



基于稀疏表示的 G^1 圆弧样条自动逼近方法

徐琳琳

内蒙古财经大学计算机信息管理学院, 呼和浩特 010070

E-mail: imufexll@163.com

收稿日期: 2020-11-29; 修回日期: 2021-01-27; 接受日期: 2021-03-04; 网络出版日期: 2021-08-03

内蒙古自治区高等学校科学研究项目 (批准号: NJZY21282) 资助

摘要 圆弧样条普遍用于数控机床的刀具轨迹设计中, 希望用尽可能少的圆弧段逼近曲线. 现有方法大多是启发式的, 圆弧样条曲率的分段常值性质还没有引起足够的重视. 本文基于这个内在的稀疏性质, 提出基于稀疏表示的 G^1 圆弧样条自动逼近方法, 主要分两步: 稀疏优化问题自动检测圆弧间连接点的全局初始化和重新调整连接点位置保证 G^1 连续性的局部修正. 实验结果显示新方法对对称性比较敏感, 部分尖锐特征、自交数据也可以逼近得很好, 充分说明了新方法的优越性.

关键词 圆弧样条, 稀疏表示, 逼近, G^1 连续

1 引言

CAD/CAM 系统在制造业中的应用非常广泛. 生产零件之前, CAD 系统首先对其进行可视化和相关理论分析, 然后用数控 (numerical control, NC) 程序为 CNC 加工刀具设计相应的数控路径. CNC 加工路径由圆弧和直线段构成, 其控制器广泛采用两种运动方式: 线性插值和圆弧插值. 线性插值模式是指 CAM 系统直接用线段连接数据点产生 NC 路径, 但是, 直线移动指令导致方向突变, 大量的移动指令还会增加数控程序的大小, 这种模式不仅会损坏数控机床, 零件的质量也会受到影响.

圆弧样条^[1]是指分段是圆弧、整体具有连续切线的曲线, 这种计算方法的提出可以很好地解决线性插值存在的问题. 此后, 大量圆弧样条的相关函数和数控机床的加工路径设计工作不断涌现, 根据方法性质, 可归纳为两类: 插值和逼近. 插值方法中输入数据点较少, 直接将输入的数据点作为相邻圆弧间的连接点^[2], 即以相邻数据点作为一条圆弧的端点, 结合 G^1 连续性^[3]构造圆弧, 显然逼近的圆弧段数取决于输入点的数量. 逼近则是输入大量的数据点, 通过误差驱动的启发式方法决定圆弧间的连接点^[4,5], 如何优化、确定连接点的位置减少圆弧段数一直是个有趣的课题^[6].

综合分析圆弧样条的逼近方法, 大多是启发式的处理模式, 一定程度上是对已有文章的改进, 逼近误差取得更好的数值表现, 圆弧样条曲率的内在几何性质没有得到重视. 图 1 显示了包含 3 段圆

引用格式: 徐琳琳. 基于稀疏表示的 G^1 圆弧样条自动逼近方法. 中国科学: 信息科学, 2021, 51: 1255–1269, doi: 10.1360/SSI-2020-0360
Xu L L. Automatic G^1 arc spline approximation via sparse representation (in Chinese). Sci Sin Inform, 2021, 51: 1255–1269, doi: 10.1360/SSI-2020-0360

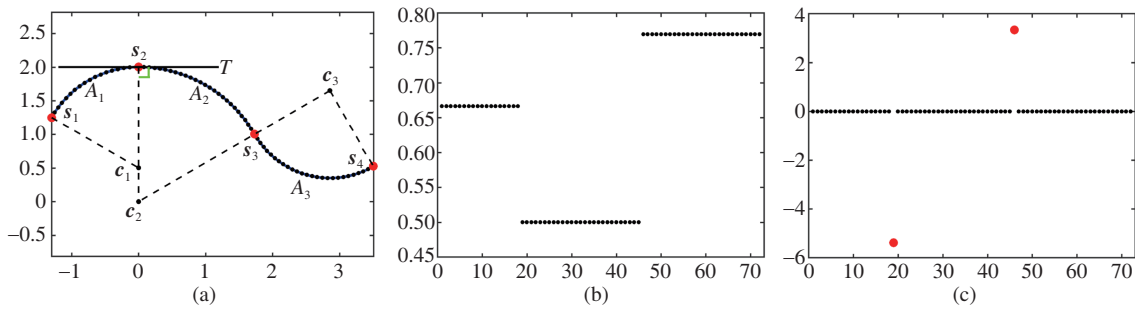


图 1 (网络版彩图) 圆弧样条中的稀疏性. (a) G^1 连续圆弧样条 (曲线) 及采样点 (黑色点); (b) 采样点曲率图; (c) 采样点曲率的拟梯度, 只有连接点处是非零的

Figure 1 (Color online) The sparsity in arc spline. (a) G^1 continuous arc spline (curve) and sampling points (black points); (b) curvatures of sampling points; (c) the quasi gradients of the sampling points, where the nonzero values correspond to joint points

弧 $A = \{A_j\}_{j=1}^3$ 的 G^1 连续的圆弧样条, 黑色点表示曲线上的采样点, 红色点 $S = \{s_i\}_{i=1}^4$ 表示相邻圆弧间的连接点, c_j 为圆弧 A_j 的圆心, T 为圆弧 A_1 和 A_2 在连接点 s_2 处的公切线. 从图 1(b) 可以看出, 同一段圆弧上采样点的曲率值相等, 3 段圆弧对应 3 个不同的曲率值. 基于稀疏表示的基本知识, 考虑相邻采样点的曲率 (图 1(b)) 变化 (图 1(c)), 本文称为拟梯度 (3.3 小节), 除了两个连接点外其他拟梯度值都是零, 这种稀疏性质可以用 l_0 范数^[7] 建模得到相应的稀疏表示^[8] 问题.

根据这个发现, 利用稀疏表示, 本文提出新的 G^1 连续圆弧样条逼近方法实现了相邻圆弧间连接点的自动检测, 这也是新方法的主要贡献. 针对稀疏优化和多个变量同时求解的问题, 本方法还设计了高计算率的两阶段优化算法: 稀疏优化问题自动检测圆弧间连接点的全局初始化和重新调整连接点位置保证 G^1 连续性的局部修正. 实验结果充分说明了稀疏表示在自动检测连接点以及最小化圆弧段数上的优越性.

2 相关工作

2.1 圆弧样条

为了减少数控机床刀具路径编程的指令和刀具变化次数, 研究人员针对不同类型的圆弧样条曲线, 以及要逼近的曲线性质提出了各种圆弧样条的逼近方法.

有些工作利用特殊的双圆弧曲线, 介绍如何插值给定数据点, 或者如何选择位于给定切线一侧的双圆弧^[9~13]. 对于增强的 G^1 连续 Hermite 数据的插值问题, Meek 等^[14] 详细介绍了双圆弧样条的单参数族.

有些样条函数具有特殊的几何性质, Ahn 等^[15] 利用二次 Bézier 曲线的曲率单调性构造连续的圆弧样条, 误差容限驱动圆弧的段数. Marciniak 等^[16] 为拟合螺旋曲线^[17] 建立了非线性方程, 得到最少分段的圆弧样条. Meek 等^[11] 提出分治策略, 用圆弧样条拟合二次 NURBS 曲线, 并研究了离散回旋曲线^[17] 进行公路路线的相关设计. Ong 等^[12] 通过求解非线性问题最小化 B 样条曲线与圆弧样条之间的边界面积, 进而构造最优圆弧样条.

