

## 点云模型谷脊特征的提取与增强算法

庞旭芳<sup>1,2</sup> 庞明勇<sup>1</sup> 肖春霞<sup>3</sup>

**摘要** 提出一种鲁棒性的谷脊线提取与增强算法. 算法采用多步逼近的策略: 首先根据每个点的局部最小二乘拟合曲面多项式计算每个点的主曲率, 并用绝对值较大的主曲率标识出谷脊潜在特征点; 然后将特征点投影到离其最近的潜在特征线上得到增强的特征点; 再对增强后的特征点进行平滑, 选择合适的平滑点生成特征折线; 最后再对特征线进行进一步的扰动滤除等操作得到光滑的谷脊线. 实验结果表明, 本文算法稳定、抗噪性强、能满足多分辨率的特征提取要求.

**关键词** 点云模型, 谷脊特征提取, 特征增强, 谷, 脊, 移动最小二乘

**DOI** 10.3724/SP.J.1004.2010.01073

## An Algorithm for Extracting and Enhancing Valley-ridge Features from Point Sets

PANG Xu-Fang<sup>1,2</sup> PANG Ming-Yong<sup>1</sup> XIAO Chun-Xia<sup>3</sup>

**Abstract** We present a robust algorithm for extracting valley-ridge lines from point set model. Our algorithm is based on multi-step refinement operations: Using moving least square method, we fit a smooth patch for neighborhood of each point, then calculate the principal curvature for each point. Potential valley-ridge points are identified according to the biggest principal curvature. After enhancing valley-ridge points by projecting onto their closest potential feature lines, we smooth the projected points and grow polylines through the smoothed points. Finally, smooth valleys and ridges are achieved by resolving the results. Experiments indicate that our algorithm has good stability and strong anti-noise performance, and it also leads to good results of multi-resolution extractions of features.

**Key words** Point-set, valley-ridge extraction, feature enhancement, valley, ridge, moving least squares (MLS)

特征提取与增强是数字几何模型处理中备受关注的底层技术之一. 谷脊特征线是模型上顺应曲面凹凸变化趋势的曲线, 能较好地描述模型的特征. 提取模型的谷脊线在几何分析、曲面重建与编辑、数据分块以及人脸识别等领域有着广泛的应用, 也为几何特征提取与增强的研究提供了重要途径.

几何造型领域已对三角网格模型与点集模型的特征提取进行了较深入的研究. 本文将特征提取算法分为三类: 1) 基于法向偏差的方法, 如 Demarsin<sup>[1]</sup> 用主元分析法 (Principal component analysis, PCA) 计算点的法向, 通过邻点间法向变

化的大小来检测并提取出特征点. 2) 基于协方差分析的方法, 如 Pauly 等<sup>[2]</sup> 提出首先用基于局部邻域的协方差分析方法, 标记出潜在的特征点, 再通过变换邻域的半径, 多尺度地处理含有噪声的模型; Gumhold 等<sup>[3]</sup> 使用黎曼图建立局部邻域点之间的连接关系, 同时用协方差分析方法为每个点计算一个用于标记其为潜在棱点、边点以及角点的权重. 文献 [2-3] 都通过建立最小生成树连接标记的潜在特征点, 并使用曲线来逼近尖锐特征. 上述两类方法对模型的质量要求较高, 在处理稀疏模型时的稳定性有待提高, 同时协方差分析方法对模型微分性质的敏感度是有限的, 无法提取细小的特征. 3) 基于投影映射的方法, 如 Daniels 等<sup>[4]</sup> 用基于最小平方中值 (Least median of squares, LMS)<sup>[5]</sup> 和前向搜索法<sup>[6]</sup> 的鲁棒移动最小二乘法 (Robust moving least squares, RMLS)<sup>[7]</sup> 在每个点的邻域内进行曲面拟合, 得到该邻域内的一系列拟合曲面, 然后将该点投影到距其最近的曲面交线上, 从而得到增强的特征点, 最后用主元分析方法直接连接特征点, 并对特征线进行平滑松弛处理得到光滑的特征线. 该算法包含投影增强运算, 在处理稀疏的点集模型时也能得到比较满意的特征提取效果, 但由于算法是基于 RMLS 的, 时间代价较高.

收稿日期 2009-07-15 录用日期 2009-12-02

Manuscript received July 15, 2009; accepted December 2, 2009  
国家自然科学基金 (060873175), 江苏省教育厅江苏高校自然科学基金  
基础研究项目 (07KJD460108), 江苏省 2009 年度普通高校研究生科研  
创新 (CX09S-009R), 南京师范大学优秀硕士论文培育项目资助

Supported by National Natural Science Foundation of China  
(060873175), Natural Science Foundation of the Jiangsu Higher  
Education Institutes of China (07KJD460108), 2009 Innovation  
Plan for University Postgraduates in Jiangsu Province of China  
(CX09S-009R), and Outstanding MSc Thesis Incubation Program  
of Nanjing Normal University

1. 南京师范大学教育技术系 南京 210097 2. 中国科学院深圳先进  
技术研究院文化集成工程研究室 深圳 518055 3. 武汉大学计算机学  
院 武汉 430072

1. Department of Educational Technology, Nanjing Normal  
University, Nanjing 210097 2. Laboratory for Culture Integra-  
tion Engineering, Shenzhen Institute of Advanced Technology,  
Chinese Academy of Sciences, Shenzhen 518055 3. Computer  
School, Wuhan University, Wuhan 430072

文献 [8] 给出一种基于特征的数据分块方法, 方法将粗略分块与精确分块相结合, 在分析曲面边界区域微分几何特性的基础上, 根据高斯曲率和平均曲率将点分为平面点、谷、脊、极小曲面、鞍形脊、鞍形谷、峰、阱等八类. 完成对散乱点云的粗略分块后, 采用基于特征点约束的区域生长法完成精确分块. Lai 等<sup>[9]</sup> 提出了一种特征分类方法: 该算法通过计算每个点一定半径测地圆的面积与周长二次方比值得到该点的紧性测度, 同时计算测地圆上离该点的最近距离. 紧性测度和测地圆最短“半径”值共同将特征点提取出来, 然后计算该点的法向与该点指向其局部逼近平面上对应点的矢量的点乘, 通过点乘结果的符号来判断该点的凹凸性, 进而将谷特征和脊特征分离开. 文献 [10] 针对三角网格模型提出一种谷脊线提取算法, 该算法使用紧性径向函数 (Compactly supported radial basis functions, CS-RBFs)<sup>[11]</sup> 拟合三角网格曲面, 并通过将网格点投影到 CS-RBF 曲面上, 计算其曲率张量和曲率梯度. 然后计算每条网格边两端点的最大/最小主曲率方向并跟踪检测出所有的谷脊特征网格点. 如果网格边的两端点都是谷特征点或者脊特征点, 则直接将它们连接起来; 如果网格三角形的三个顶点都是谷特征点或者脊特征点, 则取三角形的重心为特征点.

## 1 算法梗概

问题描述: 对于给定的一个不含有法向信息和任何连接信息的点云模型:  $\mathbf{P} = \{p_i\}$ ,  $p_i \in \mathbf{R}^3$ ,  $i \in \{1, \dots, N\}$ , 本文算法提取模型  $\mathbf{P}$  的谷脊线.

本文针对点云模型提出一种谷脊线提取算法, 算法基于多步逼近策略: 首先计算每个点的主曲率并根据绝对值较大的主曲率分别标记出谷潜在特征点和脊潜在特征点, 再用权重敏感的最小二乘法计算每个特征点在一定半径范围内的潜在特征线, 并将该点投影到离其最近的潜在特征线上, 然后用 PCA 方法对投影点进行自适应平滑处理, 最后对生成的特征折线进行优化得到光滑的谷脊线. 算法通过以下四个步骤来完成, 如图 1 (b)~1 (e) 所示:

1) 计算点的主曲率, 提取潜在的谷脊特征点: 计算每个点的局部逼近平面, 并在平面上建立  $Z$  轴朝向待重建曲面外部的局部坐标系, 用移动最小二乘法 (Moving least squares, MLS)<sup>[12]</sup> 在局部坐标系内拟合该点邻域, 得到局部曲面拟合多项式, 并计算该点的主曲率. 根据每个点的绝对值较大的主曲率分别标识出谷脊特征点. 曲率为负且绝对值大于一定阈值的点加入谷潜在特征点集, 曲率为正且大

于一定阈值的点加入脊潜在特征点集, 然后分别对两个点集执行以下步骤.

