Greedy Remeshing

2022 年 4 月 5 日

1 概念与定义

定义 1.1 *开集 $\mathcal{S} \in \mathbb{R}^d$ 的中轴 (medial axis)M 是 $\mathbb{R}^d$ 中距离 $\mathcal{S}$ 的边界 $\partial\mathcal{S}$ 有至少两个最近点的点所组成的集合的闭包.*

一种理解方式是, 如果以中轴上的点为球心作球, 则一定存在一个半径 r, 使得该球与 $\partial\mathcal{S}$ 相切且至少有两个相切点. 这样的球也称为 medial ball.
中轴不一定包含于 $\mathcal{S}$ 内部中. 例如 $\mathbb{R}^2$ 中两个不相交的半径相同的圆周, 它们圆心的垂直平分线就是中轴的一部分.

定义 1.2 *$\forall x \in \mathcal{S}$, x 的 local feature size/reach 是 x 到中轴 M 的距离, 用 $\rho(x)$ 表示.*

$$\rho(x) = d(x, M)$$

定义 1.3 *对于曲面 $\mathcal{S}$, 称 P 是 $\mathcal{S}$ 的一个 ϵ-sample, 若 $\forall x \in \mathcal{S}$, $\exists p \in P$,* s.t.

$$\|x - p\| \le \epsilon\rho(x)$$

, 又或者说 $\forall x \in \mathcal{S} \cap B(x, \epsilon\rho(x)) \neq \emptyset$. (感觉类似泛函里的 ϵ-网)

中轴的端点一般靠近那些曲率比较大的地方 (椭圆的横轴两边), 曲线上离中轴越近的点, 根据定义, 应该在这些点附近有更密集的采样. 而对于曲线上比较平的地方, 也会离中轴比较远, 因此采样不需要很密集. 这样 ϵ-sample 就能很" 灵活" 地刻画曲线.

定义 1.4 *限制在 $\mathcal{S}$ 上的 $\mathcal{P}$ 的 Delaunay 三角化是指*

$$\mathrm{Del}_{\mathcal{S}}(\mathcal{P}) = \{f \in \mathrm{Del}(\mathcal{P}) | f\text{对偶的 } \textit{Voronoi} \text{ 边与}\mathcal{S}\text{相交}\}$$

这里 f 是 $\mathrm{Del}(\mathcal{P})$ 中 facet(包括顶点, 边和三角形)

定义 1.5 *$\forall f \in \mathrm{Del}_{\mathcal{S}}(\mathcal{P})$, f 的 surface Delaunay ball 指使得 f 的三个点在球面上, 球心在 $\mathcal{S} \cap f^*$ 上的球, 其中 f^* 是对偶于 f 的 Voronoi 边. 称 surface Delaunay ball 与 $\mathcal{S}$ 的交集为 surface Delaunay patch.*

定义 1.6 *对于曲面 $\mathcal{S}$, 称 P 是 $\mathcal{S}$ 的一个 loose ϵ-sample, 如果满足以下两个条件:*
1. $\forall x \in \mathcal{S} \cap \mathcal{V}(E), E \cap B(x, \epsilon\rho_M(x)) \neq \emptyset$
2. $\mathrm{Del}_{\mathcal{S}}(\mathcal{P})$ 在 $\mathcal{S}$ 的连通区域上存在顶点

$\mathrm{Del}_{\mathcal{S}}(\mathcal{P})$ 可能为空. 由定义可以看出, loose ϵ-sample 是 ϵ-sample.

定理 1.1 *若 E 是 $\mathcal{S}$ 的 ϵ-sample, 那么当 $\epsilon < 0.1$, 它也是 $\mathcal{S}$ 的一个 loose ϵ-sample.*

2 算法

2.1 算法的理论依据

一个光滑曲面 $\mathcal{S}$ 的 ϵ-sample 具有这样的性质: 设 E 是 $\mathcal{S}$ 的 ϵ-sample, 则当 ϵ 充分小时, E 的 ϵ-sample 的限制在 $\mathcal{S}$ 上的 Delaunay 三角化 $\mathrm{Del}_{\mathcal{S}}(\mathcal{P})$ 会充分接近 $\mathcal{S}$.

