Reverse Engineering

Wang Chunxiang, Zhou Guoyong\*, Liu Liu

*Department of Mechanical Engineering, Inner Mongolia University of Technology, Baotou,  
Inner Mongolia 014010, China*

**Abstract** The detection of the hole boundary of the point cloud model in reverse engineering is necessary for the hole repair, and a perfect hole contour line is conducive to improving the quality of the hole repair. Based on the definition of boundary in two-dimensional images, the definition of hole boundary in three-dimensional point cloud was given. By analyzing the relationship between the eigenvalues of the covariance matrix of neighboring points, a boundary point detection operator was proposed to initially extract the feature points of the hole boundary, and then, the improved Kruskal minimum spanning tree algorithm was used to construct the minimum generated map of the point cloud. Subsequently, the hole boundary points were sorted and outline of the hole was extracted in the minimum generated graph. The experimental results show that the proposed boundary detection operator has simple steps and high efficiency. The hole contour line extracted by the hole contour line generation algorithm based on computer graph theory has the advantages of being smooth and continuous, and hole aggregation is completed while generating the contour line.

**Key words** detection; point cloud; covariance matrix; reverse engineering; edge detection

**OCIS codes** 280.3400; 140.3430; 270.3430; 200.3050

## 1 引言

采用扫描仪采集零件表面的点云数据, 然后对采集到的点云数据进行加工处理, 再根据处理后的零件表面点云数据重建零件的三维模型, 进而根据三维模型实现零件的开发再设计, 是机械产品逆向

工程的主要思想。在点云数据采集过程中, 受零件自身几何形状、扫描速率不均匀以及环境等因素的影响, 测得的点云数据不可避免地存在噪声、欠采样等问题, 从而严重影响了三维模型表面重建的质量。在去除噪声或者修补欠采样区域时, 应当首先检测模型的边缘轮廓, 这样不仅能确定待修补区域的轮

收稿日期: 2020-04-20; 修回日期: 2020-05-22; 录用日期: 2020-06-09

基金项目: 内蒙古自治区自然科学基金(2017MS(LH)0530)、包头市科技计划项目(2019Z3004-6)

\* E-mail: alex554332@163.com

廓及位置,还能在去除噪声时最大程度地保留边缘特征点。因此,基于实测点云数据的轮廓线检测对于三维模型的重建具有重要意义。一般地,边缘线在二维图像中定义<sup>[1]</sup>为:1)灰度边缘,通常位于与平均灰度差异较大的位置;2)纹理边缘,它是包含相同纹理的相邻区域之间在不同比例下突然的“粗度”变化区域,或是相邻区域中定向纹理之间突然的“方向性”变化区域。

文献[2]将二维边界线定义为强度图像中的一维不连续区域。然而,强度或者光谱信息无法描述3D点云完整的几何属性。因此,人们参照二维图像中边界的定义,将三维点云的边界定义为三维空间的几何不连续性,具体如下:1)边界,通常它们的邻域点在某一区域会出现数据点不连续的现象;2)折边,即邻域点的法向量不连续变化的区域,通常位于模型各曲面的交线处。

吕源治等<sup>[3]</sup>根据三角网格模型中边界边有且仅有一个邻接三角形的特性,获取第一个孔洞边界的特征点,然后采用多源最短路径搜索 Floyd 算法在三角网格模型中提取到了孔洞多边形。该方法能准确识别、连接有孤立边的孔洞以及两个相邻孔洞的特殊孔洞结构,具有更强的通用性和鲁棒性,但该方法需要将测得的点云数据转化成三角网格的形式,转化过程耗时且存在一定程度的失真。因此,该方法不适合用于形状复杂且精度要求较高的机械类零件。苏云龙等<sup>[4]</sup>基于高斯映射聚类算法提出了一种点云模型折边特征提取算法,但该算法不能提取孔洞边界的特征点。刘庆等<sup>[5]</sup>先在待测点的局部邻域内拟合平面,然后采用在拟合平面内求矢量和的方法提取到了点云的边缘特征。该算法简单高效,但无法处理密度不均匀的点云模型。曾露露等<sup>[6]</sup>依据孔洞边界点在二维相位图中的跳变提取到了孔洞特征点,但该算法需要将点云数据在二维与三维空间相互转换,限制了算法的效率和适用性。文献[7]将KNN(k-nearest neighbor)邻域投影到待测点的局部微切拟合平面,然后建立局部坐标系,在局部坐标系中求待测点与其邻域点构成的相邻向量之间的最大夹角,通过判断最大夹角是否大于预设的阈值来提取边界特征点,同时通过多尺度分析方法确定最佳角度阈值,取得了不错的边界提取效果。文献[8]通过分析协方差矩阵特征根之间的联系定义了三个形状因子,用以提取相应的特征,该方法的复杂度低,检测效果较好。Lin等<sup>[9]</sup>通过邻域点加权的方法,将采用协方差矩阵提取点云特征的方法拓展到