第 1 节简单介绍了圆弧样条的插值和逼近方法. 针对插值问题, 除了输入点本身, 一般要求包含其他信息, 比如圆弧样条的切向数据^[18]. Chen 等^[2] 直接以输入点作为连接点, 根据连续性要求进行插值计算. Kim 等^[19] 利用双圆弧曲线, 通过优化内能最小化模型插值输入点序列, 同时保证 G^1 连

续性. 对于逼近问题, 除了启发式方法 [4,5], 基于上述插值方法, 很多研究人员希望用一些广泛使用的 NURBS 等参数曲线 [20~22] 逼近输入数据, 确定连接点的位置, 并获取相应的切线信息. 为了能够在理论上分析逼近结果, Maier 等 [6,23] 介绍了连接点的自动选择方法, 圆弧样条曲率的稀疏性质还是没有利用起来.

2.2 稀疏表示

20 多年来, 为了更简洁地表示信号, 稀疏表示技术 [8,24] 成为信号处理界非常引人注目且发展成熟的研究领域, 其目的是在给定的过完备字典中选择尽可能少的原子表示信号, 方便进一步对信号进行加工处理. 稀疏表示技术在过去十年间被迅速地应用到机器学习 [25,26]、信号处理 [27,28]、神经系统科学 [29,30] 以及统计学 [31~33] 等多个领域, 对于图像、音视频等具有稀疏性质的二维信号, 是非常有效的处理工具.

鉴于稀疏表示方法的明显优势, 很多研究人员深入挖掘三维几何中的稀疏表现, 成功进行了二维向三维的推广 [34]. Digne 等 [35] 采用 K-SVD 算法准确地完成了点云的压缩问题. Avron 等 [36] 针对点云的法向计算提出 ℓ_1 优化模型. He 等 [37] 从图像平滑工作出发, 利用 ℓ_0 优化实现了三维网格的去噪. Wang 等 [38] 基于噪声和特征分离的创新方法, 通过求解加权 ℓ_1 的压缩感知模型完成网格去噪. 上述工作专注于几何应用, Xu 等 [39] 在稀疏表示中引入参数域优化, 用复合稀疏表示技术实现几何逼近.

3 预备知识

为了更准确地理解新方法, 首先介绍圆弧样条的概念, 并简单讨论曲率的一些基础知识.

3.1 圆弧样条

G^1 连续的圆弧样条曲线是指分段是圆弧、在连接点处满足 G^1 连续性的曲线, 如图 1(a) 所示. 每段圆弧 A_j 对应一个圆心 c_j 、一个半径 r_j 和两个连接点 s_j, s_{j+1} .

G^1 连续. 本文方法中, 根据圆的定义和几何性质定义 G^1 连续. 以 A_1, A_2 两段圆弧为例, 利用连接点 s_2 处切线的垂直关系, G^1 连续要求两段圆弧有共同的切线 T , 即 c_1, c_2 和 s_2 必须共线, 令 $\rho(s_2, c_1)$ 为过 s_2 和 c_1 直线段的斜率, s_2 处的 G^1 连续性可表示为 $\rho(s_2, c_1) = \rho(s_2, c_2)$, 一般表达式为

$$\rho(s_l, c_{l-1}) = \rho(s_l, c_l).$$

内在稀疏性. 现在给出第 1 节中提到的圆弧样条的内在稀疏性.

图 1(b) 显示了圆弧样条 (图 1(a)) 上采样点 $P = \{p_i(x_i, y_i)\}_{i=1}^n$ 的曲率 κ^P , 其拟梯度 (图 1(c)) $\nabla \kappa^P$ 只在 s_2, s_3 两个连接点处非零, 即 $\|\nabla \kappa^P\|_0 = 2$, 其中 $\|\mathbf{x}\|_0$ 表示向量 \mathbf{x} 中非零元素个数, 表示向量的稀疏性. $\|\nabla \kappa^P\|_0 = 2$ 说明曲率的拟梯度具有稀疏性质, 且在连接点处得到体现. 区别于启发式工作, 稀疏表示将是一个强大的检测连接点工具.

3.2 优化问题

图 2 对本文方法的优化问题给出清晰的解释. 输入有序点集 $P = \{p_i(x_i, y_i)\}_{i=1}^n$ (图 2(a)), 新方法希望通过稀疏表示技术自动地找到最优连接点的位置 $S = \{s_i\}_{i=1}^{m+1}$, 并产生尽可能少的圆弧段 $\{A_j\}_{j=1}^m$ 来逼近 P (图 2(b)), 即确定圆心 $C = \{c_j\}_{j=1}^m$ 和半径 $R = \{r_j\}_{j=1}^m$.

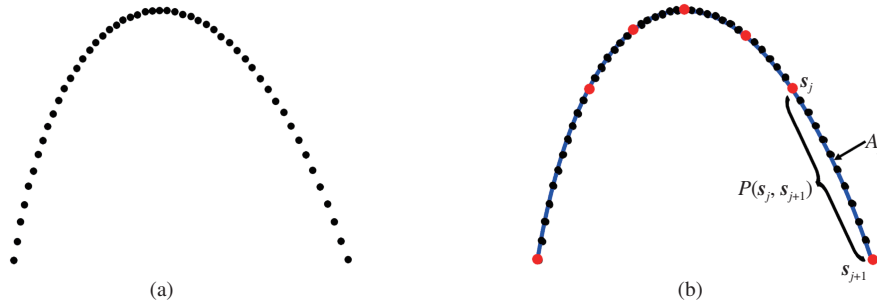


图 2 (网络版彩图) 基于稀疏表示的圆弧样条逼近问题. 输入有序点集 (a), 希望用尽可能少的圆弧 (b) 逼近 (a). A_j 为第 j 段圆弧, 对应两个连接点 s_j 和 s_{j+1} . $P(s_j, s_{j+1})$ 表示输入点集 P 中位于连接点 s_j 和 s_{j+1} 之间的点

Figure 2 (Color online) Arc spline approximation based on sparse representation. Given an ordered data set (a), the optimization aims for as few arcs (b) as possible to approximate (a). A_j is the j -th arc corresponding to two joint points s_j and s_{j+1} . $P(s_j, s_{j+1})$ indicates the data points in P between s_j and s_{j+1}

这里, m 是最终圆弧的数量, $m+1$ 个连接点包含 P 的起点和终点. 令 $P(s_j, s_{j+1})$ 是点集 P 位于连接点 s_j 和 s_{j+1} 之间的子集, 每一段圆弧 $A_j(s_j, s_{j+1})$ 的圆心 c_j 和半径 r_j 由 $P(s_j, s_{j+1})$ 决定.

3.3 曲率计算

对于平面参数曲线 $p(t) = (x(t), y(t))$, 曲率公式为

$$\kappa(t) = \frac{x'(t)y''(t) - y'(t)x''(t)}{(x'(t)^2 + y'(t)^2)^{\frac{3}{2}}}.$$

假设某个采样点为 $\mathbf{p}_i = (x_i, y_i)$, 其中 $x_i = x(t_i)$, $y_i = y(t_i)$, 令 $h = t_{i+1} - t_i$, 一阶导数、二阶导数分别用一阶差商、中心差商近似, 代入公式推导得到 \mathbf{p}_i 点处曲率的离散表达形式:

$$\kappa(\mathbf{p}_i) = \frac{(x_{i+1} - x_i)(y_{i+1} - 2y_i + y_{i-1}) - (y_{i+1} - y_i)(x_{i+1} - 2x_i + x_{i-1})}{((x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2)^{\frac{3}{2}}} = \mathcal{T}_i \mathcal{P}, \quad (1)$$

其中 $\mathcal{P} = (x_1, \dots, x_n, y_1, \dots, y_n)$ 是输入点集坐标构成的一维向量, \mathbf{p}_i 点处的曲率记为 κ_i^P , 与变量 t 无关, \mathcal{T}_i (计算过程见附录 A) 是一个行向量, 所有这些行向量构成了点集 P 到其曲率的变换矩阵 \mathcal{T} . 定义 $\nabla \kappa^P = \{\kappa_{i+1}^P - \kappa_i^P\}_{i=1}^{n-1}$, 用于表示点集 P 内每一点处曲率的变化情况, 图 1(c) 说明非零元素与相邻圆弧间连接点的对应关系, 本文中称 ∇ 为拟梯度算子. 对于一个密集的有序点集, 不妨假设它是均匀采样的, 方便曲率的近似. 第 5 节的实验结果和附录图 A1 都证明了两个关键变换矩阵的有效性.

