

网格模型上的离散测地线

赵俊莉^{①②}, 辛士庆^③, 刘永进^④, 王醒策^{①*}, 武仲科^①, 周明全^①, 贺英^⑤

① 北京师范大学信息科学与技术学院, 北京 100875

② 青岛大学软件技术学院, 山东 266071

③ 宁波大学信息科学与工程学院, 浙江 315211

④ 清华大学计算机科学与技术系, 北京 100084

⑤ School of Computer Engineering, Nanyang Technological University, Singapore 639798, Singapore

* 通信作者. E-mail: wangxingce@bnu.edu.cn

收稿日期: 2014-01-03; 接受日期: 2014-08-06; 网络出版日期: 2014-12-16

国家自然科学基金 (批准号: 61170170, 61170203, 61271366, 61322206) 资助项目

摘要 测地线是微分几何中的重要概念, 用于描述曲面上两点之间的最短曲线, 相当于平面上两点之间的直线段, 它在计算机图形学、图像处理、计算几何、计算机视觉等学科中有着广泛的应用. 自 20 世纪 80 年代以来, 关于离散测地线已有广泛研究, 众多学者提出了许多切实可行的算法. 本文将在介绍光滑表面上的测地线和离散网格上测地线概念的基础上, 对网格模型上的离散最短测地线 and 最直测地线的定义、性质及相关算法进行归纳总结, 重点讨论网格模型上离散最短测地线的相关算法, 包括完整网格和有缺陷网格上最短测地线的精确算法和逼近算法, 对各类算法进行深入研究, 详细论述每个算法的基本思想与实现方法, 从多个角度分析每个算法的优缺点, 并对他们各自的时间复杂度、空间复杂度及适用范围等进行对比, 最后对离散测地线的相关研究进行展望, 有利于后续对测地线算法的深入研究.

关键词 测地线 测地距离 最短测地线 最直测地线 网格 算法

1 引言

测地线起源于大地测量学, 最初用于测量大地, 又称大地线或短程线. 数学上测地线可视作直线在弯曲空间中的推广, 用于度量曲面上两点之间的最短路径长度, 是微分几何中的重要概念. 测地线作为曲面上一个重要的内蕴量, 在计算机图形学、图像处理、计算几何、计算机视觉等学科中有着广泛的应用, 如形状分析、匹配、参数化和纹理映射等. 三维网格模型的分割、人脸识别、曲面上流体的运动等很多应用都因引入测地线而取得了更好的结果. 因而如何求解离散测地线是计算机图形学主要研究的问题之一, 在数字几何处理中起着至关重要的作用.

自 20 世纪 80 年代以来, 关于离散测地线已有广泛研究, 众多学者提出了许多切实可行的算法. 在 2000 年, Mitchell^[1] 就对最短路径问题作了综述, 对之前的最短路径算法进行了总结. 对于之后的相关算法, 在 2011 年, Bose 等人^[2] 对完整网格模型上的最短测地路径问题, 从基于图的方法和基于采样的方法两方面进行了总结. 本文将在分析光滑表面上的测地线和离散网格上测地线概念的基础

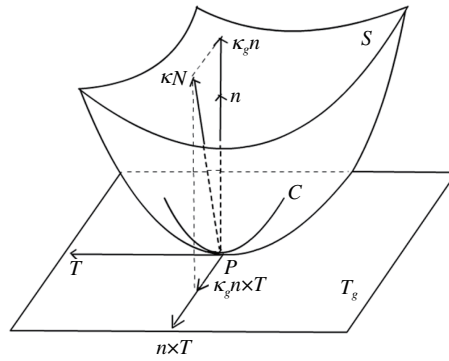


图 1 曲率向量分解图: 曲面 S 上的曲线 C 在点 P 处的曲率向量分解为法曲率 κ_n 和测地曲率 κ_g

Figure 1 Curvature vector decomposition: the curvature vector on Point P of Curve C on Surface S can be split into normal curvature κ_n and geodesic curvature κ_g

上, 重点对网格模型上离散测地线的相关算法进行归纳总结, 包括最短测地线算法和最直测地线算法, 其中最短测地线算法重点论述基于计算几何的精确算法和基于 PDE 的近似算法, 另外还阐述了近几年出现的在有缺陷的网格模型上计算最短测地线的算法。

2 测地线的基本概念

2.1 光滑曲面上测地线的定义

光滑曲面上的测地线的定义可以从测地曲率的角度来考察, 设曲面 $S: r = r(u, v)$ 上的弧长 s 的参数化曲线为

$$C: \begin{cases} u = u(s) \\ v = v(s) \end{cases},$$

\mathbf{T} 是曲线 C 的单位切向, \mathbf{n} 是曲面 S 的单位法向, κ 为曲线 C 在点 $r(u, v)$ 处的曲率, \mathbf{N} 为曲线 C 的单位法向, 则测地曲率的定义如下。

定义 1 曲面 $S: r = r(u, v)$ 上的曲线 C 在点 $r(u, v)$ 处的曲率向量 $\kappa\mathbf{N}$ 分解为

$$\frac{d\mathbf{T}}{ds} = \kappa_n \mathbf{n} + \kappa_g \mathbf{n} \times \mathbf{T}, \quad (1)$$

则称 $\kappa_n \mathbf{n}$ 为曲线 C 在曲面 S 上的法曲率向量 (场), 称 κ_n 为曲线 C 在曲面 S 上的法曲率 (函数); 称 $\kappa_g \mathbf{n} \times \mathbf{T}$ 为曲线 C 在曲面 S 上的测地曲率向量 (场), 称 κ_g 为曲线 C 在曲面 S 上的测地曲率 (函数)。如图 1 所示。