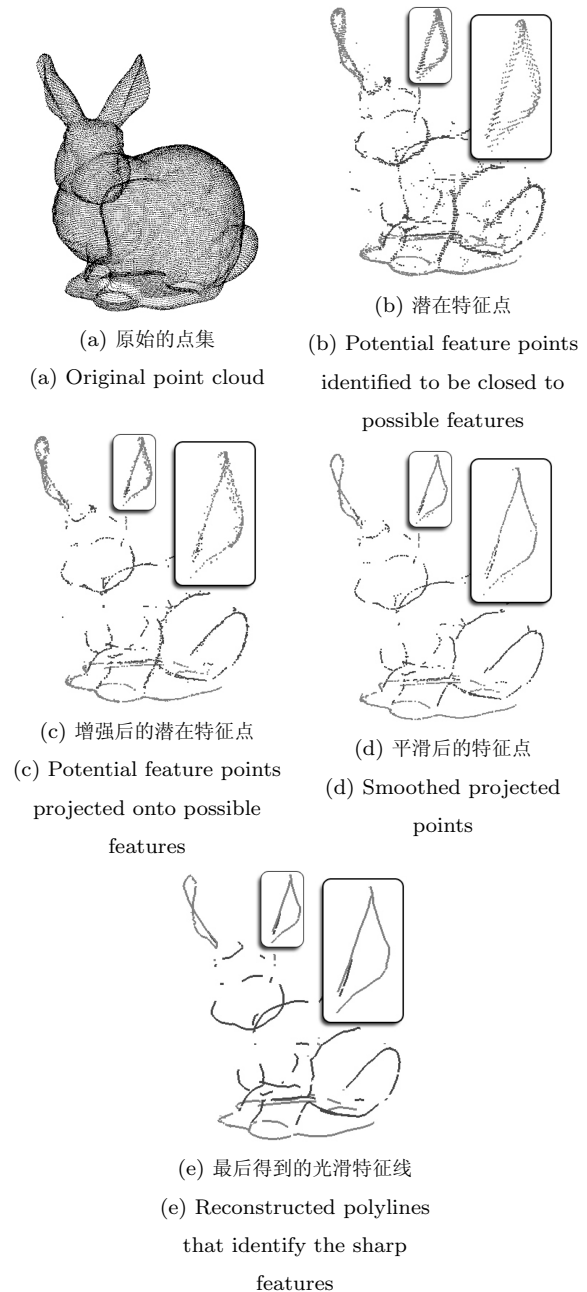


图 1 本文特征提取方法的流程图

Fig. 1 Over view of our feature extraction pipeline

2) 增强潜在谷脊特征点: 用权重敏感的最小二乘法在每个潜在特征点局部逼近平面上拟合局部潜在特征线的投影, 然后利用最小二乘曲面拟合多项式间接地表示潜在特征线, 最后将该点投影到离其最近的潜在特征线上得到增强的特征点.

3) 平滑谷脊特征点: 首先对每个特征点的邻域进行相关系数分析, 自适应地调整邻域半径, 然后用 PCA 方法找出主轴, 并将该点投影到主轴上得到平

滑的特征点.

4) 生成谷脊特征线并对其进行优化: 根据一定的特征折线“生长原则”生成逼近谷脊线的初始特征折线. 然后分析谷脊特征折线端点的情况, 修复谷脊线之间的裂缝, 并对其进行平滑处理.

## 2 谷脊潜在特征点的提取

本文算法通过点云中每个点  $p_i$  绝对值较大的主曲率  $k_i$  来识别潜在的谷脊特征点, 谷点为相对待重建曲面凹的部分, 其对应的  $k_i$  小于 0; 而脊点为相对待重建曲面凸的部分, 其对应的  $k_i$  大于 0.

用移动最小二乘法拟合  $p_i$  点在  $r$  半径邻域内的点  $NBHD(p_i) = \{p_j\}, \|p_j - p_i\| \leq r, j = 0, \dots, k$ , 得到  $p_i$  点局部曲面拟合多项式  $g_i$ , 然后通过  $g_i$  点的曲率. 计算局部曲面拟合多项式分为两步: 1) 寻找支撑平面, 建立局部坐标系; 2) 计算多项式.

首先建立  $NBHD(p_i)$  的协方差矩阵,

$$B = \sum_{p_j \in NBHD(p)} (p_j - o_i)(p_j - o_i)^T$$

其中

$$o_i = \frac{1}{k} \sum_{j=0}^k p_j$$

是  $NBHD(p_i)$  的重心,  $B$  是一个对称的半正定  $3 \times 3$  矩阵. 计算  $B$  的特征值和特征向量, 设特征值为  $\lambda_0, \lambda_1, \lambda_2$  ( $\lambda_0 < \lambda_1 < \lambda_2$ ),  $\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2$  分别为  $\lambda_0, \lambda_1, \lambda_2$  对应的特征向量.  $\mathbf{v}_1, \mathbf{v}_2, o_i$  共同定义逼近  $NBHD(p_i)$  的平面  $H_i$ , 也称之为支撑平面,  $\mathbf{v}_0$  即为  $o_i$  的法向, 记为  $\mathbf{n}_i$ . 在  $H_i$  平面上建立局部坐标系  $(o_i, \mathbf{u}, \mathbf{v}, \mathbf{n}_i)$ ,  $o_i$  为原心,  $\mathbf{n}_i$  为  $Z$  轴.

实验结果显示上述方法得到的局部坐标系  $Z$  轴的方向有的指向待重建曲面的内部, 有的则指向外部, 见图 2(a). 平均曲率和主曲率都与  $Z$  轴的朝向有关, 所以为使点的凹凸性判断有一个统一的参照, 需要调整每个局部坐标系的  $Z$  轴, 使其指向待重建曲面的外部. 局部坐标系  $Z$  轴的调整过程也就是对点  $o_i$  的法向  $\mathbf{n}_i$  进行调整的过程.

本文算法从点集  $O = \{o_i\}, i \in \{1, \dots, N\}$  中选择 6 个稳定传播源点, 分别为  $x$ 、 $y$ 、 $z$  分量最大和最小的点. 然后从这些传播源点出发对  $O$  中每个点进行法向传播调整.  $o_i$  点  $r$  半径内的点记为  $NBHD(o_i) = \{o_j\}, j = 0, \dots, k$ , 相邻点  $o_i, o_j$  的法向点乘绝对值  $\omega_j$  用于标识两点之间的法向传播能力, 即  $\omega_j = |\mathbf{n}_i \cdot \mathbf{n}_j|$ ,  $\mathbf{n}_i$  和  $\mathbf{n}_j$  都是单位法向, 所以  $1 \geq \omega_j \geq 0$ . 当两点法向越接近平行, 则  $\omega_j$  越接

近 1, 表示两点间的传播能力越强. 设置传播阈值  $\tau$ , 本文取  $\tau = 0.6$ , 当  $\omega_j \geq \tau$  时, 表明  $o_i$  和  $o_j$  之间存在法向连续性, 可进行法向传播, 即如果  $\mathbf{n}_i \cdot \mathbf{n}_j < 0$ , 则  $\mathbf{n}_j$  取  $-\mathbf{n}_j$ . 当  $\omega_j < \tau$  时证明  $o_i$  和  $o_j$  之间不能进行法向传播. 我们为  $o_i$  设置两个标志: bNormal, bNVisited, 分别表示该点是否完成法向调整, 以及该点  $r$  半径内的邻点是否全部完成法向调整. 算法采用局部深度优先策略, 在传播的过程中选择法向已经调整好, 但其  $r$  半径内邻近点法向未全部被调整的点作为候选传播源点压入堆栈, 当一次传播终止后, 从堆栈中弹出一个元素, 作为下一次法向传播的源点. 多源点法向传播算法具体过程如下:

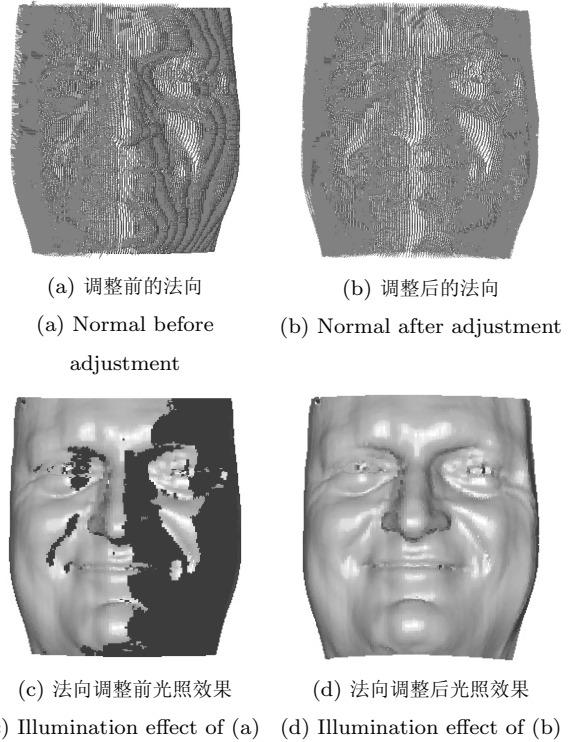


图 2 Fran-rgb 模型的法向调整

Fig. 2 Normal adjustment for fran-rgb model

1) 调整 6 个稳定源点的法向, 即使  $x$  分量最大点的法向指向  $+X$ ,  $x$  分量最小点的法向指向  $-X$ , 依此类推.

2) 建立 6 个堆栈 CSX, CSX-, CSY, CSY-, CSZ, CSZ-, 用于存储从上述 6 个源点出发传播过程中产生的候选传播源点.

3) 依次从 6 个源点  $o$  进行法向传播, 从  $NBHD(o)$  中选择  $\omega_j \geq \tau$  且其 bNormal 标记为 False 的点, 按照  $\omega_j$  的值, 从大到小依次对  $p_j$  点的法向进行调整, 并将其 bNormal 标识设置为 True. 同时按照  $o$  邻近点法向传播的反顺序将  $o_j$  压入其

对应的堆栈 (如从  $x$  分量最大源点出发, 则将候选点压入堆栈 CSX), 即  $\omega_j$  值越大越后压入,  $\omega_j$  最大的点置于栈顶. 如果  $NBHD(o)$  中点的法向全部被调整过了, 则将点  $o$  的 `bNVisited` 标记设置为 `True`.

4) 依次从 6 个堆栈弹出栈顶元素, 如果弹出的栈顶点其 `bNVisited` 为 `True`, 则继续弹出下一个元素, 如果弹出的栈顶点其 `bNVisited` 为 `False` 并且检测其  $r$  半径内邻近点未全部被调整, 则该点为源点, 若检测其  $r$  半径内邻近点已全部调整, 则将该点的 `bNVisited` 标记为 `True`, 并继续从该堆栈弹出点, 直至找到满足条件的源点, 然后重复步骤 3).

5) 直到 6 个堆栈都为空, 再对整个点云进行检测, 找到法向未被调整的点, 按照三次最近距离法<sup>[8]</sup>继续对其进行法向调整, 直到所有点的法向都被调整完.

在极少数特殊情况下, 6 个源点中会有重合的点, 排除重复点后再完成以上步骤. 图 2 (b) 为调整后的效果. 该法向调整算法适用于封闭模型, 而对于其他类型的模型, 可以根据模型的特点选择合适的几个传播源点, 如单片模型 (参见图 2 模型), 则使用一个方向的法向传播即可.

$p_i$  点局部坐标系的  $Z$  轴取为  $o_i$  点调整后的法向方向  $\mathbf{n}_i$ , 然后在  $(o_i, \mathbf{u}, \mathbf{v}, \mathbf{n}_i)$  坐标系内用二次多项式  $g_i$  拟合  $NBHD(p_i)$ , 使  $g_i$  满足式 (1):

$$\min \sum_{p_j \in NBHD(p_i)} \langle p_j - p_g, \mathbf{n}_i \rangle^2 \quad (1)$$

其中,  $p_g$  表示  $p_j$  沿着  $\mathbf{n}_i$  在多项式曲面  $g_i$  上的投影. 式 (1) 的意义是: 使  $p_j$  点与其在拟合曲面上投影点的距离平方和最小. 设  $g_i$  表达式为

$$g_i(u, v) = a + bu + cv + duv + eu^2 + fv^2 \quad (2)$$

式 (1) 可以改写成

$$\min \sum_{p_j \in NBHD(p_i)} ((p_j - o_i) \cdot \mathbf{n}_i - g(u_j, v_j))^2 \quad (3)$$

$(u_j, v_j)$  为  $p_j$  在局部坐标系内  $(\mathbf{u}, \mathbf{v})$  平面上的坐标. 由式 (3) 对多项式的各个系数求一阶导数得到 6 个线性方程, 然后用高斯消元法计算出  $g_i$  的各个系数.  $(\mathbf{u}, \mathbf{v})$  平面上任一点  $q$  在  $g_i$  上的投影点为  $o_i + (u_q, v_q, g(u_q, v_q))$ .

求得  $g_i$  的表达式后, 计算  $p_i$  点的曲率<sup>[13]</sup>, 由式 (2) 可得

$$\mathbf{g}_u = (1, 0, b + dv + 2eu), \quad \mathbf{g}_v = (0, 1, c + dv + 2fv)$$

$$\mathbf{g}_{uu} = (0, 0, 2e), \quad \mathbf{g}_{vv} = (0, 0, 2f), \quad \mathbf{g}_{uv} = (0, 0, d)$$

$$\mathbf{n} = \frac{\mathbf{g}_u \times \mathbf{g}_v}{|\mathbf{g}_u \times \mathbf{g}_v|}$$

由此, 可得逼近曲面的第一和第二基本量:

$$L = \mathbf{g}_{uu} \cdot \mathbf{n} = \frac{2e}{\sqrt{(b + dv + 2eu)^2 + (c + du + 2fv)^2 + 1}}$$

$$E = 1 + (b + dv + 2eu)^2$$

$$M = \mathbf{g}_{uv} \cdot \mathbf{n} = \frac{d}{\sqrt{(b + dv + 2eu)^2 + (c + du + 2fv)^2 + 1}}$$

$$F = (b + dv + 2eu)(c + du + 2fv)$$

$$N = \mathbf{g}_{vv} \cdot \mathbf{n} = \frac{2f}{\sqrt{(b + dv + 2eu)^2 + (c + du + 2fv)^2 + 1}}$$

$$G = 1 + (c + du + 2fv)^2$$

至此, 可得曲面的高斯曲率

$$k_g = \frac{4ef - d^2}{((b + dv + 2eu)^2 + (c + du + 2fv)^2 + 1)^2}$$

与平均曲率  $k_h$  (见置底通栏公式). 进而可得曲面的主曲率:

$$k_1 = k_h + \sqrt{k_h^2 - k_g}, \quad k_2 = k_h - \sqrt{k_h^2 - k_g}$$

计算  $p_i$  点的高斯曲率、平均曲率以及该点的主曲率, 通过分析可知用绝对值较大的主曲率能更好地标识出谷脊特征, 如图 3 所示.  $k_i$  取  $k_1, k_2$  中绝对值较大的曲率值. 设定一个曲率阈值  $\alpha$ , 当  $k_i$  小于 0 且  $|k_i| > \alpha$  时, 将  $p_i$  点加入谷点点集  $V$ , 当  $k_i > \alpha$  时, 将  $p_i$  点加入脊点点集  $R$ , 如图 4 所示, 灰色点为脊点, 黑色点为谷点.