4 最优稀疏圆弧样条逼近

现在介绍可以自动检测连接点的 G^1 连续圆弧样条逼近方法. 首先给出基于稀疏表示的逼近方法的定义, 然后详细讨论高效的优化算法.

4.1 公式

第 3.2 小节指出, 连接点 S , 圆心 C 和半径 R 都是优化目标. 根据半径 r 和曲率 κ 之间的倒数关系: $r = \frac{1}{\kappa}$, 半径可与曲率一致地反应圆弧样条中的稀疏性, 图 3 利用图 1(a) 所示圆弧样条进行了解释. 图 3(a) 是采样点处的曲率半径, 跟曲率图的表现 (图 1(b)) 一致, 都是分段常值, 相应地, 曲率半径

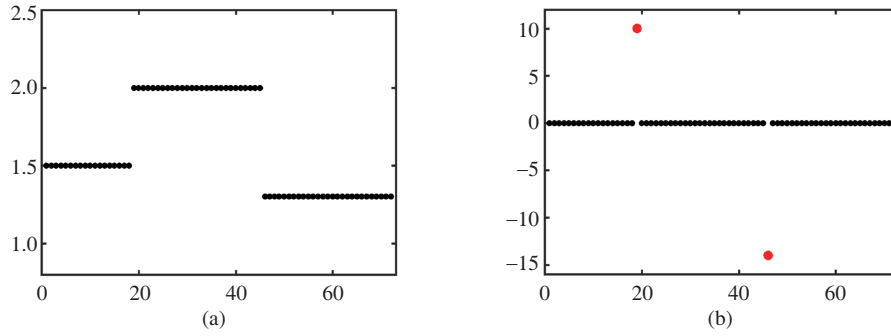


图 3 (网络版彩图) 图 1(a) 中圆弧样条在采样点处曲率半径的稀疏性. (a) 采样点的曲率半径; (b) 曲率半径的拟梯度

Figure 3 (Color online) The sparsity of curvature radiuses at sampling points on the arc spline in Figure 1(a). (a) Curvature radiuses of sampling points; (b) quasi gradients of curvature radiuses

的拟梯度值 (图 3(b)) 大部分是零. 假设 $\bar{R} = \{\bar{r}_i\}_{i=1}^n$ 为输入点集 $P = \{\mathbf{p}_i\}_{i=1}^n$ 的曲率半径, 和 $\nabla\kappa$ 的定义一样, 令 $\nabla\bar{R} = \{\bar{r}_{i+1} - \bar{r}_i\}_{i=1}^{n-1}$, 同样可以确定圆弧样条连接点的位置. 为了最小化圆弧段数, 本文采用 ℓ_0 范数最小化技术计算 $\nabla\bar{R}$ 的非零个数:

$$\min_{\{S, C, \bar{R}\}} \|\nabla\bar{R}\|_0, \tag{2a}$$

$$\text{s.t. } \sum_{j=1}^m \sum_{\mathbf{p}_i \in A_j} (\|\mathbf{p}_i - \mathbf{c}_j\|_2 - r_j)^2 < n\varepsilon, \tag{2b}$$

$$\rho(\mathbf{s}_l, \mathbf{c}_{l-1}) = \rho(\mathbf{s}_l, \mathbf{c}_l), \tag{2c}$$

$$\|\mathbf{s}_l - \mathbf{c}_{l-1}\|_2 = r_{l-1}, \quad l = 2, \dots, m+1, \tag{2d}$$

$$\|\mathbf{s}_l - \mathbf{c}_l\|_2 = r_l, \quad l = 1, \dots, m,$$

其中式 (2b) 是逼近误差的数据项. 式 (2d) 通过点到圆心距离的硬约束保证连接点处的连续性, 与式 (2c) 共同保证圆弧样条的 G^1 连续性. 根据 3.2 小节对圆弧 $\{A_j\}_{j=1}^m$ 的介绍, 每一段最优圆弧 A_j 将对应 \bar{R} 的一个子集, 子集内的所有元素值等于同一个值 r_j .

问题 (2) 中, ℓ_0 优化的非凸计数问题已经很难找到最优结果, 再同时优化圆弧数量 m 、连接点 S 、圆心 C 和曲率半径 \bar{R} , 问题的求解难度大大增加. 为了有效地解决这一问题, 将其分为两个子问题: 基于稀疏表示的全局样条逼近 (4.2 小节) 和进一步的局部修正 (4.3 小节).

4.2 稀疏圆弧样条逼近

到目前为止, 点集 P 和问题 (2) 中的未知量之间仅有的关系是曲率计算公式 (1). 所以, 可以通过优化下面的 ℓ_1 范数问题, 对圆弧样条进行初始化:

$$\min_Q \|DQ\|_1, \tag{3a}$$

$$\text{s.t. } \|Q - P\|_2 \leq \varepsilon. \tag{3b}$$

其中, $Q = \{\mathbf{q}_i\}_{i=1}^n$, $DQ = \nabla\kappa^Q$, D 是从输入数据点到其曲率拟梯度 $\nabla\kappa$ 的变换矩阵 (式 (A2)). 约束条件 (3b) 说明该问题是全局的逼近方法. 这里, ℓ_1 凸优化问题代替问题 (2) 中的 ℓ_0 范数, 可利

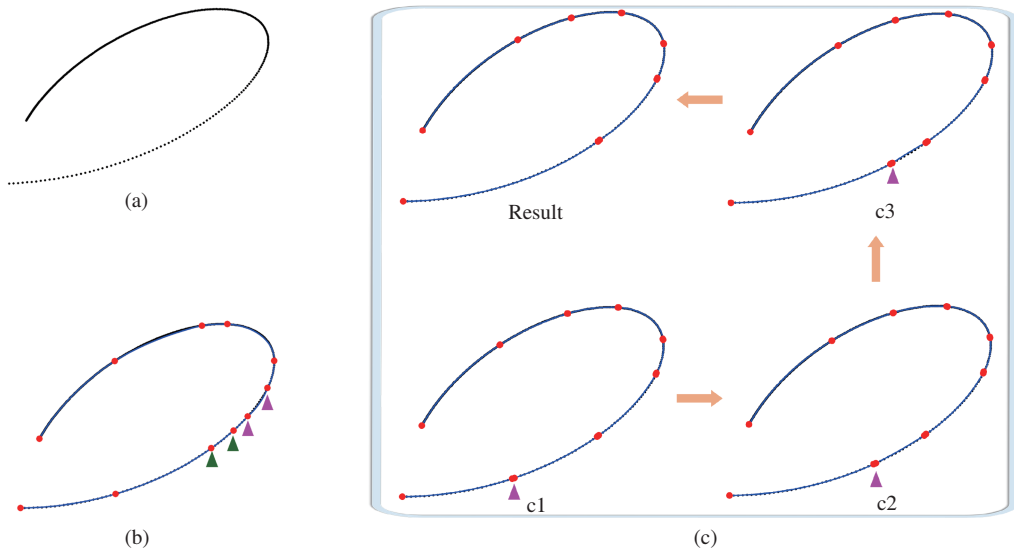


图 4 (网络版彩图) 基于稀疏表示的 G^1 连续圆弧样条自动逼近方法流程图. 输入数据集 (a), 通过求解全局 ℓ_1 稀疏优化问题 (3) 对其自动分段, 初始化圆弧样条 (b), 结合连接点的局部修正 (c) 产生满足 G^1 连续的逼近结果

Figure 4 (Color online) The flow chart of automatic G^1 continuous arc spline approximation based on sparse representation. Given a set of data points (a), the global ℓ_1 sparse optimization problem (3) automatically segments (a) to initialize the arc spline (b), coupled with the local modification of joint points (c), the G^1 continuous arc spline is obtained

用 Matlab 中的 cvx 工具箱求解, 本文所有实验采用默认求解器 sdpt3, 与 mosek 求解器的运行时间整体一致.