定义 2 “若曲面 S 上的曲线 C 的测地曲率向量恒等于零, 即曲线 C 的测地曲率 $k_g \equiv 0$, 则称 C 为 S 的一条测地线。”^[3] 其中测地曲率为曲线 C 的曲率向量在曲面的法向量 \mathbf{n} 和切向量 \mathbf{T} 叉成的向量上的投影。由以上定义可以看出, 测地线的曲率向量在曲面的法向量 \mathbf{n} 和切向量 \mathbf{T} 叉成的向量 $\mathbf{n} \times \mathbf{T}$ 上的投影为 0, 则该曲线的曲率向量仅在法向 \mathbf{n} 上有投影, 即测地线的法向与曲面的法向同向。因此有定理 1。

定理 1 曲面 S 上的一条曲线 C 是测地线, 当且仅当它或者是一条直线, 或者它的主法向量处是曲面 S 的法向量^[4]。很明显, 直线的曲率为 0, 因而测地曲率也为 0, 所以直线是平面上的测地线。

测地线是直线在曲面上的推广, 曲面上的测地线按照测地曲率衡量内蕴弯曲时是“直”的. 通过弧长的第一变分公式及其相应的定理可得, 曲面上两点之间的最短线只能由测地线实现^[1]. 因此, 测地线在曲面内蕴几何中的地位, 相当于直线在平面几何中的地位. 测地线具有明确的外在几何意义, 即: 曲线 C 为曲面 S 上的测地线当且仅当曲线 C 的曲率向量处处垂直于曲面 S 的切平面.

2.2 光滑曲面上测地线的性质

通过以上分析, 可将光滑曲面上的测地线的性质总结如下^[5]:

- (1) 如果将可展曲面展成平面时, 该表面上的曲线中只有测地线对应直线;
- (2) 曲面上非直线的曲线是测地线的充分必要条件是, 除了曲率为零的点以外, 曲线的主法线重合于曲面的法线;
- (3) 过曲面上任一点, 对于给定的一个曲面的切方向, 存在唯一一条测地线;
- (4) 光滑曲面上的任意两点之间存在一条测地线, 而且这两点之间的测地线可能不唯一;
- (5) 若给定曲面上充分小邻域内的两点, 则过该两点的测地线是连接这两点的表面上的曲线中弧长最短的曲线, 因此测地线又称为短程线.

因而, 光滑曲面上测地线具备很多互相等价的定义:

- (1) 测地线是局部最短的曲线.
- (2) 测地线是测地曲率为零的曲线.
- (3) 测地线是曲线的法向平行于曲面的法向的曲线.

2.3 光滑曲面上求解测地线

设 $\sigma: D \subset \mathbb{R}^2 \rightarrow S$ 是曲面 S 上的一个小区域,

$$\sigma(u, v) = (x(u, v), y(u, v), z(u, v)), (u, v) \in D. \quad (2)$$

表面上的一条光滑曲线 $\gamma(t) = \sigma(u(t), v(t))$ 是测地线, 当且仅当它满足测地线微分方程^[4]:

$$(E(u, v)u' + F(u, v)v')' = \frac{1}{2}((\partial_u E(u, v)(u')^2 + 2\partial_u E(u, v)u'v' + \partial_u G(u, v)(v')^2) \quad (3)$$

和

$$(F(u, v)u' + G(u, v)v')' = \frac{1}{2}((\partial_v E(u, v)(u')^2 + 2\partial_v E(u, v)u'v' + \partial_v G(u, v)(v')^2), \quad (4)$$

其中, $E(u, v) = \sigma_u(u, v)\sigma_u(u, v)$, $F(u, v) = \sigma_u(u, v)\sigma_v(u, v) = \sigma_v(u, v)\sigma_u(u, v)$, $G(u, v) = \sigma_v(u, v)\sigma_v(u, v)$.

在光滑曲面上求解测地线, 可以通过求解以上两个测地线微分方程得到测地线. 该方法能求得一条精确的测地线, 但只能应用于参数曲面上, 不能直接应用于其他模型, 而且计算复杂性较高. 因此, 后来的研究尽量避开解微分方程求测地线的纯数学思路, 转而考虑近似算法. 本文对此不做过多研究, 重点研究离散网格上测地线的求解算法.

2.4 离散网格上测地线的定义

离散的测地线求解问题最早以计算几何最短路径问题的形式出现, 最短路径问题最早出现在地理导航系统和机器人的运动路线控制^[6]等应用领域. Sharir 和 Schorr^[7]在 1986 年首次提出了表面上的最短路径问题, Mitchell 等人^[8]最早研究一种特殊情况的三维最短路径问题, 称为离散测地线问题

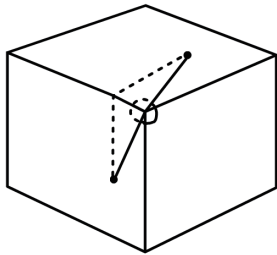


图 2 最直测地线与最短测地线

Figure 2 The straightest geodesic and the shortest geodesic

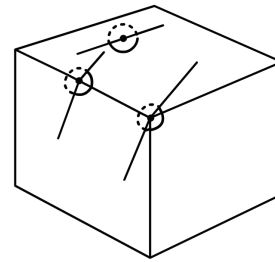


图 3 左周角 θ_l 和右周角 θ_r 相等的最直测地线

Figure 3 The straightest geodesic with equal left and right curve angles $\theta_l = \theta_r$

(discrete geodesic problem, DGP), 即给出位于一个给定的多面体网格表面上的两点, 我们寻找位于多面体网格表面上这两点之间的最短路径.

由于使用测地线微分方程求解测地线只能应用于参数曲面上, 因此在离散网格上求解测地线不能通过此方法求最短路径, 而是要研究离散情况下测地线的一些特殊性质, 从而求解离散测地线.