了采样密度不均匀的点云模型上,并基于机载激光雷达点云验证了该方法的效果。不同于传统的特征提取方法,文献[10]和[11]首先定义了边界特征场,然后用少量的数据集训练卷积神经网络,再用训练好的卷积神经网络提取点云模型中的边界特征点。该方法提高了边界特征场的计算效率,边界提取效果好,但该方法依赖于点云的采样密度,仅适用于均匀采样的点云模型。

本文首先分析了协方差矩阵的特性,然后根据特征值之间的关系定义了一种用于检测孔洞边界点的算子,并提出了一种新的孔洞轮廓线生成算法。

## 2 边界点检测算子

由于协方差矩阵能够简洁地表达图像某一区域内的统计特性,因此,被广泛应用于计算机视觉中。Tuzel等<sup>[12]</sup>首先将协方差矩阵作为描述子用于目标的检测。随后,协方差矩阵被应用于解决各种各样的计算机视觉问题。最近,基于协方差矩阵的三维描述子取得了不错的效果<sup>[13-14]</sup>。

假设点云中的待测点为 $p_i$ ,以 $p_i$ 为球心、半径为 $r$ 的球内的邻域点集为 $N_i$ ,则待测点 $p_i$ 的协方差矩阵 $C_{p_i}$ 的计算公式为

$$C_{p_i} = \sum_{p_k \in N_i} (p_k - \bar{p}_i)(p_k - \bar{p}_i)^T, \quad (1)$$

式中: $p_k$ 为待测点 $p_i$ 的邻域点; $p_k$ 为 $p_k$ 点的向量; $\bar{p}_i = \frac{1}{n} \sum_{p_k \in N_i} p_k$ , $n$ 为 $N_i$ 中邻域点的个数。分解协方差矩阵 $C_{p_i}$ 可求得其特征值 $\lambda_j$ 以及相应的特征向量 $v_j$ ( $j=1,2,3$ )。协方差矩阵是半正定矩阵,其特征值为非负实数,其大小表示邻域点在相应特征向量方向上相对于坐标均值点 $\bar{p}_i$ 的方差。

为检测孔洞边界点,需要分析邻域点集 $N_i$ 在待测点 $p_i$ 周边的分布情况。协方差矩阵 $C_{p_i}$ 的特征值与特征向量在以 $\bar{p}_i$ 为原点、特征向量为坐标轴的局部坐标系中定义了一个椭球,其表达式为

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1 \quad (2)$$

式中: $a=\lambda_1, b=\lambda_2, c=\lambda_3$ ,且 $\lambda_1 \geq \lambda_2 \geq \lambda_3$ 。通过分析椭球的形状可以判定待测点 $p_i$ 邻域的分布情况:当 $\lambda_1 \gg \lambda_2 \approx \lambda_3 \approx 0$ 时,椭球退化为沿特征向量 $v_1$ 方向的直线,表明邻域点分布在一条近似的直线上;当 $\lambda_1 \approx \lambda_2 \approx \lambda_3 \gg 0$ 时,椭球变成空间圆球,表明邻域点在空间内任意分布;当 $\lambda_3 \approx 0$ 时,椭球退化成平面椭圆,此时,不同特征值对应的邻域点分布情况如图1