图 4 以叶形线 (leaf line)

$$\begin{cases} x(t) = \frac{3at}{1+t^3}, \\ y(t) = \frac{3at^2}{1+t^3}, \end{cases} \quad 0 \leq t \leq 5, \quad a = 10$$

为例展示了新方法的优化过程. 本文所有图中, 红色点表示圆弧间的连接点, 蓝色曲线表示优化得到的 G^1 连续圆弧样条, 黑色点表示输入点集. 圆弧段数 m 会在局部修正过程中发生变化, 例如, 图 4(b) 中两个紫色、绿色三角形标注的连接点, 经过局部修正算法的第一次迭代更新, 分别合并成一个连接点, 如图 4(c) 的 c1 所示, 圆弧数第一次减少. 接下来的优化过程中, 图 4(c) 的 c1 中紫色三角形标注的连接点逐渐移动到邻点上, 两点间的圆弧消失, 圆弧数再次减少, 即 m 减小, 算法收敛时得到最优的圆弧样条, 如图 4(c) 的 Result 所示.

通过求解问题 (3), 将得到一个逼近输入点集 P 的点序列 Q , 满足逼近误差 ε , 如图 4(b) 和 5(a) 所示. 重要的是, 曲率 $\kappa^Q = \mathcal{T}Q$ (图 5(b)) 基本满足分段常值性质, 拟梯度向量 $\nabla \kappa^Q = \mathcal{D}Q$ (图 5(c)) 中的大部分元素为零. 但是, 与图 1 中曲率 (图 1(b))、拟梯度 (图 1(c)) 的理论值对比发现, 相邻圆弧间, κ^Q 存在少量散乱点, $\nabla \kappa^Q$ 可能不止一个非零元素 (图 5(c)). 对于这种情况, 本文直接将散乱点分组到任意相邻的圆弧上, 进而计算出初始曲率, 这种处理方法对逼近结果没有任何影响. 最终, 每段圆弧 A_j 上, 半径 r_j 根据 $r = \frac{1}{\kappa}$ 计算得到, 两个端点 (s_j, s_{j+1}) 根据曲率拟梯度的非零值确定, 求出圆心 c_j 便可确定初始圆弧样条 (图 4(a)).

需要注意的是, 优化问题 (3) 得到的初始化样条满足曲率的稀疏性质, 但不能保证相邻圆弧间

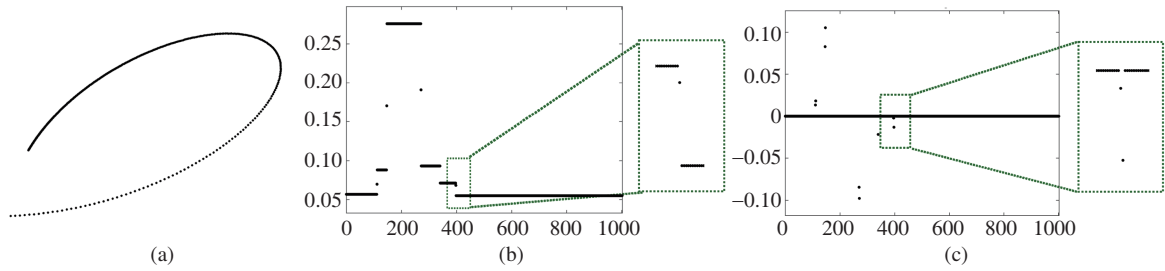


图 5 (网络版彩图) ℓ_1 稀疏优化问题 (3) 的补充说明. 输入叶形线上的采样点 (图 4(a)), 求解问题 (3) 得到逼近点集 (a). (b) 和 (c) 是分别利用变换矩阵 \mathcal{T} , \mathcal{D} 计算得到的逼近点集的曲率及其拟梯度. 与图 1 理论情况对比发现, 变换矩阵的计算结果中会有少量冗余点, 优化过程中将这些冗余点分组到任意相邻的圆弧中

Figure 5 (Color online) Supplementary explanation of ℓ_1 sparse optimization problem. Input the sampling points of leaf curve (Figure 4(a)), (a) is the approximation point set by solving problem (3). Its curvature (b) and quasi gradient (c) are computed respectively with transformations \mathcal{T} and \mathcal{D} . Compared with the theoretical performance in Figure 1, there are a few redundant points which will be grouped to any neighboring arc

是 G^1 连续的, 需进一步优化连接点位置, 以便更好地满足逼近误差 (方程 (2b)) 和 G^1 连续要求 (4.3 小节).

4.3 局部修正

本小节将局部调整连接点的位置, 即重新调整每段圆弧应该对应的点集 P 的子集, 继续优化已经得到的圆弧样条, 与约束条件 (2c) 和 (2d) 相同, 这个局部修正定义为

$$\begin{aligned} \min_{\{S, C, R\}} & \sum_{j=1}^m \sum_{\mathbf{p}_i \in A_j} (\|\mathbf{p}_i - \mathbf{c}_j\|_2 - r_j)^2, \\ \text{s.t.} & \rho(\mathbf{s}_l, \mathbf{c}_{l-1}) = \rho(\mathbf{s}_l, \mathbf{c}_l), \\ & \|\mathbf{s}_l - \mathbf{c}_{l-1}\|_2 = r_{l-1}, \quad l = 2, \dots, m+1, \\ & \|\mathbf{s}_l - \mathbf{c}_l\|_2 = r_l, \quad l = 1, \dots, m. \end{aligned} \tag{4}$$

类似多变量问题的一般处理方法, 用 Gauss-Seidel 方式迭代更新 $S = \{\mathbf{s}_l\}_{l=1}^{m+1}$ 和 $C = \{\mathbf{c}_j\}_{j=1}^m$, $R = \{r_j\}_{j=1}^m$.

(1) S -子问题. 仔细观察问题 (4), 目标函数实际上跟连接点 S 无关, 所以, 连接点 S 是下面问题的可行解:

$$\begin{aligned} \min_S & \sum_{l=1}^m (\|\mathbf{s}_l - \mathbf{c}_l\|_2 - r_l)^2 + \sum_{l=2}^{m+1} (\|\mathbf{s}_l - \mathbf{c}_{l-1}\|_2 - r_{l-1})^2, \\ \text{s.t.} & \rho(\mathbf{u}_{i_l}, \mathbf{c}_{l-1}) = \rho(\mathbf{u}_{i_l}, \mathbf{c}_l), \\ & \|\mathbf{s}_l - \mathbf{c}_{l-1}\|_2 = r_{l-1}, \quad l = 2, \dots, m+1, \\ & \|\mathbf{s}_l - \mathbf{c}_l\|_2 = r_l, \quad l = 1, \dots, m, \\ & \|\mathbf{s}_l - \mathbf{p}_{i_l}\| \leq \delta, \end{aligned}$$

其中, δ 可用连接点所属分段中采样点间的最大距离或包围盒的 t ($0 < t < 1$) 规模确定, 防止连接点因 G^1 连续的要求移动距离过大.