在离散网格模型中不能保持连续曲面的一些测地线的性质, 如在连续曲面上测地线既是最短的, 也是最直的, 但在离散曲面上最短测地线不一定是最直的, 最直测地线也不一定是最短的. 因此, 离散测地线可以分为最短测地线和最直测地线两类. 光滑曲面上测地线满足以下 3 个条件: (1) 是局部最短的曲线; (2) 测地曲率为零; (3) 曲线的法向平行于曲面的法向. 在离散情况下, 满足局部最短的曲线称为最短离散测地线, 或简称最短测地线. 满足测地曲率为 0 的曲线称为最直测地线. 图 2 中显示了两种类型的离散测地线, 它们都是最直测地线, 但只有左边虚线的那条是最短测地线.

最直测地线的形式化的定义为, 设 r 是多边形曲面 S 上的曲线, 如果 r 上的每一个点 p 的左周角 θ_l 和右周角 θ_r 相等, r 就是 S 上的最直测地线^[9](如图 3 所示).

其中关于左周角、右周角及周角, 辛士庆^[10]给出了如下定义: 在网格曲面 S 上的顶点 p 处, 如果与圆盘同胚, 定义 p 处的周角大小为以 p 为顶点的各角度之和. 若 S 的一条曲线 \hat{st} 经由顶点 p 并且一进一出, 则该曲线把 p 处的周角一分为二, 沿着曲线行进的方向, 把周角在左边的部分称作左周角, 把右边的部分称作右周角. 如果在顶点 p 处, 曲面 S 与半圆同胚, 则只能定义出左周角和右周角的其中之一, 于是, 本文规定另一个角度的大小为 ∞ . 在图 3 中用 θ_l 表示左周角 (虚线表示的角), θ_r 表示右周角 (实线表示的角).

离散最短测地线的长度最短性质为连接流形上给定两点之间的局部最短曲线的边界值问题的解, 而离散最直测地线则给出了对于流形上给定点开始沿任一方向有唯一测地线的初值问题的解^[11]. 最短测地线是度量, 不经过球面顶点, 而最直测地线因其不满足三角不等式, 所以不是度量, 任何通过球面点且平分其总曲面角的路径是最直测地线, 这是最短测地线与最直测地线的不同, 见表 1. 由于最短路径满足度量的性质, 在许多领域都有广泛的应用, 因而绝大部分的研究是关于最短测地线的, 而最直测地线的研究主要针对某些特殊场合的应用.

2.5 离散网格上测地线的性质

Mitchell 等人^[8]证明了离散网格曲面上两点间最短路径 (离散测地线) 的存在性, 并给出了离散网格曲面上测地线的一些基本性质, 这些基本性质是求解离散测地线的基础, 整理如下:

(1) 设 S 为连通离散曲面, 曲面 S 上从点 p 到其上任一点 q 存在一条测地线. 而且, 在 p, q 两点之间的测地线中, 至少有一条是最短的.

表 1 最短测地线与最直测地线的对比

Table 1 The comparison of the shortest geodesic and the straightest geodesic

Shortest geodesic	Straightest Geodesic
The shortest geodesic distance is a metric	The straightest geodesic distance is not a metric, due to the violation of the triangle inequality
The shortest geodesic path cannot pass through a spherical vertex	Any path through a spherical vertex p that bisects the total angle of p is a straightest geodesic
The shortest geodesic in metric geometry solves the boundary value problem of connecting two given points on a manifold with a locally shortest curve	The straightest geodesic gives a unique solution of initial value problem for geodesics with a given point and a given tangential direction

表 2 离散网格上的最短测地线与光滑曲面上的最短测地线的对比

Table 2 The comparison of the shortest geodesic on polyhedral surface and shortest geodesic on smooth surface

Shortest geodesics on smooth surfaces	Shortest geodesics on polyhedral mesh
Locally shortest geodesic is equal to locally straightest geodesic	Locally shortest geodesic is not equal to locally straightest geodesic (the locally straightest geodesic is not even a metric)
The geodesic path can pass through any point	The shortest geodesic path cannot pass through the spherical vertex
Given an initial point and a tangent direction, the geodesic path exists and is unique	When passing through a saddle vertex, a geodesic path can be split into many geodesic paths

(2) 在三角形内, 测地路径是直线段; 当经过边时, 如果将三角形展开使得两个相邻的三角形共面时, 测地路径也是直线段.

(3) 最短测地路径不经过球面点 (周角之和小于 2π 的点).

(4) 存在着的一组最短测地线通过鞍点 (周角之和大于 2π 的点), 但其中只有一条是最直测地线.

(5) 局部最短路径是一个交替的经过顶点序列和边序列 (可能为空) 的路径, 该路径使得沿任意的边序列展开成一个平面时, 它可展成一条直线段, 并且该测地路径通过的顶点处的角度大于或等于 π . 全局最短路径除了满足测地路径的上述条件外, 另外还要满足每个边序列必须是简单序列, 而且一条边不会同时出现在一个以上的边序列中.

2.6 光滑曲面上的测地线与离散网格上测地线的对比

离散网格上的最短测地线与光滑曲面上的最短测地线主要不同之处可总结为表 2 中列出的 3 点: 在光滑曲面上局部最短测地线等同于局部最直测地线, 测地线的路径可以通过曲面上的任何一点, 给定一个初始点和切线方向测地线存在且唯一; 而在离散网格上, 局部最短测地线与局部最直测地线不能等同 (局部最短测地线是度量, 而局部最直测地线不是度量), 最短测地线不能穿过球面顶点, 通过鞍顶点时, 测地线路径可以分裂成许多条测地线路径.