所示。

由图1可以看出,待测点 $p_i$ 的邻域点分布有3种情况:1)当 $\lambda_1 \approx \lambda_2 \neq 0$ 时,待测点 $p_i$ 的邻域点均匀分布在四周,并且待测点 $p_i$ 为内部点,如图1(a)所示。2)当 $\lambda_1 > \lambda_2 \neq 0$ 时,邻域点集 $N_i$ 的某一区域存在数据缺失,此时待测点 $p_i$ 位于边界上,如图1(b)所示,因此 $\lambda_1 - \lambda_2$ 能反映点 $p_i$ 的位置情况。3)当 $\lambda_1 - \lambda_2 \approx 0$ 时,待测点 $p_i$ 为内部点;当 $\lambda_1 - \lambda_2$ 大于某一预先设定的阈值时,待测点 $p_i$ 为边界点。当邻域点集 $N_i$ 成直线分布时, $\lambda_1 - \lambda_2$ 达到最大值,

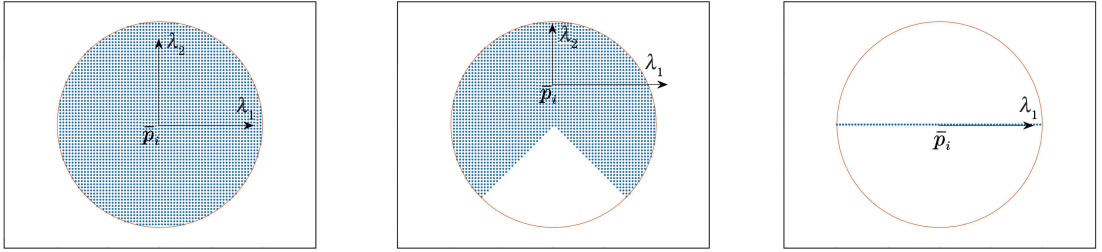


图1  $\lambda_3 \approx 0$ 时,不同特征值对应的邻域分布情况。(a)  $\lambda_1 \approx \lambda_2 \neq 0$ ; (b)  $\lambda_1 > \lambda_2 \neq 0$ ; (c)  $\lambda_1 > \lambda_2 \approx 0$

Fig. 1 Neighborhood distribution corresponding to different eigenvalues when  $\lambda_3 \approx 0$ . (a)  $\lambda_1 \approx \lambda_2 \neq 0$ ; (b)  $\lambda_1 > \lambda_2 \neq 0$ ; (c)  $\lambda_1 > \lambda_2 \approx 0$

### 3 孔洞轮廓的提取

#### 3.1 最小生成图的构建

由于散乱点云的无序性,基于点云的边界线检测算法在提取边界点后需要将边界点连接成边界线,以便于确定后续孔洞修补算法的插值区域。文献[15]采用了一种以距离为参考依据的孔洞边界点连接算法,即:在提取出来的边界点中任意取一点,搜索该点的最近点,并以这两点构成的线段作为基准,向两边生长。该算法适用于细长的孔洞边界,而本文算法提取到的孔洞轮廓往往是孔洞附近的一圈点带。采用这种算法生成的孔洞多边形为锯齿状,并有折返现象。图2(b)与图2(c)为图2(a)中圈选部分的放大,可以明显看到锯齿状多边形与折返现象。

基于上述原因,本文采用了一种基于改进最小生成树的孔洞边界点规则化算法。该算法首先通过边界检测算子计算点云中每一个点的 $c_b$ 值,然后在整体点云模型中生成最小生成图,通过搜索最小生成图完成孔洞轮廓点的提取及排序。定义两点 $p_i$ 和 $p_j$ 构成的边的权值 $w(i, j)$ 由两部分组成。第一部分为 $w_p(i, j)$ ,用以控制 $c_b$ 值的大小,其公式为

$$w_p(i, j) = 2 - c_{bi} - c_{bj}, \quad (4)$$

如图1(c)所示。将 $\lambda_1 - \lambda_2$ 的值归一化,并定义边界点检测算子 $c_b$ 为

$$c_b = \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2 + \lambda_3}. \quad (3)$$