(2) $\{C, R\}$ -子问题. 如上所述, 每一对 $\{\mathbf{c}_j, r_j\}$ 由子集 $P(\mathbf{s}_j, \mathbf{s}_{j+1})$ 局部确定, 所以, 这里只需要输入数据 P 的分段分割信息. 令 i_l 为距离点 \mathbf{s}_l 最近点 \mathbf{p}_{i_l} 的下标, 指标集 $\{i_l\}_{l=1}^m$ 将 P 分成 m 个子集,

分别对应已经得到的圆弧, 其中 $i_l = 1, i_{m+1} = n$, 第 j 个子集 $P(\mathbf{s}_j, \mathbf{s}_{j+1})$ 也可以表示为 $P(i_j, i_{j+1})$. 利用新的指标记号, 问题 (4) 转化为

$$\begin{aligned} \min_{\{C, R\}} & \sum_{j=1}^m \sum_{i \in (i_l, i_{l+1})} (\|\mathbf{p}_i - \mathbf{c}_j\|_2 - r_j)^2, \\ \text{s.t.} & \rho(\mathbf{p}_{i_l}, \mathbf{c}_{l-1}) = \rho(\mathbf{p}_{i_l}, \mathbf{c}_l), \\ & \|\mathbf{p}_{i_l} - \mathbf{c}_{l-1}\|_2 = r_{l-1}, \quad l = 2, \dots, m+1, \\ & \|\mathbf{p}_{i_l} - \mathbf{c}_l\|_2 = r_l, \quad l = 1, \dots, m. \end{aligned}$$

但是, 这里的约束条件直接将输入点作为连接点, 对于逼近问题太强了, 可能不会产生好的结果. 为了放松这些插值约束, 引入辅助变量 $U = \{\mathbf{u}_i\}_{i=1}^n$ 得到最终的局部修正优化问题:

$$\begin{aligned} \min_{\{U, C, R\}} & \sum_{j=1}^m \sum_{i \in (i_l, i_{l+1})} (\|\mathbf{u}_i - \mathbf{c}_j\|_2 - r_j)^2, \\ \text{s.t.} & \|U - P\|_2 \leq n\tau, \\ & \rho(\mathbf{u}_{i_l}, \mathbf{c}_{l-1}) = \rho(\mathbf{u}_{i_l}, \mathbf{c}_l), \\ & \|\mathbf{u}_{i_l} - \mathbf{c}_{l-1}\|_2 = r_{l-1}, \quad l = 2, \dots, m+1, \\ & \|\mathbf{u}_{i_l} - \mathbf{c}_l\|_2 = r_l, \quad l = 1, \dots, m. \end{aligned} \quad (5)$$

与问题 (4) 相似, 需要另外一个迭代算法更新 U 和 $\{C, R\}$.

(a) U -更新:

$$\begin{aligned} \min_U & \sum_{j=1}^m \sum_{i \in (i_l, i_{l+1})} (\|\mathbf{u}_i - \mathbf{c}_j\|_2 - r_j)^2, \\ \text{s.t.} & \|U - P\|_2 \leq n\tau, \\ & \rho(\mathbf{u}_{i_l}, \mathbf{c}_{l-1}) = \rho(\mathbf{u}_{i_l}, \mathbf{c}_l), \\ & \|\mathbf{u}_{i_l} - \mathbf{c}_{l-1}\|_2 = r_{l-1}, \quad l = 2, \dots, m+1, \\ & \|\mathbf{u}_{i_l} - \mathbf{c}_l\|_2 = r_l, \quad l = 1, \dots, m. \end{aligned}$$

(b) $\{C, R\}$ -更新:

$$\begin{aligned} \min_{\{C, R\}} & \sum_{j=1}^m \sum_{i \in (i_l, i_{l+1})} (\|\mathbf{u}_i - \mathbf{c}_j\|_2 - r_j)^2, \\ \text{s.t.} & \rho(\mathbf{u}_{i_l}, \mathbf{c}_{l-1}) = \rho(\mathbf{u}_{i_l}, \mathbf{c}_l), \\ & \|\mathbf{u}_{i_l} - \mathbf{c}_{l-1}\|_2 = r_{l-1}, \quad l = 2, \dots, m+1, \\ & \|\mathbf{u}_{i_l} - \mathbf{c}_l\|_2 = r_l, \quad l = 1, \dots, m. \end{aligned}$$

使用 Matlab 中求解带约束非线性多元函数最小值的 `fmincon` 函数, 给定 $\tau = 10^{-5}$, 经过 3 次迭代, 两个更新问题的算法可以很快收敛.

4.4 算法

算法 1 将 4.2 和 4.3 两小节的优化过程进行了总结.

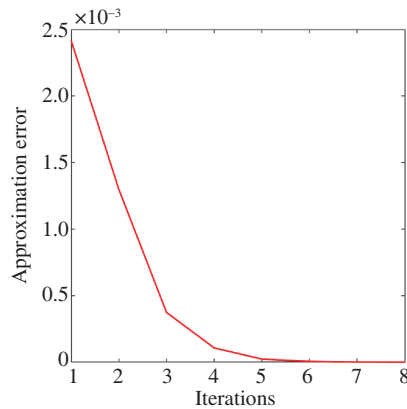


图 6 (网络版彩图) Leaf line (图 4) 优化过程中的误差变化趋势
Figure 6 (Color online) The trend of the optimization error of the leaf line (Figure 4)

算法 1 Sparse arc spline approximation

Input: Sampling points P of curves;

- 1: Step 1. Global sparse arc spline approximation;
- 2: Read the sampling points P and calculate $C, R, \{i_l\}_{l=1}^{m+1}$ by (3);
- 3: Step 2. Local modification;
- 4: Modify joint points S and $\{C, R\}$ using Gauss-Seidel method;
- 5: Step 2.1. Fix $\{C, R\}$ and solve S -subproblem;
- 6: Update $\{i_l\}_{l=1}^{m+1}$ by finding the subscript of the closest point according to the location of the joint point;
- 7: Step 2.2. Fix S and introduce auxiliary variable U ; U and $\{C, R\}$ are solved after three iterations;
- 8: Update the center and radius of each arc according to the new joint points.
- 9: After Steps 2.1 and 2.2 iterate about 8 times, the algorithm converges and outputs;

Output: Centers C , radiuses R , subscripts $\{i_l\}_{l=1}^{m+1}$ of joint points, approximation error e .

图 6 显示了 leaf line (图 4) 的优化过程中误差变化情况. 显然, 逼近误差呈下降趋势, 且从图中可以看出, 本算法 Step 2.1, Step 2.2 循环迭代 8 次左右误差趋于稳定, 说明了算法的收敛性. 下文中的所有实验结果均附能量变化子图进行收敛性说明.

5 实验结果

本文所有实验是在双核 2.6 GHz, 8 G 内存计算机上 Matlab R2018a 环境下实现的.

为了说明本文方法的优势, 首先与最相关的二分法^[5]工作中的两条 Bézier 曲线进行比较, 这个工作已经比 Ahn 等^[15]的方法优化出更少的圆弧, 逼近能力也和文献 [20] 中的工作基本一样. 二次 Bézier 曲线定义如下:

$$\mathbf{q}(t) = (1-t)^2 \mathbf{b}_0 + 2t(1-t) \mathbf{b}_1 + t^2 \mathbf{b}_2, \quad 0 \leq t \leq 1,$$