网格模型上的离散测地线的算法较多, 按所求的测地线类型, 可以分为最短测地线算法和最直测地线算法. 按所求的测地线精度, 可以分为精确测地线算法和近似测地线算法. 按所使用的方法来分, 可以分为基于计算几何的方法、基于图的方法和基于 PDE 的方法. 按源点和终点的数目可以分为单源单终点方法、单源多终点方法和多源多终点方法. 其中, “单源多终点” 的离散测地问题尤为重要, 它是解决其余两个问题的基础^[10]. 本文主要讨论在网格模型上的测地线算法, 由于目前绝大多数的测地线算法是在完整网格模型中求解最短测地线的, 因此在第 3 节详细讨论完整网格模型上的最短测

表 3 典型的精确测地线算法比较

Table 3 The comparison of typical exact geodesic algorithms

Application domain	Algorithm	Time complexity	Space complexity
Convex polyhedron mesh	Sharir and Schorr	$O(n^3 \log n)$	$O(n^2)$
	Mount	$O(n^2 \log n)$	$O(n \log n)$
	Schreiber and Sharir	$O(n \log n)$	$O(n \log n)$
General polyhedron mesh	O'Rourke	$O(n^5)$	Unknown
	MMP	$O(n^2 \log n)$	$O(n^2)$
	CH	$O(n^2)$	$O(n)$

地线算法, 包括基于计算几何的精确最短测地线算法和基于 PDE 的近似最短测地线算法. 第 4 节总结目前最新出现的在有缺陷的网格模型上的最短测地线算法. 第 5 节对最直测地线算法进行阐述. 最后在第 6 节对全文进行总结. 第 7 节阐述了挑战和展望.

3 完整网格模型上的最短测地线算法

本节主要讨论在完整网格模型上的最短测地线算法, 其中完整的三角网格模型是 \mathbb{R}^3 空间中的一张连通的流形曲面 S , 即 S 是由 n 个面、若干条边和若干个顶点组成的三角网格曲面, 并且 S 满足: (1) 不存在孤立的顶点; (2) 每条边只属于一个三角形面或被两个三角形面共享; (3) 任何两个三角形面要么不相交, 要么只有一个公共顶点或公共边; (4) 曲面上的任何两点间都存在限制于面上的路径 [10, 12].

基于图和计算几何的近似最短测地线算法在 Bose 等人 [2] 的综述中已有较为详细的论述, 因此本文主要论述基于计算几何的精确最短测地线算法和基于 PDE 的近似最短测地线算法.

3.1 基于计算几何的精确最短测地线算法

基于计算几何的精确最短测地线算法主要利用计算几何中点线面的关系, 对多面体网格模型求解精确测地线. 精确测地线算法的基本思想是: 在三角网格上, 从源点到目标点的精确测地路径是由一系列经过三角形网格面的线段组成, 将这些相邻的面展成一个平面时, 这些线段应当为一条直线. 具有代表性的精确离散测地线算法主要有 Sharir 和 Schorr [7] 提出的 SS 算法, Mitchell 等人 [8] 提出的 MMP 算法, Chen 和 Han [13] 提出的 CH 算法, Schreiber 和 Sharir [14] 提出的算法等, 其时间复杂度、空间复杂度及相应的适用范围如表 3 所示. 下面从凸多面体网格和一般多面体网格两方面分别进行阐述.

3.1.1 凸多面体网格上的精确算法

测地线算法的研究最早开始于凸多面体上的三维最短路径问题, 主要的算法有 3 个.

Sharir 和 Schorr 的 SS 算法. 1986 年, Sharir 和 Schorr [7] 最早研究了特殊情况下的三维最短路径问题, 提出了一个适用于凸多面体曲面上两点之间最短路径的精确算法, 其时间复杂度为 $O(n^3 \log n)$.

该算法的基本思想是将凸多面体曲面 S 分割成若干个连通区域 (peels), 使这些区域内不包含任何脊点 (脊点即沿着曲面 S 从源点到该点至少存在两条最短路径的点), 并且使得在每一个这样的区域中, 到其上任意一点的最短路径都完整的包含在该区域中. 这样就将复杂的凸多面体曲面上求解最

短路径的问题, 转化为在若干个分割后的区域中求解最短路径的子问题. 对于每个区域的求解最短路径的子问题, Sharir 和 Schorr 又证明了一系列的性质, 使得每个区域中的若干个面可以展成平面, 从而将空间中的最短路径求解问题转化为平面中的最短路径求解问题, 进而可以借鉴广泛研究的二维空间中求解最短路径的算法, 如 Dijkstra 算法等加以解决.

计算从源点 s 到凸多面体 S 上任意一点 t 的最短路径的算法步骤为: (a) 求分割 S 后的含有 t 的连通区域 P (假设 t 是 P 内部的一个点), 该区域内不包含任何脊点, 且其上任意一点的最短路径都完整的包含在该区域中. (b) 在最短路径 $\pi(t)$ 通过的区域 J_P 中找间隔边序列 ζ . (c) 将间隔边序列 ζ 相关联的面展成平面, 在平面上求得所需要的最短路径 $\pi(t)$. 该算法首先以 $O(n^3 \log n)$ 的时间预处理, 从源点构建分割 S 的连通区域 (peels), 将其存储在树中, 称为切片树 (slice tree), 使用切片树可以在 $O(n)$ 时间内确定 S 上任意一点和源点之间的最短路径. 该切片树的数据结构使用 $O(n^2)$ 的空间. 构建连通区域的算法与 Dijkstra 的图搜索算法类似, 然后将每个连通区域迭代地展开成平面, 最后在平面中求最短路径.