当 $c_b > \delta$ 时,将点 $p_i$ 判定为边界点, $\delta$ 为预先设定的阈值参数。然而,采用该算子检测孔洞时效果并不理想,原因是该算子仅仅考虑了 $\lambda_1$ 与 $\lambda_2$ 之间的关系,从而将 $\lambda_3 \neq 0$ 的位于模型折边上的点误判为孔洞边界点,因此在限制 $c_b$ 取值的同时还需要保证 $\lambda_3 \approx 0$ 。

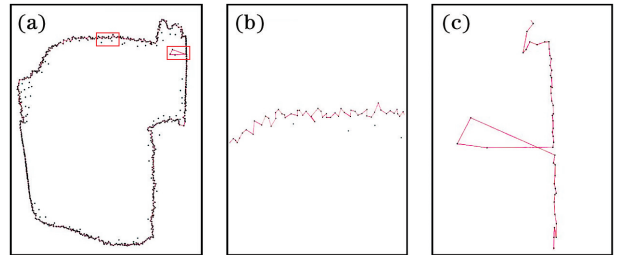


图2 仅以距离为参考的孔洞特征线连接。(a)孔洞多边形; (b)锯齿; (c)折返

Fig. 2 Hole feature line connection with reference to distance only. (a) Hole polygon; (b) sawtooth; (c) reentry

式中: $c_{bi}$ 为 $p_i$ 点的边界检测值; $c_{bj}$ 为 $p_j$ 点的边界检测值。由于越靠近边界上的点,其 $c_b$ 值越大,相应地, $w_p(i, j)$ 的值就越小,因此在构建最小生成图时会被优先选取,从而避免了边界向轮廓内部生长形成锯齿状多边形。第二部分为 $w_d(i, j)$ ,它通过引入点的局部采样密度计算得到,即

$$w_d(i, j) = \frac{2 \| \mathbf{p}_i - \mathbf{p}_j \|}{r_{p_i} + r_{p_j}}, \quad (5)$$

式中: $r_{p_i}$ 为待测点 $p_i$ 到其邻域点距离的平均值; $r_{p_j}$ 为待测点 $p_j$ 到其邻域点距离的平均值。 $w_d(i, j)$ 的值与边 $e(i, j)$ 的长度正相关,因此,相同条件下短边会被优先选取用来构建最小生成图,从而使得最终得到的孔洞多边形包含更多的边界点。

采用局部采样密度是为了适应采样不均匀的点云模型。综上,  $p_i$  和  $p_j$  构成的边的权值  $w(i, j)$  可以表示为

$$w(i, j) = w_p(i, j) + w_d(i, j). \quad (6)$$

最小生成图采用类似于 Kruskal 的最小生成树算法在点云邻接图(即以点云模型所有顶点为图的顶点, 点云中各点与相应邻域内各点构成的边为图的边)中构建, 最后得到的最小生成图中允许边数大于一定阈值  $\rho$  (阈值  $\rho$  的选取根据需要提取孔洞的大小确定)的圈出现。在算法运行的初期, 邻接图中的每一个顶点都是一个独立的单元, 然后将图中的所有有效边根据权值的大小从小到大排列, 这里的有效边是指  $w_p(i, j)$  和  $w(i, j)$  都小于预设阈值的边。通过不相交集的 Union/Find 算法查找边  $e(i, j)$  的两个端点, 然后判断它们是否在同一个单元内。若不在同一单元, 则将边  $e(i, j)$  加入到最小的生成图中, 合并这两个单元; 否则, 需要判断加入边  $e(i, j)$  后在图中形成的圈的边数, 若边数大于预设阈值, 同样将边  $e(i, j)$  加入到最小生成图中。