其中 $\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2$ 为控制点. 两条 Bézier 曲线的控制点分别为 Bézier curve 1:

$$\{(1, 1), (5, 1), (1, 2.75)\}.$$

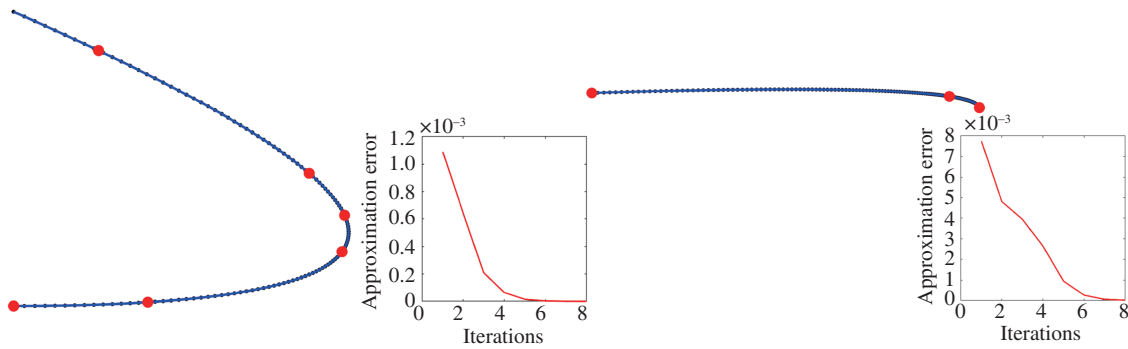


图 7 (网络版彩图) 本文方法对二分法^[5]中 Bézier 曲线的逼近结果

Figure 7 (Color online) The approximation results of Bézier curves in bisection algorithm^[5] using our method. (a) Bézier curve 1; (b) Bézier curve 2

表 1 本文方法与二分法^[5]的逼近数据比较

Table 1 The approximation data comparison of our method and the bisection algorithm^[5]

Bézier curves	N_{Arc}		e	
	Our method	Bisection algorithm ^[5]	Our method	Bisection algorithm ^[5]
Bézier curve 1	5	10	5.0159×10^{-7}	1.0000×10^{-3}
Bézier curve 2	2	14	6.4544×10^{-5}	9.6696×10^{-5}

Bézier curve 2:

$$\{(0.54, 3.40), (7.22, 3.61), (7.39, 3.14)\}.$$

在两条曲线上采样 $N_p = 1000$ 个点作为输入点集 P .

图 7 显示了圆弧样条的逼近结果. 表 1 列出二分法^[5]和本文方法的圆弧段数 N_{Arc} 与逼近误差 e , 结合连接点的自动检测能力, 充分说明了新方法较启发式方法的明显优势.

为了说明算法的逼近能力, 本文对多种不同类型的参数曲线进行了测试, 如图 8 所示, 各曲线表示如下:

(1) B-spline 1 (图 8(a)):

$$\text{控制点: } \{(0, 0), (1, 5), (4, 5), (6, 1), (9, -3), (11, -4), (12, 5), (14, 6), (17, -2), (20, -2), (21, 1)\}.$$

(2) B-spline 2 (图 8(b)):

$$\text{控制点: } \{(0, 10), (1, 8), (4, 7.5), (5, 10), (6, 10), (7, 8.5), (8, 8.3), (12, 11), (12, -1), (6, 4.3), (4, 10), (2, 8), (1, 7.5), (0, 0), (3, 0), (5, 2.7), (8, 7), (10, 7), (11, 3), (7.3, -1), (9.3, -2)\}.$$

(3) Polynomial (图 8(c)):

$$\begin{cases} x(t) = t, \\ y(t) = t^3, \end{cases} \quad -2 \leq t \leq 2.$$

(4) Plum line (图 8(d)):

$$\begin{cases} x(t) = (10 - 9 \sin^2 2.5t) \cos t, \\ y(t) = (10 - 9 \sin^2 2.5t) \sin t, \end{cases} \quad 0 \leq t \leq 1.5\pi.$$

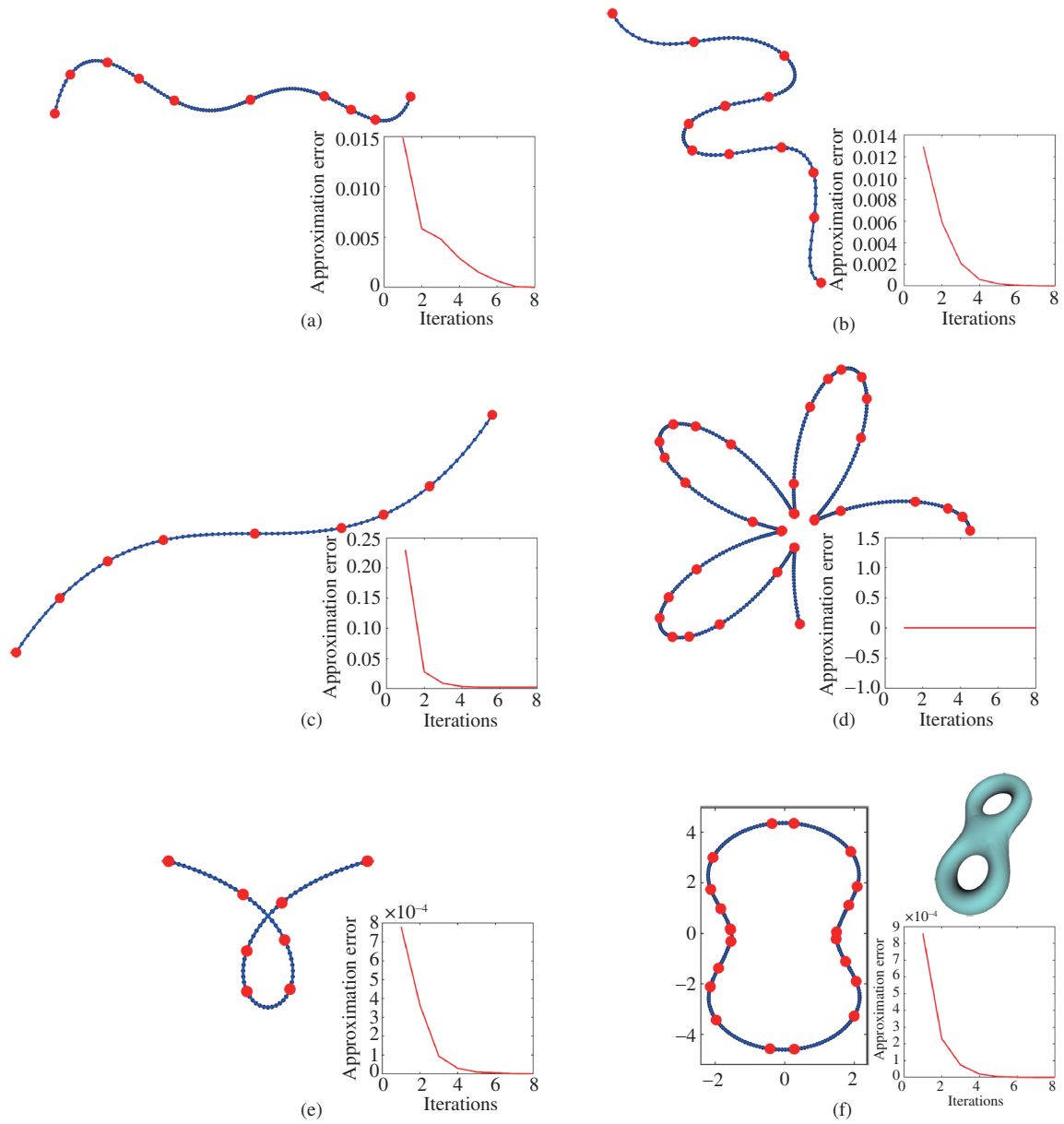


图 8 (网络版彩图) 更多类型曲线的逼近结果

Figure 8 (Color online) Approximation results of more types of curves. (a) B-spline 1; (b) B-spline 2; (c) polynomial; (d) plum line; (e) loop line; (f) section contour of Eight model

(5) Loop line (图 8(e)):

$$\begin{cases} x(t) = 10 \frac{t^3 - t}{t^2 + 1}, \\ y(t) = 10 \frac{t^2 - 1}{t^2 + 1}. \end{cases}$$

(6) 一般曲线 (图 8(f)): Eight 模型截面轮廓线, 模型如图 8(f) 右上子图所示.