最小二乘曲面的拟合半径  $r$  会影响谷脊特征检测的敏感度, 见图 4. 拟合半径越大, 特征被平滑得越多. 为使算法能自适应地根据模型选择合适的半径  $r$ , 本文计算模型平均点距值  $d$ , 即每个点与其邻近 5 ~ 7 个点的平均距离的均值. 为减少计算量, 可以计算模型中任意  $m$  个点与其邻近点的平均距离的均值 ( $m > N/20$ ). 算法以平均点距值作为半径  $r$  的选择参照,  $r$  一般取 2 ~ 4 倍  $d$ .

分别对  $V$  和  $R$  执行步骤 2) ~ 4), 最后分别得到光滑的谷特征线和脊特征线. 因为谷和脊可以通过转换坐标  $Z$  轴的朝向相互转换, 所以只需研究谷或者脊. 为方便表述, 下文中用  $F$  来任意表示  $V$  或者  $R$ .

$$k_h = \frac{e + f + e(c + du + 2fv)^2 + f(b + dv + 2eu)^2 - d(b + dv + 2eu)(c + du + 2fv)}{(\sqrt{(b + dv + 2eu)^2 + (c + du + 2fv)^2 + 1})^3}$$

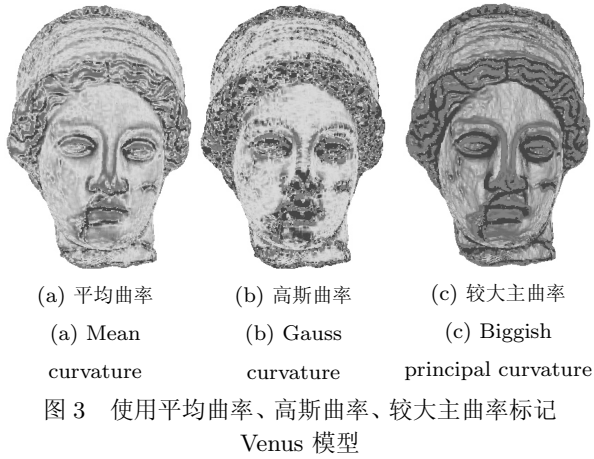


图3 使用平均曲率、高斯曲率、较大主曲率标记 Venus 模型

Fig.3 Venus models marked according to the mean curvature, Gauss curvature, and biggish principal curvature, respectively

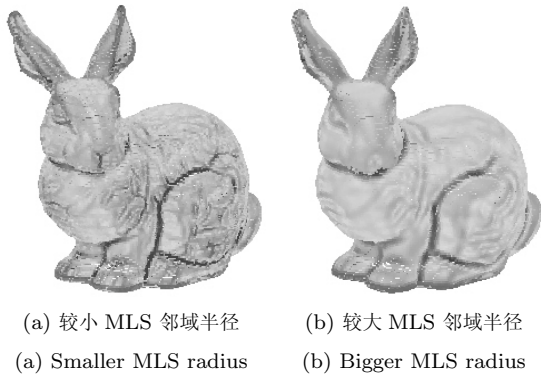


图4 使用不同最小二乘曲面拟合半径的谷脊特征点提取效果

Fig.4 Valley-ridge points extraction results corresponding to different MLS radius

### 3 谷脊潜在特征点的增强

为使潜在特征点进一步逼近特征线, 将  $F$  中的每个点投影到其邻域内离其最近的潜在特征线上, 投影点即为增强的特征点. 本文算法首先在局部逼近平面  $H_i$  上拟合并逼近潜在谷脊特征线的投影, 然后, 利用最小二乘曲面拟合多项式  $g_i$ , 将投影曲线映射到点集曲面上得到潜在谷脊特征线的间接表示.

为计算潜在特征点邻域内的潜在特征线投影, 将  $F$  中每个点  $p_i$  及其  $\varphi$  半径内邻域点  $NBHD(p_i) = \{p_j\}$  投影到支撑平面  $H_i$  上, 并将投影点转换到局部坐标系  $(o_i, \mathbf{u}, \mathbf{v}, \mathbf{n}_i)$  内, 记为  $q_j$ ,  $p_j$  的曲率  $k_j$  为  $q_j$  的权重, 得到  $(\mathbf{u}, \mathbf{v})$  平面内的点集  $Q = \{q_j | k_j\}, Q \in \mathbf{R}^2$ .

为从  $Q$  中拟合出潜在特征线的投影, 本文采用一种权重敏感的最小二乘方法. 与一般应用中坐标敏感的最小二乘法不同, 该方法的拟合结果不局限

于点数据的坐标影响, 而是将点数据的权重影响纳入拟合结果, 让拟合曲线更接近权重较大的点, 即计算满足式 (4) 的曲线拟合多项式  $f$ .

$$\min \sum_{q_j \in Q} \|q_j - q_{fj}\|^2 \theta(\cdot) \quad (4)$$

$q_{fj}$  表示  $q_j$  在  $f$  上的投影,  $\theta(\cdot)$  是权函数, 本文为实现权重敏感性, 设  $\theta(\cdot) = k_j$ . 权重敏感拟合结果如图 5 所示. 黑色曲线为权重敏感拟合曲线, 可见权重敏感的最小二乘法能更接近权重较大的点, 从而更加准确地逼近潜在特征线的投影.

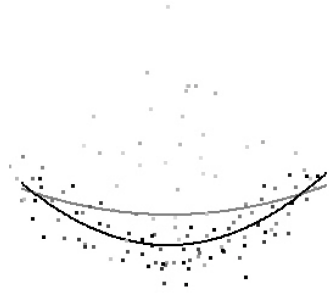


图5 最小二乘拟合结果 (灰色) 和权重敏感最小二乘拟合结果 (黑色) (二维散乱点的权重越大颜色越深)

Fig.5 Results of MLS fit (gray) and our weight sensitive MLS fit (black) (Two dimensional point set are marked by color according to the weight.)

受文献 [5] 中 LMS 算法和前向搜索方法<sup>[6]</sup> 的启发, 本文给出一种针对权重敏感最小二乘法的预处理算法: 根据权重从“种子”点开始进行“区域生长”, 找出对某条曲线拟合结果产生较大影响的一组点, 为表述方便称这组点为该曲线的曲线拟合点集. 该预处理方法能有效地提高权重敏感最小二乘算法的稳定性, 具体算法如下:

设  $W = \{W_j\}, j \in (1, \dots, m)$  为  $Q$  中所有曲线拟合点集的集合,  $W_0$  表示拟合第 0 条曲线的曲线拟合点集. 每条拟合曲线都有两个初始“生长点”, 设为  $G_1$  和  $G_2$ . 预处理算法首先根据“生长圆”规则计算每条曲线的初始“生长点”, 然后利用“区域生长”算法从两个初始“生长点”开始探测出该条曲线的所有曲线拟合点.

首先, 设置一个权重阈值  $\kappa$ , 如果  $Q$  中存在权重大于  $\kappa$  的点, 则选择其中权重最大的点  $A$  作为“种子点”. 以  $A$  为圆心,  $R$  为半径的圆, 即为上文提到的“生长圆”. 设圆内点的集合为  $D = \{q_k | (\|q_k - A\|) < R\}$ . 记  $D$  内权重次大的点为  $B$ . 设  $\overline{AB}$  为“区域生长”的一个“生长轴”,  $Perp(\overline{AB})$  为过  $A$  点的法向量. 半径  $R$  和角度  $\alpha$

同时控制曲线拟合结果的精度. 然后根据  $\overline{AB}$  和  $\alpha$  将圆  $(A, R)$  划分为新生长点区域 (深色), 相交曲线生长点区域 (斜线标记), 近点区域 (浅色) 三个区域, 如图 6(a) 所示. 为保证在一定局部范围内不产生环线, 且适用于多项式拟合, 本文设  $\alpha = \pi/3$ . 令  $G_1 = B$  作为该曲线的第一个“生长点”, 再根据“生长圆”规则在  $D$  内寻找该曲线的第二个“生长点”, 并对“生长圆”内不同区域内的点做相应的处理:

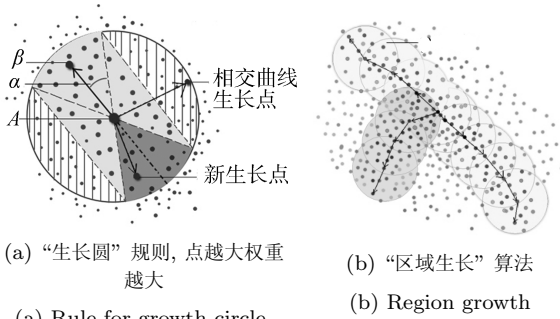


图 6 基于“生长圆”规则和“区域生长”算法的权重敏感最小二乘预处理

Fig. 6 Weight sensitive pre-processing based on rule for growth circle and region growth scheme

1) 在相交曲线生长区域内, 选择在  $Perp(\overline{AB})$  方向上投影最远且权重大于  $\kappa$  的两个远端点作为当前曲线的相交曲线的“生长点”, 即为  $G_1$  和  $G_2$ .