Sharir 和 Schorr 的算法是第一个求解三维曲面上的离散测地线的算法, 但它仅适用于凸多面体, 时间复杂度也较高, 而且算法偏重理论层面, 不易于编程实现. 该算法的实现很大程度上依赖于脊线的求解, 只有通过求解脊线将凸多面体曲面分区结束后, 才能求出最短路径, 因而繁琐费时.

Mount 的算法. Mount^[15] 改进了 [7] 的算法, 将该算法的时间复杂度优化至 $O(n^2 \log n)$; 同时, 空间复杂度降低到 $O(n \log n)$.

Mount 证明了 Sharir 和 Schorr 分割的连通区域 (peels) 可以看作是一个包含源点的展开平面的点集 R 的 Voronoi 区域, 而且 R 至多含有 n 个点, 所以从源点 s 到凸多面体上任意点 t 的最短路径至多为从 t 到 R 上任何一点 $r \in R$ 的最短路径加上 r 和 s 之间的最短路径距离. 因而可以同时使用遵循 Dijkstra 框架的算法计算点集 R 和 R 的 Voronoi 区域, 从而使算法的时间复杂度降到 $O(n^2 \log n)$. 另外, Mount 使用 $O(n^2)$ 的空间进行预处理, 然后通过排序类似的列表, 在构建之后存储数据结构时空间需求可以减少到 $O(n \log n)$.

Mount^[16] 后来又通过在凸多面体的边与从源点 s 开始的测地路径的交叉点上建立分层结构, 改进了存储 Voronoi 图的数据结构, 该数据结构以树存储凸多面体的每一条边 e , 树的叶子为边 e 和测地路径的交点. 由于不同的边可以共享相同的子树, 从而将空间复杂度减少到 $O(n \log n)$.

Schreiber 和 Sharir 的对凸多面体曲面的算法. 2006 年, Schreiber 和 Sharir^[14] 提出了在三维凸多边形曲面上计算从固定源点出发的最短路径的精确算法, 其时间复杂度为 $O(n \log n)$, 达到理论下限. 该算法扩展 Dijkstra 的算法, 允许连续更新. 也就是说, 一个波从源点沿凸多面体的边界传播, 在波前的拓扑结构改变这个事件发生时更新波前. 该算法基于 Hershberger 和 Suri^[17] 的 $O(n \log n)$ 的平面最短路径算法, 通过调整 R^3 中的凸多面体, 允许算法以离散的步骤在细分的“透明”的边之间完成传播. 该算法构建了一个动态版本的 Mount 数据结构^[16], 使用隐式编码编码曲面上从源点 s 到其他所有点的最短路径. 这种结构使他们的算法能在 $O(n \log n)$ 的时间内计算出凸多面体表面上的最短路径, 而且其空间复杂度也仅为 $O(n \log n)$. 算法推广到 m 个源点的情况下, 时间复杂度为 $O((n+m) \log(n+m))$.

Schreiber^[18] 扩展了文献 [14] 的方法, 用于所谓的真实多面体. 真实多面体被定义为三类非凸多面体. 第一类真实多面体有一个边界形成一个区域, 该区域的最大斜率的面受一个常数约束. 第二类真实多面体是一个不拥挤的多面体 (uncrowded polyhedron), 具有每个平行轴的正方形的边长为 l , 从多面体的任一顶点到其距离至少是 l , 并且它最多被多面体一个恒定面数分割, 这就是通常所说的低密度模型 (low density model). 第三类真实多面体是自共形的 (self-conforming), 每一条边 e 只有常数

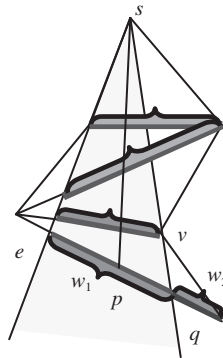


图 4 窗口的定义: w_1 为有向边 e 上的一个区间窗口, w_2 为伪源点窗口

Figure 4 The definition of window: w_1 is an interval window of the directional edge e , w_2 is a pseudosource window

个面的最短路径距离为 $O(|E|)$ ($|E|$ 为边 e 的长度). 上述三种情况下, 它们的算法运行时间复杂度和空间复杂度均为 $O(n \log n)$, 算法运行后产生一个表示最短路径的隐式图, 从而使得从源点 s 到任何查询点 t 的最短路径可以在 $O(\log n)$ 确定时间.

虽然文献 [14] 的算法已达到了理论下限, 但其只能适用于凸多面体, 对一般多面体曲面, 这个最佳时间仍然是一个公开的难题.

3.1.2 一般多面体网格上的精确算法

对于非凸多面体, 求解测地路径的主要难点在于非凸多面体 S 上两点间的测地路径可能会通过多面体的顶点, 而顶点处的展开不好定义. 众多学者都对一般凸多面体上测地路径的求解进行了研究, 下面对主要的算法分别进行阐述.

O'Rourke 的算法. 早在 1985 年, O'Rourke 等人^[19] 就提出一个在一般多面体上求解两顶点之间精确测地线的算法, 其时间复杂度为 $O(n^5)$. 该算法所求的测地线的源点和目标点都是多面体的顶点. 首先, 找出 S 的所有顶点对之间的“最短直线距离”. 设 π 是从 s 到 t 通过边 e_1, \dots, e_k 的序列, 如果与边 e_1, \dots, e_k 相邻的面展成平面时, π 成为直线, 则将 π 称为是一个直线路径 (straight line path, slp), 并将最短的那条直线路径称为最短直线路径. 最短直线路径对应的距离为“最短直线距离”, 这是通过扩展文献 [7] 的分割区域的技术来计算的. 第二步, 在顶点到顶点的图上找出源点和目标顶点之间的最短距离. 这相当于在图中找最短路径. 该算法需要 $O(n^5)$ 的时间来计算最短路径, 由于运行的时间复杂度太高, 因此该算法并没有得到实现.