图 8 显示了圆弧样条逼近和分段结果. Polynomial (图 8(c)), plum line (图 8(d)), loop line (图 8(e))

表 2 实验数据^{a)}
Table 2 Experimental data^{a)}

Curves	N_P	ε	N_{Arc}	e	t (s)
Bézier curve 1	1000	0.0001	5	5.0159×10^{-7}	15.0975
Bézier curve 2	1000	0.0005	2	6.6544×10^{-5}	12.3846
Leaf line	1000	0.0001	7	4.7171×10^{-7}	16.9222
B-spline 1	1000	0.0008	9	5.8841×10^{-6}	16.7884
B-spline 2	1000	0.0008	11	8.7299×10^{-6}	22.1343
Polynomial	2000	0.0001	8	2.8000×10^{-3}	20.1243
Plum line	4700	0.00008	30	2.9000×10^{-3}	76.8150
Loop line	1000	0.0008	7	1.3867×10^{-5}	13.3155
Section contour of Eight model	1000	0.0001	20	3.3010×10^{-8}	32.7293

a) N_P : input data size; ε : error tolerance; N_{Arc} : number of approximation arcs; e : approximation error; t : time.

和 Eight 模型截面轮廓线 (图 8(f)) 的分段结果说明, 新方法对对称性比较敏感, 在某些实际问题中, 可根据对称性简化计算. 从 plum line (图 8(d)) 的中心部分发现, 尖锐特征可以被自动检测出来, 能量图显示局部修正 (算法 1: Step 2) 循环 1 次即可得到最优结果, 进而说明初始稀疏分段方法 (问题 (3)) 的优势. Loop line (图 8(e)) 的实验结果显示了新方法对自我交叉点集数据的逼近能力. 表 2 列出本文所有例子的采样点数量 N_P 、稀疏逼近误差限 ε (问题 (3))、逼近圆弧段数 N_{Arc} 、逼近误差 e 和花费时间 t .

所有实验结果中, 圆弧段数和逼近误差充分说明了新方法的逼近能力, 但也有局限性. 优化问题 (3) 中变换矩阵 \mathcal{D} 依赖于输入数据, 抗噪能力差, 某些尖锐特征不是总可以得到理想结果的, 而且, 问题 (3) 中误差容限 ε 不固定, 可从表 2 中多组实验结果进行估计. 此外, 稀疏表示与多组未知量的共同优化算法需要更多的时间, 图 7 所示二分法^[5]的两条实验曲线在毫秒 (ms) 时间内即可完成, 新方法需要 12~15 s.

6 结语

本文提出了一个新的基于稀疏表示的 G^1 圆弧样条自动逼近方法. 全局的稀疏优化问题再次证明了稀疏表示技术对分段常值数据的强大处理能力, 新方法中连接点的自动检测能力较启发式方法有了较大提升. 稀疏分段得到的初始化圆弧样条, 需进行局部修正处理, 将初始连接点逐渐移动到最优位置满足 G^1 连续性质, 有趣的是, 在局部优化过程中, 随着连接点融合或者新的连接点出现, 圆弧段数也会发生变化. 实验结果表明, 本文逼近方法对对称性比较敏感, 部分尖锐特征、自交数据也可以得到较好的逼近结果.

圆弧样条的逼近问题一直是研究人员关注的研究课题, 稀疏表示也发展得非常成熟, 但圆弧样条的内在稀疏性质却一直没被发现. 所以, 在几何建模问题中, 对每个几何对象进行深层挖掘, 可能会有更多可以用稀疏表示方法解决的课题, 比如光滑曲线的高阶导数. 未来将会深入挖掘几何对象中的稀疏性质, 进一步研究稀疏表示方法在几何处理中的应用, 并探索算法的优化工作.

参考文献

- 1 Bolton K M. Biarc curves. *Comput-Aided Des*, 1975, 7: 89–92

- 2 Chen X D, Yong J H, Zheng G Q, et al. Automatic G^1 arc spline interpolation for closed point set. *Comput-Aided Des*, 2004, 36: 1205–1218
- 3 Walton D J, Meek D S. Approximation of quadratic Bézier curves by arc splines. *J Comput Appl Math*, 1994, 54: 107–120
- 4 Yeung M K, Walton D J. Curve fitting with arc splines for NC toolpath generation. *Comput-Aided Des*, 1994, 26: 845–849
- 5 Yong J H, Hu S M, Sun J G. Bisection algorithms for approximating quadratic Bézier curves by G^1 arc splines. *Comput-Aided Des*, 2000, 32: 253–260
- 6 Maier G. Optimal arc spline approximation. *Comput Aided Geometric Des*, 2014, 31: 211–226
- 7 Xu Z B, Sun J. Image inpainting by patch propagation using patch sparsity. *IEEE Trans Image Process*, 2010, 19: 1153–1165
- 8 Elad M. *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. New York: Springer, 2010
- 9 Parkinson D B, Moreton D N. Optimal biarc-curve fitting. *Comput-Aided Des*, 1991, 23: 411–419
- 10 Meek D S, Walton D J. Approximation of discrete data by G^1 arc splines. *Comput-Aided Des*, 1992, 24: 301–306
- 11 Meek D S, Walton D J. Approximating quadratic NURBS curves by arc splines. *Comput-Aided Des*, 1993, 25: 371–376
- 12 Ong C J, Wong Y S, Loh H T, et al. An optimization approach for biarc curve-fitting of B-spline curves. *Comput-Aided Des*, 1996, 28: 951–959
- 13 Held M, Eibl J. Biarc approximation of polygons within asymmetric tolerance bands. *Comput-Aided Des*, 2005, 37: 357–371
- 14 Meek D S, Walton D J. The family of biarcs that matches planar, two-point G^1 Hermite data. *J Comput Appl Math*, 2008, 212: 31–45
- 15 Ahn Y J, Kim H O, Lee K Y. G^1 arc spline approximation of quadratic Bézier curves. *Comput-Aided Des*, 1998, 30: 615–620
- 16 Marciniak K, Putz B. Approximation of spirals by piecewise curves of fewest circular arc segments. *Comput-Aided Des*, 1984, 16: 87–90
- 17 Meek D S, Walton D J. An arc spline approximation to a clothoid. *J Comput Appl Math*, 2004, 170: 59–77
- 18 Yang X N. Efficient circular arc interpolation based on active tolerance control. *Comput-Aided Des*, 2002, 34: 1037–1046
- 19 Kim T W, Kim Y C, Suh J C, et al. Internal energy minimization in biarc interpolation. *Int J Adv Manuf Technol*, 2009, 44: 1165–1174
- 20 Yang X N. Approximating NURBS curves by arc splines. In: *Proceedings of Geometric Modeling and Processing*, 2000. 175–183
- 21 Piegl L A, Tiller W. Data approximation using biarcs. *Eng Comput*, 2002, 18: 59–65
- 22 Piegl L A, Tiller W. Biarc approximation of NURBS curves. *Comput-Aided Des*, 2002, 34: 807–814
- 23 Maier G, Pisinger G. Approximation of a closed polygon with a minimum number of circular arcs and line segments. *Comput Geometry*, 2013, 46: 263–275
- 24 Wright J, Ma Y, Mairal J, et al. Sparse representation for computer vision and pattern recognition. *Proc IEEE*, 2010, 98: 1031–1044
- 25 Bishop C M. *Pattern Recognition and Machine Learning*. Berlin: Springer, 2006
- 26 Kaufmann M. *Data Mining Practical Machine Learning Tools and Techniques*. San Francisco: Morgan Kaufmann, 2005
- 27 Mallat S. *A Wavelet Tour of Signal Processing: The Sparse Way*. Burlington: Academic, 2008
- 28 Bruckstein A M, Donoho D L, Elad M. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Rev*, 2009, 51: 34–81
- 29 Waydo S, Kraskov A, Quiroga R Q, et al. Sparse representation in the human medial temporal lobe. *J Neurosci*, 2006, 26: 10232–10234
- 30 Rinberg D, Koulakov A, Gelperin A. Sparse odor coding in awake behaving mice. *J Neurosci*, 2006, 26: 8857–8865
- 31 Vapnik V N. An overview of statistical learning theory. *IEEE Trans Neural Netw*, 1999, 10: 988–999
- 32 Vapnik V. *The Nature of Statistical Learning Theory*. New York: Springer, 2013