2) 在新生长点区域内, 选择权重最大且大于  $\kappa$  的点  $q_k$  作为当前曲线的第二个“生长点”, 记为  $G_2$ , 同时将该点和该区域内的其他点加入  $W_0$  作为曲线拟合点.

3) 将近点区域内的所有点  $q_k$  都加入  $W_0$ .

找到曲线的两个初始“生长点”  $G_1$  和  $G_2$  后, 对该条曲线执行“区域生长”算法, 如图 6(b) 所示. 每次以当前“生长点”为圆心, 选择  $R$  半径内未被加入  $W$  的点, 并以由当前“生长点”指向其前一个“生长点”的矢量为“生长轴”, 按照“生长圆”规则寻找下一个“生长点”, 直到找不到满足条件的“生长点”. 然后重新开始为新的曲线查找曲线拟合点集, 直到  $Q$  中的点都加入到  $W$ , 或者找不到合适的“种子点”.

为使增强算法具有局部协调性, 可将上述权重阈值  $\kappa$  设为一个局部适应值, 即将其设置为局部点集  $Q$  的平均权重的绝对值.

最后, 得到  $m$  组拟合潜在特征线投影的点集, 剔除点数太少的  $W_j$  后, 在  $H_i$  上用权重敏感的最小二乘方法拟合剩下的点集.

本文用二次多项式拟合潜在特征线的投影:

$$f_j = a_j + b_j \cdot x + c_j \cdot x^2 \quad (5)$$

其中,  $f_j$  表示  $NBHD(p_i)$  内第  $j$  条潜在特征线的投影曲线. 分析可知  $f_j$  上任意一点在点集曲面上的映射点为  $o_i + (x, f_j(x), g_i(x, f_j(x)))$ , 如图 7 所示.

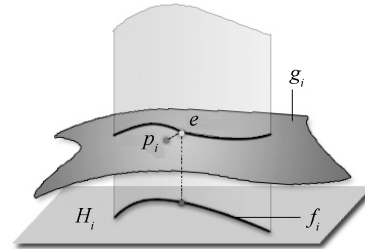


图 7 增强投影过程  
Fig. 7 The enhanced projection

潜在特征线投影曲线的数目决定  $p_i$  是特征点还是特征无关点: 如果没有拟合曲线, 则说明  $NBHD(p_i)$  内不存在潜在特征线; 如果存在一条拟合曲线, 则认为  $NBHD(p_i)$  内存在棱边. 根据  $p_i$  点对应的局部曲面拟合多项式  $g_i$ , 将特征曲线投影  $f$  映射到点集曲面上得到潜在特征线的表示, 运用一阶线性求极小值的方法<sup>[14]</sup>, 求  $p_i$  到潜在特征线上的投影, 并将投影点加入特征棱点点集  $E^{(p)}$ . 如果存在两条或者两条以上的拟合曲线则说明  $NBHD(p_i)$  内存在多条棱边. 这时, 除了将  $p_i$  点投影到离其最近的棱边上产生一个棱点  $e_i$  还, 还会产生一个角点. 计算特征曲线投影在有效范围内的交点, 根据  $g_i$  计算交点在点集曲面上的映射点, 选择离  $p_i$  最近的映射点作为角点  $c_i$  计算交点在点集曲面上的映射点, 选择离  $p_i$  最近的映射点作为角点  $c_i / \|c_i - e_i\|$ , 即角点  $c_i$  到其对应棱点  $e_i$  的距离倒数.

为简化角点的表示, 将在一定半径  $\varepsilon$  内的角点用一个角点表示<sup>[4]</sup>.  $NBHD(c_i)$  指与  $c_i$  相邻的  $\varepsilon$  半径内的角点, 选择  $NBHD(c_i)$  中最接近其平均权重的角点作为代表, 保留在  $C^{(p)}$  中, 同时将  $NBHD(c_i)$  内其他角点转移到棱点点集  $E^{(p)}$ .  $F^{(p)} = E^{(p)} + C^{(p)}$ .

#### 4 平滑谷脊特征点

由于特征增强方法本身具有鲁棒性的特点, 经过增强后的潜在特征点还不能直接生成平滑的特征线, 所以必须通过平滑处理以使其进一步向特征线靠近. 本文平滑算法基于邻域相关系数分析方法<sup>[15]</sup>和 PCA 方法, 将每个特征点投影到其邻域点主轴上, 自适应地平滑特征点.



对于任一特征点  $p_i \in F^{(p)}$ , 选取其邻域半径  $\delta$  内的点  $NBHD(p_i) = \{p_j | (||p_j - p_i||) \leq \delta, j = 1, \dots, k\}$ , 然后得到其协方差矩阵  $C$ ,  $\bar{p} = 1/k \sum_{j=1}^k p_j$  是  $NBHD(p_i)$  的重心. 计算  $C$  的特征值和特征向量, 设特征值  $\lambda_0 \geq \lambda_1 \geq \lambda_2$ , 则  $\mathbf{V}_0, \mathbf{V}_1, \mathbf{V}_2$  分别为特征值  $\lambda_0, \lambda_1, \lambda_2$  所对应的特征向量. 主轴方向即为  $\bar{p}$  和  $\mathbf{V}_0$  所定义的直线方向. 将  $p_i$  投影到主轴上, 得到的投影点  $\hat{p}_i$  即为平滑后的特征点.

特征点平滑的程度决定于  $\delta$  的大小, 可以在计算 PCA 主轴之前通过分析  $NBHD(p_i)$  在逼近平面上投影点的相关程度来控制  $\delta$  的增长, 以得到自适应的平滑效果, 如图 8 所示. 计算协方差矩阵  $C$  的特征向量和特征值, 将  $NBHD(p_i)$  内的点投影到  $\mathbf{V}_0$  和  $\mathbf{V}_1$  所决定的平面上, 并将投影点转换到局部坐标系  $(\bar{p}, \mathbf{u}, \mathbf{v}, \mathbf{V}_2)$  内,  $\bar{p}$  为坐标原点. 分析投影点在  $(\mathbf{u}, \mathbf{v})$  平面内的分布情况, 及计算投影点的两组坐标值  $\mathbf{X}$  和  $\mathbf{Y}$  的相关系数.