2.2 算法步骤

该贪心算法需要输入一个曲面 $\mathcal{S}$ 和一个用户定义的 (user-defined) 函数 $\sigma : \mathcal{S} \to \mathbb{R}$(可取上面定义的 $\epsilon\rho$, ρ 是 $\mathcal{S}$ 上的点到中轴的距离), σ 时正的且满足 1 阶 Lipschitz 条件.
算法从一个小的样本点集 $E = E_0$ 开始, 在每一次迭代中, 更新 $\mathrm{Vor}(E)$ 和 $\mathrm{Del}_{\mathcal{S}}(\mathcal{E})$, 并更新" 不好的"surface Delaunay ball, 即 $B(c,r), \mathrm{s.t.} r > \sigma(c)$, 这些" 不好的" 球会被记录在列表 L 中. 在 L 中选择一个球, 将其球心 (Voronoi 边与 $\mathcal{S}$ 相交的点) 加入到 E 中. 直到 L 为空时, 算法结束 (从数学上证明了 L 最终一定为空).
论文证明了, 只要 $\sigma \leq \epsilon\rho$, 则该算法最终得到的 E 是 $\mathcal{S}$ 的一个 loose ϵ-sample.
该算法需要更新的量如下

- $\mathcal{P}$: 在 $\mathcal{S}$ 上的样本点集
- $\mathrm{Del}(\mathcal{P})$
- $\mathrm{Del}_{\mathcal{S}}(\mathcal{P})$
- L: 记录了 $\mathrm{Del}_{\mathcal{S}}(\mathcal{P})$ 中的"bad" facets

2.3 初始样本点集 E_0 的选取

定义 2.1 *设 $\mathrm{Del}_{\mathcal{S}}(\mathcal{E}_{\prime})$ 的一个面 f_0 的 surface Delaunay ball $B_0 = B(c_0, r_0)$ 满足 $r_0 \leq \frac{1}{3}\sigma(c_0)$, 则称 f_0 是一个 persistent facet.*

定理 2.1 *若 persistent facet f 在 E_0 中, 那么在上述算法的每一次迭代中, f 也会在 $\mathrm{Del}_{\mathcal{S}}(\mathcal{E})$ 中, 从而 f 也会在最终输出的结果中.*

为了保证 E_{end} 满足定义1.6-2, 可以让 $\mathcal{S}$ 的每一个连通分支至少包含一个 persistent facet 的一个顶点. 为此, 构造 E_0 如下:
在 $\mathcal{S}$ 的每个连通分支选取一个点, 组成点集 E'_0. 对于 E'_0 的任一点 x, 作一个以 x 为球心, 半径小于

$$\min\{\frac{1}{2}\mathrm{dist}(x, E'_i \setminus \{x\}), \rho(x), \frac{1}{3}\sigma(x)\}$$

的球 B_x, 并设 B_x 的球面为 Σ_x, 在 $S \cap \Sigma_x$ 上选三个点, 加入到 E_0 中, 就得到了初始样本点集. 我们有 $|E_0| = 3|E'_0|$.
E_0 的构造保证由 $x \in E'_0$ 确定的三个点组成了 x 所在的连通分支上的 persistent facet. 上式的第一项保证了 $\forall x \in B_x$, B_x 两两不相交. 因此由这三个点连接形成的三角形是一个 Delaunay 三角形, 第三项则保证了这个三角形是 persistent facet. 而第二项保证了这三个点一定会在 x 所在的连通分支上 (由中轴定义的注知, 会有中轴分隔开各个连通分支, 因此当半径小于点到中轴的距离时, B_x 就不会其他连通分支相交了).

注: 如果 $\mathcal{S}$ 是连通的, 也就是说 E_0' 只有一个点, E_0 只有三个点, $\mathrm{Del}_{\mathcal{S}}(\mathcal{E})$ 只有一个三角形, 那么算法可以继续吗? 是的, 因为 S 是闭合的, 因此与该 Delaunay 三角形对偶的 Voronoi 边 (因为只有一个 Delaunay 三角形, 所以它的 Voronoi 边是一条直线) 会与 S 有两个交点, 其中一个是 E_0' 的点, 另一个就是在第一次迭代中应该插入到 E 中的点.

2.4 算法分析

除了计算 Voronoi 图和 Delaunay 三角化, 其余只需计算 Voronoi 图中的边与 $\mathcal{S}$ 的相交点, 而且 Voronoi 图和 Delaunay 三角化不需要在每一次迭代中全局计算, 因为插入一个点只会影响一部分, 只需要原来的数据上更新部分, L 的更新同理. 从论文的实验结果可以看出, 该算法的速度还是比较快的.

论文还提到很多关于改进、加速、σ 函数选取等方面的内容. 但我看这篇论文的主要目的是为了理解 PMP 上提及的算法, 因此包括所有的数学证明以及更多的内容我都没有进一步阅读.