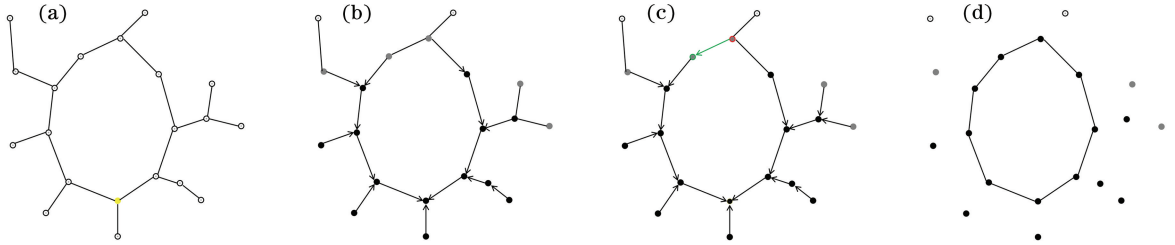


图3 最小生成树中孔洞边界点的排序。(a)边界最小生成图;(b)经过4次迭代后各点的状态,箭头指向父节点;(c)相邻点中出现灰色点,代表找到一个孔洞轮廓;(d)最终提取的孔洞多边形

Fig. 3 Sorting of hole boundary points in minimum spanning tree. (a) Minimum generated graph of boundary; (b) state of each point after four iterations, the arrow points to the parent node; (c) gray points appear in the adjacent points, which means that a hole outline is found; (d) final extracted hole polygon

## 4 实验及结果分析

为了验证所提算法的有效性, 本文算法基于点云库 PCL(Point Cloud Library)以 C++ 语言实现, 在内存为 3 GB 的 Inter Core i3-6100 CPU 计算机上对不同的点云模型进行测试。

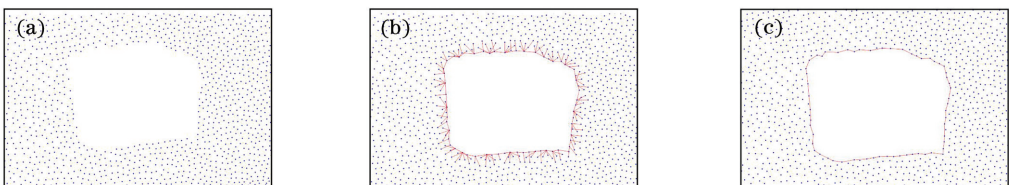


图4 汽车摇臂点云模型中截取的孔洞边界提取效果。(a)孔洞附近的点云;(b)最小生成图;(c)提取的孔洞多边形

Fig. 4 Extraction effect of hole boundary in the car rocker point cloud model. (a) Point cloud near the hole; (b) minimum generated map; (c) extracted hole polygon

在提取圈的边数时, 采用 BFS(Breath First Search) 算法查找最小生成图中两点的最短路径。

### 3.2 边界特征点的排序

在得到边界的最小生成图后, 同样采用 BFS 算法对孔洞环上的点进行排序, 以便后续拟合孔洞轮廓线。BFS 算法首先对最小生成图中每一个未访问的顶点执行一次广度优先搜索, 在搜索过程中以三种状态标记各顶点的处理情况。假设三种状态分别为白色、灰色和黑色, 白色代表未处理的点, 黑色表示已经处理过的点, 灰色则代表待处理的顶点。从图 3(a)中的最小生成图中任意取一点作为起始点, 将其状态标为灰色, 并加入到储存待处理点的队列中等待处理。每次从队列中弹出队首元素, 把相对于队首元素深度为 1 的白色点标为黑色, 把深度为 2 的白色点标为灰色。在记录各点的父节点后把灰色点压入队尾。在访问过程中, 如果相邻点为黑色则忽略, 若为灰色则代表已经形成一个圈, 此时沿着这两个节点反向查找其父节点即可将孔洞边界点排好序。具体过程如图 3 所示。

为了清晰地展示本文所提孔洞轮廓提取算法的过程, 本文给出了汽车发动机摇臂实测点云模型中一个孔洞轮廓线的提取过程。图 4(a)为点云模型中孔洞的放大示意图, 图 4(b)为在  $w_p(i, j) > 1.5$ 、 $w(i, j) > 2.5$  且  $\rho = 10$  的情况下生成的最小生成图, 图 4(c)为最终得到的孔洞多边形。在实验过程