MMP 算法. 1987 年, Mitchell 等人^[8] 提出了适用于一般网格曲面的 MMP 算法, 其时间复杂度降低为 $O(n^2 \log n)$ 、空间复杂度为 $O(n^2)$. 该算法使用与文献 [7] 算法类似的方法将多面体网格进行分割, 但是由于是在一般多面体中, 从源点发出的测地路径可以通过网格的顶点, 在这种情况下, Mitchell 等人将该点作为一个伪源点 v 进行处理. 因此, 多面体网格上任意一点 t 的测地距离, 是从源点 s 到 t 未通过 v 的最短距离, 和从最近的 v 到 t 的测地距离加上 v 到 s 的距离的最小值.

该算法是目前使用较多的一种精确测地线算法, 其基本思想是, 基于点光源光线直线传播原理求解三角网格表面上的测地线, 该算法的基本思想是将每个网格边分隔成一组间隔, 称之为窗口, 用窗口来记录具有共同边序列的一组最短路径信息, 然后从源点以 Dijkstra 扫描的方式将窗口传播到整个网格曲面, 使得每条边都被互不重叠的窗口覆盖, 这样就可以求出曲面上任意一点到源点的测地路径.

窗口可以分为伪源点窗口和区间窗口两种. 如图 4 所示, w_1 为边 e 上的一个区间窗口 (用粗线

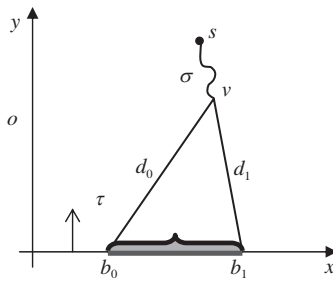


图 5 窗口参数化: b_0, b_1 为窗口两 endpoint 距离边缘的距离, 且 $b_0, b_1 \in [0, \|e\|]$, d_0, d_1 为伪源点 v 到窗口的两个端点的距离, σ 表示伪源点 v 到源点的测地距离, τ 表示伪源点 v 在边 e 的方向

Figure 5 The parameterization of window: b_0, b_1 measure the distance between the endpoints of the window along the edge, $b_0, b_1 \in [0, \|e\|]$, d_0, d_1 are the distances from pseudosource vertex v to the endpoints of the window respectively, σ represents the geodesic distance from v to the source, τ indicates the direction of pseudosource v about edge e

表示), 该窗口中的点到源点的测地线在展开平面上是直线段, 且与 w_1 中的任意一点 p 到源点的测地线经过相同的三角面片. 与上述 w_1 相似的区间窗口并不能完全覆盖网格上的所有边, 如图 4 中窗口 w_2 . w_2 中的任意点 q 到源点的最短路径都经过边界点 v , 即伪源点, 将 w_2 这类窗口称为伪源点窗口, 伪源点 v 到源点 s 的路径长度称为伪源点距离^[20]. 可以看出, 区间窗口用来记录当最短路径经由某条边时携带的信息, 而伪源点窗口是用来记录当最短路径经由某个顶点时携带的信息^[10].

Mitchell 等人只给出了精确求解测地线的算法, 但并未实现. 直到 2005 年, 才由 Surazhsky 等人^[21] 给出了一种实现方法. 整个算法实现时首先定义一个窗口函数, 然后生成窗口, 根据窗口计算出伪源点的位置, 则对于模型上任意一点, 可以计算出其经过窗口到达源点的最短路径长度, 即测地距离. 测地线通过回溯可以求出.

(1) 定义窗口函数. 算法在实现时首先定义一个窗口函数: $W(b_0, b_1, d_0, d_1, \sigma, \tau)$, 如图 5 所示.

(2) 窗口的生成、扩展及裁剪. 首先, 将与源点相邻三角面片的一条边作为窗口的初值, 然后在该面片的另外两条边上进行扩展生成新窗口. 所有生成的窗口, 都会根据其距源点的距离顺序存储在一个优先级队列中. 然后, MMP 算法从源点以 Dijkstra 的方式传播距离信息. 其基本步骤是从优先队列中选择 (和删除) 一个窗口并以以下方式扩展窗口, 重复这个过程, 直到队列为空. 窗口的扩展方式要考虑如图 6 所示的 3 种情况: 一个区间窗口可能生成一个新窗口 (如图 6(a) 所示); 也可能生成两个新窗口 (如图 6(b) 所示); 另外还要考虑伪源点窗口, 对于伪源点窗口, 若位于球面点 (周角之和小于 2π 的点) 处, 因为最短测地路径不经过球面点, 因此不必派生子窗口; 若位于鞍点 (周角之和大于 2π 的点) 处, 则该伪源点窗口应同时派生区间窗口和伪源点窗口 (如图 6(c) 所示). 当新生成的窗口与所在边的原有窗口重叠时, 就需要对这两个窗口进行裁剪.

(3) 测地距离的计算和测地路径的构建. 计算测地距离. 计算测地距离前, 首先要计算伪源点的位置. 对于模型上任意一点 p , 分别考虑其在三角面片的内部和在边或顶点上两种情况, 计算测地距离和测地路径. 对于位于三角面片内部的点 p , 如图 7(a) 所示, 其经过所在面的窗口 w_i 到达源点的最短路径长度为

$$D_i = \min_{p_{ij} \subseteq w_i} (|pp_{ij}| + |p_{ij}v_i|) + \sigma_i, \tag{5}$$

其中 p_{ij} 为第 i 个窗口上的第 j 个点, v_i 为第 i 个伪源点. 点 p 到源点的测地距离是其所在面的三条边的所有窗口中距离最小的, 即 $D = \min(D_i)$. 当点 p 位于边或顶点上时, 其到源点的测地距离为 $D = |ps| + \sigma$.