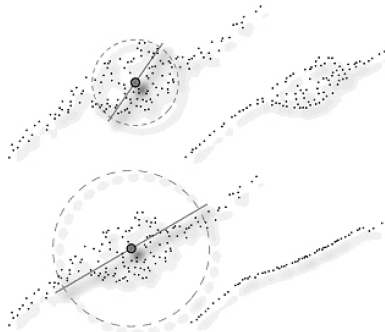


图 8 使用不同邻域半径的 PCA 平滑效果<sup>[4, 15]</sup>

Fig. 8 Smoothing results in different neighborhood sizes<sup>[4, 15]</sup>

$$\rho_{XY} = \frac{\text{cov}(\mathbf{X}, \mathbf{Y})}{\sqrt{D(\mathbf{X})}\sqrt{D(\mathbf{Y})}}$$

其中,  $\text{cov}(\mathbf{X}, \mathbf{Y}) = E\{[\mathbf{X} - E(\mathbf{X})][\mathbf{Y} - E(\mathbf{Y})]\}$  为  $\mathbf{X}$  与  $\mathbf{Y}$  的协方差;  $\sqrt{D(\mathbf{X})}$  表示均方差.  $\rho$  的大小表示  $\mathbf{X}$  和  $\mathbf{Y}$  线性相关的程度. 当  $\rho = 0$  时表示  $\mathbf{X}$  与  $\mathbf{Y}$  不相关, 当  $|\rho|$  越接近 1, 则说明  $NBHD(p_i)$  的线性相关程度越大, 也就是  $NBHD(p_i)$  的分布越接近特征线. 当给定点  $p_i$  邻域  $NBHD(p_i)$  的相关系数  $\rho$  太小时, 以一定速率扩大半径  $\delta$ , 直到  $\rho$  大于某个阈值. 本文实验使用的阈值是  $\rho \geq 0.6$ . 如果  $\delta$  扩大过程中将某个角点纳入到邻域, 则需要将包括角点在内的可能与其他特征线相关的点从邻域中剔除<sup>[4]</sup>, 当  $\delta$  扩大到一定程度仍然不能得到满意的  $\rho$

值, 则放弃该点的处理, 视其为噪声点. 得到自适应的  $\delta$  后, 计算  $p_i$  点  $\delta$  半径邻域点的主轴和  $p_i$  在主轴上的投影点, 得到  $p_i$  点对应的平滑点.

分别将  $E^{(p)}$  和  $C^{(p)}$  中平滑后的点加入平滑特征点集  $E^{(s)}$  和  $C^{(s)}$  中得到平滑后的特征点集  $F^{(s)}$ .

## 5 特征折线的生成与优化

为了得到特征线, 需要在点集  $F^{(s)}$  中选取合适的点并建立相应的连接关系.

借鉴文献 [15] 的方法, 由用户输入一个半径值  $\gamma_{\max}$  来控制特征折线的精度, 用 PCA 方法完成特征折线的生成. 如图 9 所示.



图 9 特征折线的生长<sup>[4, 15]</sup>

Fig. 9 Feature polyline propagation<sup>[4, 15]</sup>

首先将特征点集中的边点按其离最近角点的距离加入优先队列 (距离最大的边点最为优先). 然后每次从优先队列中取出一个边点  $p$ , 作为特征折线生长的“初始生长点”, 选取“初始生长点”半径  $\gamma_{\max}$  内的邻点  $NBHD(p)$  进行主元分析, 取  $NBHD(p)$  协方差矩阵的最大特征值对应的特征向量作为主轴矢量, 然后将  $NBHD(p)$  内的每个点都投影到  $p$  点和主轴矢量确定的直线上, 取投影最远的两个远端点作为新的“生长点”, 同时将  $NBHD(p)$  内的点从优先队列中删除. 接着从新的“生长点”开始继续进行下一个“生长点”的检测. 当检测不到新的“生长点”时, 证明该条特征折线“生长”完毕. 重新从优先队列中选取“初始生长点”, 开始生成新的特征折线, 直至优先队列为空.

要得到光滑的特征线, 还必须对特征折线进行优化处理, 包括特征折线端点间裂缝的修补, 特征折线扰动的滤除.

当点集模型过于稀疏或者含过多的噪声时, 特征折线间可能会出现不必要的断裂. 为修复这些裂缝, 本文算法遍历每条特征折线的端点, 并将其与在一定有效空间内的其他特征折线点连接起来, 生成新的角点, 从而将两条或者多条特征线连接成一条特征线.

特征折线的每个端点都有一个有效连接空间<sup>[4]</sup>, 该空间为由一个距离阈值  $\alpha\gamma_{\max}$ , 以及特征线端点

切向  $\mathbf{T}$  和一个预设的角度  $\mu$  定义的锥体空间.  $\mathbf{T}$  可通过计算特征线每端最后四个节点的三次多项式拟合求得, 或者直接用特征折线最后一段的方向作为端点的切向. 给定一个特征折线端点  $p$ , 其有效连接空间内存在一特征点  $q$ , 如果  $q$  是另一条特征线的端点, 且两个端点的切向方向相反, 即  $\mathbf{T}_q \cdot \mathbf{T}_p < 0$ , 则将两条特征线的端点连接起来合并成为一条特征线. 当  $q$  点是另一条特征线的中间节点, 则连接  $p$  和  $q$ , 并将  $q$  点标记为新的角点. 当  $p$  的有效连接空间内不存在特征点, 则表示  $p$  点为该特征线的一个结束点, 不需要进行连接处理.

为滤除噪声对特征折线的扰动, 使特征线更为平滑, 还需要对特征线进行松弛平滑处理. 本文采用的方法是将特征曲线的端点固定, 然后对特征折线内部的点进行 1~2 次松弛均匀化处理, 如图 10 所示.

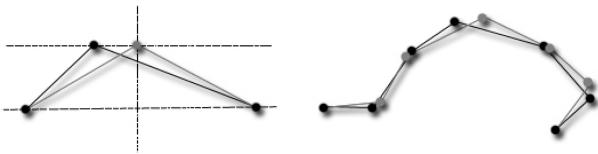


图 10 特征折线的平滑: 对邻近三点进行分析, 在满足三邻点构成的三角形面积不变的条件下, 调整中间点使其置于其他两点连线的中垂线上

Fig. 10 Smoothing feature polylines (Fix the end-points of the polylines and adjust the internal vertices with restriction that the area of any triangle enclosed by three contiguous points keeps invariant on their own defined plane.)

具体算法为: 给定一条特征线  $\mathfrak{R} = \{q_0, \dots, q_k\}$ , 首先固定  $q_0$  点和  $q_2$  点, 然后在  $q_0, q_1, q_2$  确定的平面上按照  $\Delta q_0q_1q_2$  面积不变的原则, 对  $q_1$  点的位置进行调整, 使其位于  $q_0$  和  $q_2$  的中垂线上. 接着再将  $q_1$  点和  $q_3$  点固定, 调整  $q_2$ , 如此, 直至调整完  $\mathfrak{R}$  特征线的所有中间点.

## 6 误差分析

通过多步逼近得到的特征点不一定是原点集中的点, 而是点集中的点在其邻域最小二乘拟合曲面上的映射点, 所以本文谷脊线提取算法的误差即为最小二乘曲面拟合的误差, 见图 11 所示.

$$error = \sum_{p_j \in NBHD(p_i)} ((p_j - o_i) \cdot n_i - g(u_j, v_j))^2$$

当  $error$  不为 0 时, 证明  $p_i$  点邻域的最小二乘拟合曲面片存在误差.

本文用最大误差和均方根误差作为最小二乘曲

面拟合的误差指标, 从而评价本文谷脊线提取算法的精度.  $E_j = \|(p_j - o_i) \cdot n_i - g(u_j, v_j)\|$  的最大值表示最大误差.  $p_i$  点邻域的最小二乘拟合曲面片均方根误差表示为

$$s_i = \frac{1}{n} \sqrt{\sum_{p_j \in NBHD(p_i)} (E_j - \bar{E})^2}$$

$\bar{E}$  表示  $p_i$  点邻域内所有点的平均误差. 整个点云模型的均方根误差  $s$  可记为

$$s = \frac{1}{N} \sqrt{\sum_{i=0}^N (s_i - \bar{s})^2}, \quad \bar{s} = \frac{1}{N} \sum_{i=0}^N s_i$$