中减小  $\omega_p(i, j)$  和  $\omega(i, j)$  的选取范围, 则在构建最小生成图时就可减少可选择边的数量, 因此算法的运行时间缩短; 但取值过小会出现最小生成图中没有封闭圈的情况, 导致孔洞多边形提取失败。  $\rho$  的取值应当根据孔洞的大小进行选择, 过大会忽略掉小孔洞, 过小则会在孔洞轮廓附近形成多个封闭圈, 影响后续孔洞边界点的排序。

为了验证所提算法在其他模型中出现多个孔洞时的轮廓提取效果, 本文选取挖掘机斗齿、链轨节以及转向节的实测点云模型作为实验对象。斗齿模型包含 56208 个坐标点, 算法运行时间为 83.4860 s, 在斗齿的不同侧面上分布着大小不等的 8 个孔洞,

其背面的大面积孔洞是零件与测量台的接触部位, 在其两侧的销孔位置处人为挖了两个相对较小的孔洞, 其他孔洞为扫描过程中由扫描速率不均匀造成的孔洞。相对于斗齿模型, 链轨节具有更加复杂的几何结构, 链轨节模型共包含 80254 个坐标点, 算法运行时间为 87.7890 s, 链轨节上分布着 8 个人为孔洞。相对于斗齿模型上的单面孔洞, 链轨节上存在 2 个跨面孔洞, 这更能验证所提算法的适用性。

从图 5 和图 6 中可以看出, 所提算法能有效提取模型中不同类型的封闭孔轮廓, 且算法的运行时间能满足实际应用要求, 相对于文献[15]中的算法得到的孔洞多边形更加平滑。

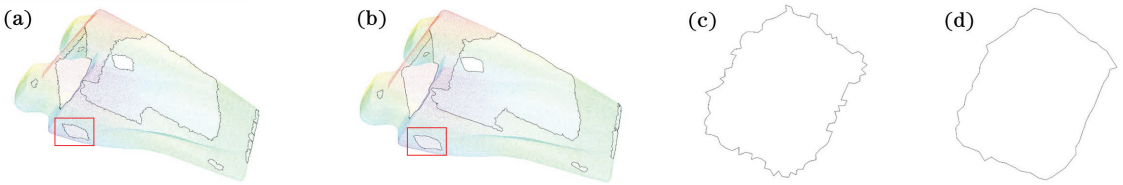


图 5 斗齿模型边界检测效果的对比。(a)(c)文献[15]中算法提取的孔洞及其放大图;  
(b)(d)本文算法提取的孔洞及其放大图

Fig. 5 Comparison of boundary detection effects of bucket teeth model. (a) Hole extracted by the algorithm in Ref. [15] and its enlargement; (b)(d) hole extracted by the algorithm in this paper and its enlargement

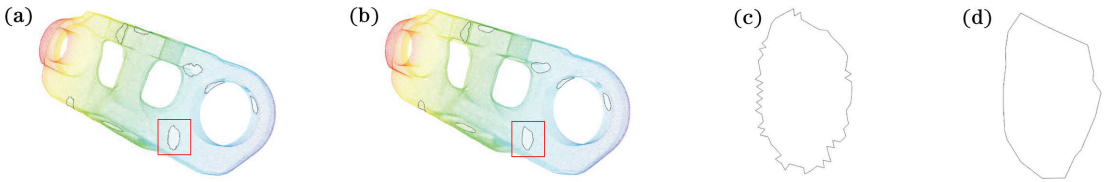


图 6 链轨节模型边界检测效果的对比。(a)(c)文献[15]中算法提取的孔洞及其放大图;  
(b)(d)本文算法提取的孔洞及其放大图

Fig. 6 Comparison of boundary detection effect of chain link model. (a)(c)Hole extracted by the algorithm in Ref. [15] and its enlargement; (b)(d) hole extracted by the algorithm in this paper and its enlargement