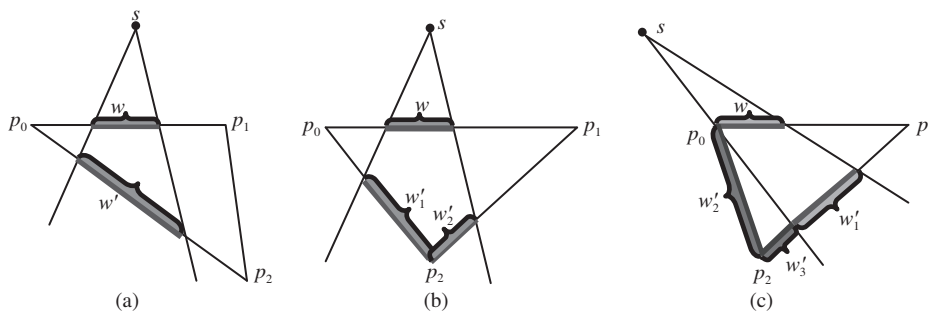


图 6 窗口扩展的三种情况

Figure 6 Three kinds of window extending. (a) Only extending one interval window; (b) extending two interval windows; (c) extending the pseudosource windows

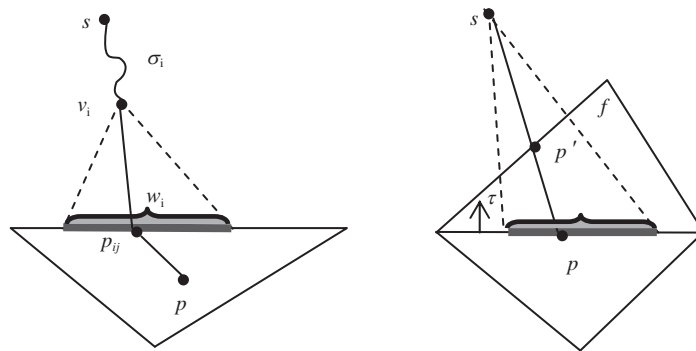


图 7 测地距离的计算

Figure 7 The computation of geodesic distances. (a) Computing the geodesic distance; (b) constructing the geodesic path

构建测地路径. 一旦所有的边被表示测地距离的窗口覆盖, 很容易追踪出从曲面上任意一点 p 返回到源点的最短路径. 首先找出 p 点在其所在的三角面片内求得最短测地距离的点所在的窗口, 该点就是测地路径上的第一个转折点, 然后计算伪源点的位置, 伪源点和该点的连线与三角形边的交点即为测地路径上下一个转折点, 如图 7(b) 所示. 接下来可以依次回溯所有到源点的最短测地距离所经过的窗口, 找到测地路径上所有的转折点, 这些点依次连接起来就组成了测地路径.

MMP 算法使用连续 Dijkstra 技术, 把 Dijkstra 算法的精神发挥到极致, 比 SS 算法的时间复杂度有很大的改进. Surazhsky 等人经过实验观察到, 虽然最坏情况下 MMP 算法的运行时间是 $O(n^2 \log n)$, 但在实验中算法的平均运行时间远远更低, 从一个源点到所有其他点的平均运行时间为 $O(n^{1.5} \log n)$. 但就复杂度而言, 依然存在着可以改进的地方: 首先, MMP 算法需要维护一个优先队列, 耗费的时间复杂度为 $O(\log n)$; 第二, 每当生成一个新窗口时, 需要定位和裁剪, 既耗时又会带来精度问题.

很多学者对 MMP 算法进行改进. 2007 年, Liu 等人^[22] 凭有效的技术来处理退化情况, 提高了 MMP 算法的鲁棒性. Bommers 和 Kobbelt^[23] 扩展了 MMP 算法, 可以处理从孤立的点到任意的嵌在网格中的多边形曲线的测地距离. 2009 年, 周竞文等人^[20] 注意到 MMP 算法中扩展得到的大部分窗口都没有用到, 针对这种情况, 结合 Dijkstra 距离, 提出了一种基于精确算法的剪枝算法, 在一定程度上提高了算法的效率. 2012 年, Liu^[24] 又针对经典的 MMP 算法进行了改进, 将精确测地线结构中使用两个半边结构合并到一个边的结构中, 并剪掉一些不必要的窗口分裂过程, 使该算法的效率比传

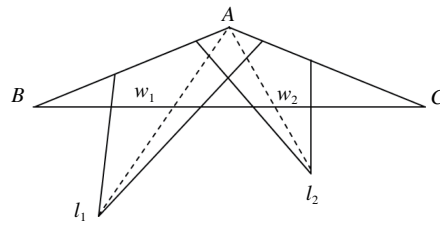


图 8 CH 算法中 one angle one split 引理
Figure 8 One angle one split lemma of CH algorithm

统的半边结构算法平均减少了 44% 的运行时间和 29% 的存储空间。

CH 算法. 1990 年, Chen 和 Han^[13] 脱离了传统的连续 Dijkstra 技术, 另辟蹊径提出了一种精确测地线算法 (CH 算法), 其时间复杂度为 $O(n^2)$, 空间复杂度为 $O(n)$. 该算法的核心思想是一个角产生一个窗口 (one angle one split), 从而避免了窗口间的相互裁剪. 同时, 建议用一个 $O(1)$ 时间的“先进先出”的普通队列替代 $O(\log n)$ 时间的优先队列. 事实上, CH 算法的思想是基于他们发现并证明的如下引理.