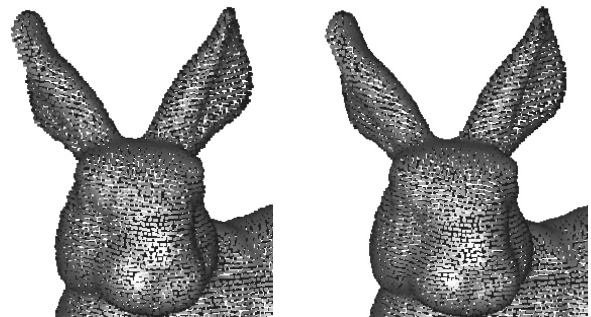


图 11 移动最小二乘法的误差和平滑滤噪效果

Fig. 11 The approximation error of MLS and its noise removal effect

本文算法的误差上限为<sup>[16]</sup>

$$\|S_P - \bigcup G_i\| \leq M_{m+1} d^{m+1}$$

$S_P$  为点云  $P = \{p_i\}$ ,  $p_i \in \mathbf{R}^3$ ,  $i \in \{1, \dots, N\}$  定义的曲面.  $G_i$  表示在局部坐标系  $(o_i, \mathbf{u}_i, \mathbf{v}_i, \mathbf{n}_i)$  内,  $p_i$  的局部最小二乘拟合曲面片,  $G_i = \{o_i + x \cdot \mathbf{u}_i + y \cdot \mathbf{v}_i + g_i(x, y) \cdot \mathbf{n}_i | (x, y) \in [-d, d]^2\}$ ,  $d$  表示点云模型的平均点距值,  $m$  表示最小二乘曲面拟合多项式的次数.  $M$  为关于  $G_i$  的  $m+1$  阶导数的常数, 即  $M \in O(\|G^{m+1}\|)$ . 本文实验使用的是二次曲面片, 即  $m = 2$ . 所以本文谷脊线提取算法的误差上限为  $\|S_P - \bigcup G_i\| \leq M_3 d^3$ .

## 7 实验结果与分析

作者在 VC++6.0 环境下实现了本文谷脊线提取算法, 硬件平台为 1.8 GHz 奔腾处理器、1 GB 内存的 PC 机.

为测试算法的抗噪性, 本文对 Bunny, Venus, Balljoint, Sper-dragon 等模型分别加入不同程度的噪声. 表 1 中记录了算法对这 4 个模型在三种不同噪声干扰下进行处理的情况. 加入噪声的方



式是使点集中的点偏离原位置任意小于  $\mu \times d$  的距离, 其中,  $d$  为平均点距值,  $\mu$  表示加入噪声的程度. 设噪声点移动方向为  $\mathbf{L}$ ,  $\mathbf{L} = (\cos \alpha, \cos \beta, \sin \alpha)$ ,  $-180 \leq \alpha \leq 180$ ,  $-90 \leq \beta \leq 90$ . 噪声点  $p'_i = p_i + (\mu \times d) \cdot \mathbf{L}$ ,  $\mathbf{L}$  为单位向量. 在实验中  $\mu$  分别取 0.0, 0.6, 1.0. 表 1 中: (a) 为提取谷脊潜在特征点的时间耗费; (b) 为增强谷脊潜在特征点的时间代价; (c) 为对特征点进行平滑所耗时间; (d) 为谷脊线的生成与优化的时间代价. 随着噪声的增加, 算法需要扩大最小二乘拟合半径以过滤掉噪声对特征表现的影响, 所以 (1) 的处理时间会随之增加, 同时整个算法的时间代价也会提高. 因为最小二乘法本身具有滤噪的功能 (见图 11), 所以算法在处理含大量噪声的模型时仍然可以保持良好的稳定性, 见图 12. 从实验数据可知本文算法抗噪性强, 算法时间代价变化不大, 且在噪声情况下仍然可以很好地提取出谷脊线.

从算法各阶段的实验数据看, (a) 阶段的时间代价比较高, 该阶段包括法向的计算、法向的调整、局部曲面拟合、曲率的计算等; (a) 中使用最小二乘方法拟合每个点的邻域时, 邻域半径越大, 拟合的点数越多, 算法的时间代价也就越高, 同时平滑掉的特征也会越多. 当拟合邻域半径太小时, 则无法反映模型局部曲率性质. 实验表明: 拟合 10~20 个邻点能很好地反映模型局部性质. 在本文实验中, 对于均匀采样的模型, 最小二乘拟合邻域半径一般取为模型平均点距值的 2~4 倍. 同时为提高计算效率, 对整个点集建立  $k$ - $d$  树结构, 从而大大减少查找时间. 而对

于曲率自适应采样的模型来说, 拟合每个点最邻近的  $k$  个点能更稳定地反映点集模型的局部微分性质. 分析实验结果可知, (b) 的时间代价与模型特征的复杂程度有关, 当模型特征越复杂, 则提取的潜在特征点越多, 导致 (b) 的时间代价增加.

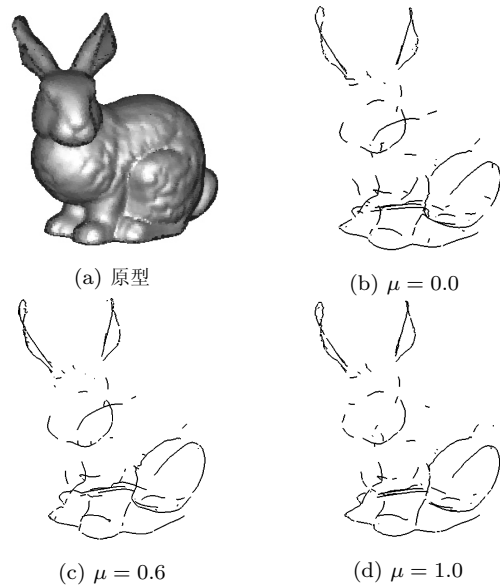


图 12 不同噪声影响下的谷脊线提取效果

Fig. 12 Valley-ridge extraction results for bunny model in different noise levels

通过调整最小二乘曲面拟合的邻域半径和提取潜在特征点时的曲率阈值, 可以得到多分辨率的特征提取效果如图 13 所示.

表 1 不同程度噪声影响下的程序运行结果 ((a) 潜在谷脊特征点的提取; (b) 特征点的增强; (c) 特征点的平滑; (d) 特征折线的生成与优化)

Table 1 Run time performances of our valley-ridge extraction implementation for models that contain different levels of noise ((a) Extracting potential valley-ridge points; (b) Enhancing feature points; (c) Smoothing feature points; (d) Propagation and completion of feature polylines)

模型	规模	平均点距值 $d$	特征点规模	噪声 ( $d$ )	曲率阈值	MLS 半径 ( $d$ )	程序运行时间 (ms)				Total
							(a)	(b)	(c)	(d)	
Bunny	35 947	0.0174793	3 156	0.0	20	1.8	7 297	687	359	438	8 781
		0.0174315	3 204	0.6	20	2.0	7 734	687	423	453	9 313
		0.0174608	3 913	1.0	17	2.6	9 876	859	672	562	12 047
Venus	134 359	0.00909488	15 974	0.0	20	2.0	30 922	3 220	2 953	2 468	39 641
		0.00905473	17 198	0.6	20	2.0	30 953	3 454	3 718	2 812	41 000
		0.00903110	11 667	1.0	20	2.8	38 984	2 422	2 406	1 547	45 453
Balljoint	137 073	0.00692489	26 911	0.0	17	4.0	64 125	6 030	6 266	4 063	80 547
		0.00686795	26 270	0.6	17	4.0	63 157	5 765	6 562	4 032	79 641
		0.00692447	26 953	1.0	17	4.0	63 577	6 032	7 187	4 157	81 046
Sper-Dragon	437 645	0.00365770	133 165	0.0	35	2.8	213 046	181 891	90 077	80 735	56 6125
		0.00374048	141 848	0.6	40	3.0	230 515	174 984	92 375	90 641	588 953
		0.00392370	123 268	1.0	38	3.5	263 750	159 375	99 641	79 344	602 484

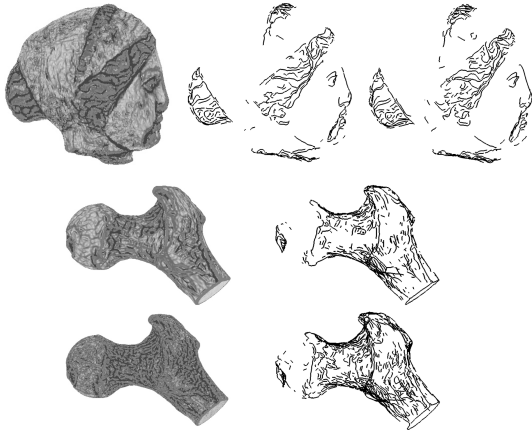


图 13 通过改变 MLS 拟合半径和曲率阈值得到不同分辨率的特征提取效果

Fig. 13 Multi-resolution result obtained from feature extraction by verifying MLS radius and biggish principle curvature threshold used for extracting potential feature points