对于斗齿模型尾端的长条形孔洞, 文献[15]中的算法将该孔洞分割为 5 个子孔洞, 本文算法则将其分割成 2 个子孔洞, 有利于提高孔洞修补的效率。另外, 由于转向节模型比较复杂, 边界点检测结果中

存在大量的误判点, 如图 7(a) 所示, 使得文献[15]中的孔洞多边形生长算法不能正确地选取起始点, 孔洞多边形提取失败, 如图 7(b) 所示。图 7(c) 为本文方法提取到的 3 个孔洞多边形。

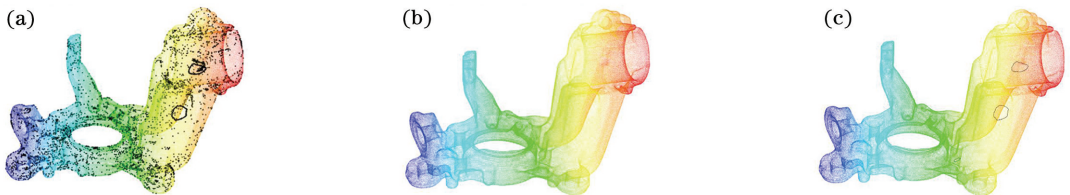


图 7 转向节模型边界检测效果的对比。(a) 包含较多误判点的边界检测结果; (b) 文献[15]  
中的算法未能提取到孔洞多边形; (c) 本文算法提取的孔洞多边形

Fig. 7 Comparison of boundary detection effect of knuckle model. (a) Boundary detection result containing more misjudgment points; (b) the algorithm in Ref. [15] fails to extract hole polygon; (c) hole polygon extracted by the algorithm in this paper

文献[2]、[16]和[17]中描述的孔洞边界检测方法通过设计边界检测算子并设定相应的阈值提取孔洞边界点,在模型存在多个孔洞的情况下,上述算法将所有的孔洞边界点作为整体输出,后续修补孔洞时需要人为选取各个孔洞边界;而本文算法在整体点云模型中进行检测,并依次输出排好序的各个孔洞轮廓点,有利于实现后续修补算法的自动化。由于孔洞形状的不规则性,提取的孔洞轮廓点可能会有稀疏或者不均匀分布的情况,从而影响了后续的孔洞修补。本文算法在输出规则排列的孔洞边界点后采用三次B样条拟合孔洞轮廓线,然后再次采样获得了更加丰富、均匀的孔洞轮廓点。

## 5 结 论

通过分析邻域点的协方差矩阵提出了一种孔洞边界检测算子,然后采用计算机图论的方法逐个输出规则化的孔洞轮廓点。相对于现有的孔洞边界检测算法,所提边界检测算子直接在三维空间中分析邻域点协方差矩阵特征值之间的关系,进而提取孔洞边界点,避免了局部平面的拟合、投影等步骤,提高了边界检测算法的效率。此外,所提算法采用计算机图论的方法提取孔洞多边形,相对于仅以距离为参考的孔洞多边形生成算法,提取的孔洞多边形具有连续、光滑的优点,避免了锯齿状轮廓线以及折返现象的发生。针对点云模型的多孔洞问题,所提算法在提取孔洞轮廓线的同时完成了孔洞聚类。一般地,由于孔洞的不规则性,提取出的孔洞边界点会出现稀疏以及不均匀现象,采用本文算法生成孔洞轮廓线后再次采样可以得到更加丰富、均匀分布的孔洞轮廓点。