- 33 Hastie T, Tibshirani R, Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. New York: Springer, 2009
- 34 Xu L L, Wang R M, Zhang J Y, et al. Survey on sparsity in geometric modeling and processing. Graphical Model, 2015, 82: 160–180
- 35 Digne J, Chaine R, Valette S. Self-similarity for accurate compression of point sampled surfaces. Comput Graph Forum, 2014, 33: 155–164
- 36 Avron H, Sharf A, Greif C, et al. ℓ_1 -Sparse reconstruction of sharp point set surfaces. ACM Trans Graph, 2010, 29: 1–12
- 37 He L, Schaefer S. Mesh denoising via ℓ_0 minimization. ACM Trans Graph, 2013, 32: 1–8
- 38 Wang R M, Yang Z W, Liu L G, et al. Decoupling noise and features via weighted ℓ_1 -analysis compressed sensing. ACM Trans Graph, 2014, 33: 1–12
- 39 Xu L L, Wang R M, Yang Z W, et al. Surface approximation via sparse representation and parameterization optimization. Comput-Aided Des, 2016, 78: 179–187

附录 A 变换矩阵 \mathcal{T} 和 \mathcal{D} 的计算

变换矩阵 \mathcal{T} 和 \mathcal{D} 对于圆弧样条初始化非常重要, 这里给出详细的计算方法.

第 3.3 小节中, 曲率的离散公式 (1) 关于数据点已经是非线性的, 拟梯度变换矩阵肯定更加复杂.

为了得到矩阵 \mathcal{T} 的显式表达, 进而有效求解问题 (3), 输入点序列 P , $P = (x_1, \dots, x_n, y_1, \dots, y_n)^T$, 通过定义下面两个系数:

$$z_i^x = \frac{x_{i+1} - x_i}{((x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2)^{\frac{3}{2}}},$$

$$z_i^y = \frac{y_{i+1} - y_i}{((x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2)^{\frac{3}{2}}},$$

式 (1) 变成

$$\begin{aligned} \kappa(\mathbf{p}_i) &= z_i^x(y_{i+1} - 2y_i + y_{i-1}) - z_i^y(x_{i+1} - 2x_i + x_{i-1}) \\ &= -z_i^y x_{i-1} + 2z_i^y x_i - z_i^y x_{i+1} + z_i^x y_{i-1} - 2z_i^x y_i + z_i^x y_{i+1} \\ &= \mathcal{T}_i P, \end{aligned}$$

将 z_i^x, z_i^y 作为常值系数, 很容易构造矩阵 \mathcal{T} , 第 i 行非零元素 \mathcal{T}_i 对应点 \mathbf{p}_i :

$$(-z_i^y, 2z_i^y, -z_i^y, z_i^x, -2z_i^x, z_i^x). \quad (\text{A1})$$

实际上, 为了避免下标越界, 不考虑 \mathbf{p}_1 和 \mathbf{p}_2 两点, 最终矩阵 \mathcal{T}_i 的大小为 $(n-2) \times 2n$, 对逼近结果没有影响.

类似地, 拟梯度为

$$\begin{aligned} \nabla \kappa_i^P &= \kappa(\mathbf{p}_{i+1}) - \kappa(\mathbf{p}_i) \\ &= z_i^y x_{i-1} - (z_{i+1}^y + 2z_i^y)x_i + (2z_{i+1}^y + z_i^y)x_{i+1} - z_{i+1}^y x_{i+2} \\ &\quad - z_i^x y_{i-1} + (z_{i+1}^x - 2z_i^x)y_i - (2z_{i+1}^x + z_i^x)y_{i+1} + z_{i+1}^x y_{i+2} \\ &= \mathcal{D}_i P, \end{aligned}$$

矩阵 \mathcal{D} 第 i 行非零元素 \mathcal{D}_i 为

$$(z_i^y, -z_{i+1}^y - 2z_i^y, 2z_{i+1}^y + z_i^y, -z_{i+1}^y, -z_i^x, z_{i+1}^x - 2z_i^x, -2z_{i+1}^x - z_i^x, z_{i+1}^x). \quad (\text{A2})$$

大小为 $(n-3) \times 2n$, 忽略 $\mathbf{p}_1, \mathbf{p}_2$ 和 \mathbf{p}_3 三点.

以图 1 中圆弧样条上的采样点为例, 矩阵 \mathcal{T} 和 \mathcal{D} 的有效性如图 A1 所示.

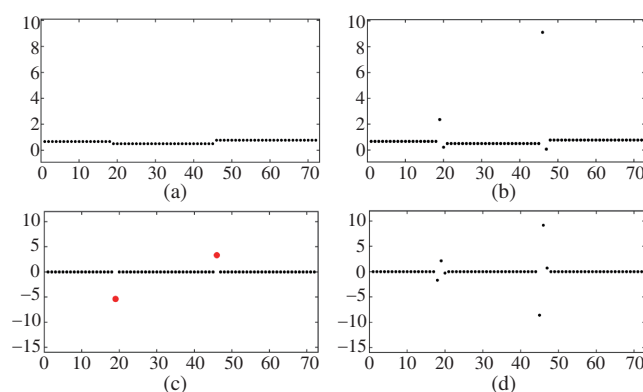


图 A1 (网络版彩图) 变换矩阵有效性示意图. (a) 采样点的曲率理论值. (b) 曲率拟梯度的理论值. (c) 变换矩阵 \mathcal{T} 计算的曲率近似值. (d) 变换矩阵 \mathcal{D} 计算的曲率拟梯度的近似值. 除了少量散乱点, 近似值与理论值一致, 实际优化过程中, 根据图 5 所述方法处理散乱点, 不影响逼近结果

Figure A1 (Color online) The effectiveness of transformation matrix. (a) Theoretical curvature values of sampling points. (b) Theoretical quasi gradient values of curvatures. (c) Approximate curvature values calculated by transformation matrix \mathcal{T} . (d) Approximate quasi gradient values of curvatures calculated by transformation matrix \mathcal{D} . Except for a few redundant points, the approximate values are consistent with the theoretical values. In the actual optimization process, the redundant points are processed according to the method described in Figure 5 without affecting the approximation results

Automatic G^1 arc spline approximation via sparse representation

Linlin XU

School of Computer Information Management, Inner Mongolia University of Finance and Economics, Hohhot 010070, China

E-mail: imufexll@163.com

Abstract Arc spline is popularized in describing the tool path for a certain CNC machine, and it is desirable to approximate curves with as few arcs as possible. Most of existing methods are heuristic; researchers have seldom paid attention to the piecewise constant property of the curvature of arc spline. Based on the intrinsic sparse property, this paper proposes an automatic G^1 arc spline approximation method via sparse representation. There are two main steps: global initialization using a sparse optimization problem for automatically detecting the joint points between adjacent arcs, and local modification by readjusting the positions of joint points to ensure the G^1 continuity. The experimental results show that the new method is sensitive to symmetry. Some sharp features and self-intersecting data can be approximated well, which fully shows the advantages of the new method.

Keywords arc spline, sparse representation, approximation, G^1 continuity



Linlin XU was born in 1988. She is a lecturer in the School of Computer Information Management, Inner Mongolia University of Finance and Economics. She received her Ph.D. degree in 2018 from University of Science and Technology of China. Her research interests include computer graphics, sparse optimization in geometry, and image processing.