图 14 为 Pauly 的特征提取算法<sup>[2]</sup>与本文谷脊线特征提取算法的效果对比,从实验结果可以看出本文谷脊线算法对特征更为敏感,能把稠密细小的特征准确地提取出来,如图 14 中 Venus 的发型特征、Sper-dragon 的头部特征等。

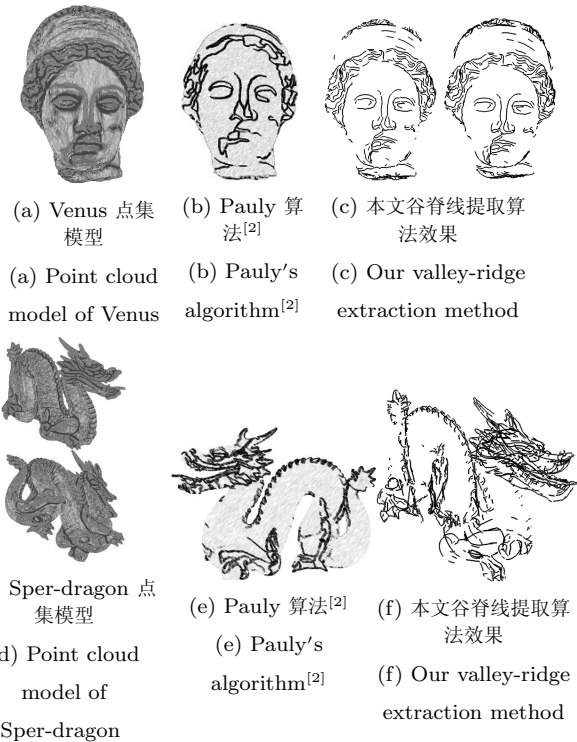


图 14 Pauly 的特征提取算法<sup>[2]</sup>与本文谷脊线提取算法的效果比较

Fig. 14 Compare the results of Pauly's feature extraction<sup>[2]</sup> and our valley-ridge extraction

## 8 结论

本文主要解决点云模型的谷脊特征提取问题,主要思想是通过计算每个点的曲率,并根据主曲率的符号将谷特征点和脊特征点分离开,然后再将特征点投影到离其最近的潜在特征线上得到增强的特征点,经过平滑特征点以及特征线的生成和优化等运算后,得到模型光滑的谷脊线.实验显示本文算法具有较强的鲁棒性,在处理特征提取较困难的稀疏点集模型时,仍能得到满意的特征提取效果,同时算法还具有局部协调性,对于较平滑细小的特征有较高的敏感性。

从实验结果看,本文算法虽然能提取出较为平滑且细小的特征,但同时也还有特征线间存在裂缝的问题.为解决这个问题,将来的一个工作是采用一种更为稳定的特征线生成算法.同时,用移动最小二乘法拟合局部曲面并计算曲面微分性质是该算法最耗时的部分,寻求一种效率更高且稳定的曲面微分性质计算方法是一个有意义的研究方向。

## References

- Demarsin K, Vanderstraeten D, Volodine T, Roose D. Detection of Closed Sharp Feature Lines in Point Clouds for Reverse Engineering Applications, Technical Report TW 458, Department of Computer Science, Katholieke University Leuven, Belgium, 2006.
- Pauly M, Keiser R, Gross M. Multi-scale feature extraction on point-sampled surfaces. *Computer Graphics Forum*, 2003, **22**(3): 281–289
- Gumhold S, Wang X, McLeod R. Feature extraction from point clouds. In: *Proceedings of the 10th International Meshing Roundtable*. Newport Beach, USA: Springer, 2001. 293–305
- Daniels I J, Ha L K, Ochotta T, Silva C T. Robust smooth feature extraction from point clouds. In: *Proceedings of the IEEE International Conference on Shape Modeling and Applications*. Lyon, France: IEEE, 2007. 123–136
- David M M, Nathan S N, Kathleen R, Kathleen R, Ruth S, Angela Y W. A practical approximation algorithm for the LMS line estimator. *Computational Statistics and Data Analysis*, 2007, **51**(5): 2461–2486
- Atkinson A C, Riani M. Regression diagnostics for binomial data from the forward search. *Statistician*, 2001, **50**(1): 63–78
- Fleishman S, Cohen O D, Silva C T. Robust moving least-squares fitting with sharp features. In: *Proceedings of the*

- International Conference on Computer Graphics and Interactive Techniques. Los Angeles, USA: ACM, 2005. 544–552
- 8 He Mei-Fang. Research on Key Technologies of Surfaces Reconstruction Based on Scattered Point Cloud Data [Ph. D. dissertation], Nanjing University of Aeronautics and Astronautics, China, 2006  
(贺美芳. 基于散乱点云数据的曲面重建关键技术研究 [博士学位论文], 南京航空航天大学, 中国, 2006)
- 9 Lai Y K, Zhou Q Y, Hu S M, Wallner J, Pottmann H. Robust feature classification and editing. *IEEE Transactions on Visualization and Computer Graphics*, 2007, **13**(1): 34–45
- 10 Ohtake Y, Belyaev A, Seidel H P. Ridge-valley lines on meshes via implicit surface fitting. *ACM Transactions on Graphics*, 2004, **23**(3): 609–612
- 11 Ohtake Y, Belyaev A, Alexa M, Turk G, Seidel H P. Multi-level partition of unity implicits. *ACM Transactions on Graphics*, 2003, **22**(3): 463–470
- 12 Alexa M, Behr J, Cohen-Or D, Fleishman S, Levin D, Silva C T. Computing and rendering point set surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 2003, **9**(1): 3–15
- 13 Wang Kui-Wu, Chen Fa-Lai, Chen Yi-Yun. Directly compute curvatures on point-based surface. *Mini-Micro Systems*, 2005, **26**(5): 813–817  
(王奎武, 陈发来, 陈意云. 基于点表示的曲面曲率计算方法. 小型微型计算机系统, 2005, **26**(5): 813–817)
- 14 Press W H, Flannery B P, Teukolsky S A, Vetterling W T. *Numerical Recipes in C: The Art of Scientific Computing (Second Edition)*. Cambridge: Cambridge University Press, 1992
- 15 Lee I K. Curve reconstruction from unorganized points. *Computer Aided Geometric Design*, 2000, **17**(2): 161–177
- 16 Levin D. The approximation power of moving least-squares. *Mathematics of Computation*, 1998, **67**(224): 1517–1531



**庞旭芳** 南京师范大学硕士研究生. 主要研究方向为计算机图形学, 数字几何处理. 本文通信作者.

E-mail: xf.pang@sub.siat.ac.cn

(**PANG Xu-Fang** Master student at Nanjing Normal University. Her research interest covers computer graphics and digital geometry processing. Corresponding author of this paper.)



**庞明勇** 南京师范大学教育技术系副教授. 2004 年获得江苏大学机械设计与理论专业博士学位. 主要研究方向为数字几何处理, 动力系统可视化及计算机辅助几何设计.

E-mail: panion@netease.com

(**PANG Ming-Yong** Associate professor in the Department of Educational Technology, Nanjing Normal University. He received his Ph.D. degree in mechanical design and theory from Jiangsu University in 2004. His research interest covers digital geometry processing, visualization of dynamic systems, and computer-aided geometric design.)



**肖春霞** 武汉大学计算机学院副教授. 2006 年获得浙江大学 CAD & CG 国家重点实验室博士学位. 主要研究方向为计算机图形学, 计算机视觉, 图像视频处理. E-mail: cxxiao@whu.edu.cn

(**XIAO Chun-Xia** Associate professor at the Computer School, Wuhan University. He received his Ph.D. degree from the State Key Laboratory of CAD & CG, Zhejiang University in 2006. His research interest covers computer graphics, computer vision, and image and video processing.)