## 参 考 文 献

- [1] Rosenfeld A, Thurston M. Edge and curve detection for visual scene analysis[J]. IEEE Transactions on Computers, 1971, C-20(5): 562-569.
- [2] Ando S. Image field categorization and edge/corner detection from gradient covariance[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22(2): 179-190.
- [3] Lü Y Z, Sun Q, Bi G L. Recognition and repairing of surface hole in three dimensional laser scanning system[J]. Chinese Optics, 2016, 9(1): 114-121.  
吕源治, 孙强, 毕国玲. 三维激光扫描系统中曲面空洞的识别与修复[J]. 中国光学, 2016, 9(1): 114-121.
- [4] Su Y L, Ping X L. Point cloud edge-extraction algorithm based on Gaussian map clustering [J]. Laser & Optoelectronics Progress, 2019, 56(11): 111506.  
苏云龙, 平雪良. 基于高斯映射聚类的点云边缘提取算法[J]. 激光与光电子学进展, 2019, 56(11): 111506.
- [5] Liu Q, Zhang G, Chen X J. Point cloud feature regularization based on fusion of improved field force and judging criterion[J]. Chinese Journal of Lasers, 2019, 46(4): 0404009.  
刘庆, 章光, 陈西江. 融合改进场力和判定准则的点云特征规则化 [J]. 中国激光, 2019, 46(4): 0404009.
- [6] Zeng L L, Gai S Y, Da F P, et al. Hole filling algorithm of three-dimensional point cloud based on structure from motion[J]. Acta Optica Sinica, 2018, 38(6): 0610002.  
曾露露, 盖绍彦, 达飞鹏, 等. 基于从运动中恢复结构的三维点云孔洞修补算法研究 [J]. 光学学报, 2018, 38(6): 0610002.
- [7] Wang Y T, Feng H Y. Outlier detection for scanned point clouds using majority voting [J]. Computer-Aided Design, 2015, 62: 31-43.
- [8] Kleppe A L, Tingelstad L, Egeland O. Coarse alignment for model fitting of point clouds using a curvature-based descriptor[J]. IEEE Transactions on Automation Science and Engineering, 2019, 16(2): 811-824.
- [9] Lin C H, Chen J Y, Su P L, et al. Eigen-feature analysis of weighted covariance matrices for LiDAR point cloud classification [J]. ISPRS Journal of Photogrammetry and Remote Sensing, 2014, 94: 70-79.
- [10] Raina P, Mudur S, Popa T. MLS2: sharpness field extraction using CNN for surface reconstruction[C]// Graphics Interface. Toronto, Ontario: [s.n.], 2018: 66-75.
- [11] Raina P, Mudur S, Popa T. Sharpness fields in point clouds using deep learning[J]. Computers & Graphics, 2019, 78: 37-53.
- [12] Tuzel O, Porikli F, Meer P. Region covariance: a fast descriptor for detection and classification[M]// Computer Vision-ECCV 2006. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006: 589-600.
- [13] Cirujeda P, Dicente Cid Y, Mateo X, et al. A 3D scene registration method via covariance descriptors and an evolutionary stable strategy game theory

- solver[J]. International Journal of Computer Vision, 2015, 115(3): 306-329.
- [14] Tabia H, Laga H. Covariance-based descriptors for efficient 3D shape matching, retrieval, and classification[J]. IEEE Transactions on Multimedia, 2015, 17(9): 1591-1603.
- [15] He X M, Zhuo Y S, Pang M Y. An algorithm for extracting hole-boundary from point clouds [J]. Transactions of the Chinese Society for Agricultural Machinery, 2014, 45(2): 291-296.  
何学铭, 嵇永盛, 庞明勇. 点云模型中孔洞边界曲线多准则提取算法[J]. 农业机械学报, 2014, 45(2): 291-296.
- [16] Chen Y R, Wang Y B, Peng Z J, et al. Improved algorithm for extraction of boundary characteristic point from scattered point cloud [J]. Computer Engineering and Applications, 2012, 48(23): 177-180, 190.  
陈义仁, 王一宾, 彭张节, 等. 一种改进的散乱点云边界特征点提取算法[J]. 计算机工程与应用, 2012, 48(23): 177-180, 190.
- [17] Sun D Z, Fan Z X, Li Y R. Automatic extraction of boundary characteristic from scatter data[J]. Journal of Huazhong University of Science and Technology (Nature Science Edition), 2008, 36(8): 82-84.  
孙殿柱, 范志先, 李延瑞. 散乱数据点云边界特征自动提取算法[J]. 华中科技大学学报(自然科学版), 2008, 36(8): 82-84.