引理 1 假设两个窗口 w_1 和 w_2 在 $\triangle ABC$ 的同一个边 BC 上, 边 BC 的阴影覆盖顶点 A , 则角 $\angle CAB$ 的顶点至多有一个窗口可以派生出两个有意义的子窗口 (如图 8 所示).

CH 算法基于引理 1, 为每条有向边保存一个关键窗口, 以阻止无用窗口的派生. 然后逐层建立起一棵大小为 $O(n)$ 的窗口树, 使用普通队列, 按层次派生窗口事件, 以从根到叶广度优先搜索扫描的方式传播窗口, 直至层数达到总的面数为止. 算法的关键在于自始至终为每个角关联当前可以提供最短路径的那个窗口, 并且他们证明序列树中的节点数目为线性的, 包含所有最短路径, 并且可以在 $O(n^2)$ 的时间内构造完成.

CH 算法的优点在于提出一个角产生一个窗口的思想, 从而避免了窗口间的相互裁剪, 并使用普通队列代替优先队列, 使算法的时间复杂度降到 $O(n^2)$. 另外该算法对之前算法的一个改进之处在于, 之前的算法是沿着脊线分块的, 而 CH 算法沿着源点到各顶点的最短路径分块, 并证明了最短路径分块的结果也是连通的、无环的.

CH 算法的缺点主要是: (a) CH 算法派生出来的窗口仍有大量的无用窗口, 形成的窗口树中有大量的无用结点. (b) CH 算法使用普通队列代替维护一个优先队列, 虽然降低了时间复杂度, 但按层次派生窗口可能会使“最短”信号到达的比较晚, 这样造成了大量的无用信号的传播, 因此其实际效率并未提高. Surazhsky 等人^[21] 通过实验发现时间复杂度为 $O(n^2 \log n)$ 的 MMP 算法远比时间复杂度为 $O(n^2)$ 的 CH 算法要快的多. (3) 由于 CH 算法仍需对网格曲面沿着源点到各顶点的最短路径进行分块, 因此实现较为困难, 而且容易引入计算误差.

2009 年, Xin 和 Wang^[25] 研究了 CH 算法效率低的原因, 发现 CH 算法生成的窗口树过于庞大, 于是提出了改进的 CH 算法, 即 ICH 算法. 改进包括两个方面: (a) 对无效窗口进行过滤, 提出了一个简单有效的窗口过滤定理, 用顶点处的距离估计值过滤掉大部分无用的窗口; (b) 将 CH 算法中的普通队列改回用优先队列, 按照由近及远的方式扩展窗口. 通过以上两点改进, ICH 算法比原来的 CH 算法在性能方面提高了千倍以上. 与 MMP 算法相比, ICH 算法不仅速度更快, 而且占用的空间仅为其百分之一. 但其理论的时间复杂度却为 $O(n^2 \log n)$.

2013 年, Ying 等人^[26] 提出了并行 CH (PCH) 算法, 它将经典的 CH 离散测地线算法扩展成了并行算法, 大幅度提高了运算速度. 同所有的精确算法一样, 该算法也遵从窗口的传播框架. 将 CH 的顺

序算法以没有数据依赖的方式分为 4 个阶段: 选择窗口、传播窗口、数据组织和事件处理, 在每个阶段中的操作可以并行. PCH 算法能够同时且独立地传播大量的窗口. 该并行算法设计考虑 3 个关键因素: 以尽可能并行的方式传播窗口; 有效控制窗口的总数; 窗口的传播过程中避免数据冲突. PCH 算法的性能取决于网格的复杂性 $n = |V|$, GPU 的计算能力 (如核的数目和频率), 选择参数 k . 在 GTX 580 测试中, PCH 算法的速度比最先进的 CPU 上的 ICH 算法快了一个数量级.

SVG 算法. Ying 等人^[27] 在 2013 年提出了离散测地问题的一种新解决方案——鞍点图 (saddle vertex graph, SVG). 该算法通过预先构建鞍点图, 将求解离散测地线的问题转化为求解图中最短路径的问题, 从而可以使用之前广泛研究的最短路径算法 (例如 Dijkstra 算法) 来进行求解, 有效地降低了网格模型上求解离散测地线问题的时间复杂度.

由于网格模型上离散测地线不经过球面点, 且经过鞍点会分裂为多条测地线, 该算法充分利用这一性质, 得出网格模型上的一条全局测地线可以分成通过一个或多个鞍点的多段测地路径 (图 S1^[27]), 从而两条测地线可能共享某一段或多段测地路径 (图 S2^[27]). 因此可以并行地将这些测地路径段构建出来, 形成鞍点图 SVG, 然后再在鞍点图中求任意两点的最短路径, 就是网格中这两点之间的测地线.

SVG 是一个稀疏无向图, 用来记录完整的测地路径信息. SVG 图中的顶点是网格中的顶点, 并可分为鞍点和非鞍点两类. SVG 图中的每条边是一条直接测地路径 (即不通过任何鞍点的测地路径). 这些边可以分为三类: SS 边, 两个鞍点之间的直接测地路径; NS 边, 一个非鞍点和一个鞍点之间的直接测地路径; NN 边, 两个非鞍点之间的直接测地路径, 用 E_{SS} , E_{NS} 和 E_{NN} 分别表示 SS, NS 和 NN 边的集合. 图 S3^[27] 显示了 Bimba 模型上 SS, NS 和 NN 边.

显然集合 E_{SS} , E_{NS} 和 E_{NN} 是不相交的. 因此鞍点图包括 3 层: 第一层子图 $S_1 = (V_S, E_{SS})$, 由所有鞍点及连接鞍点的直接测地路径, 即 SS 边组成, 这是核心网络. 第二层子图 $S_2 = (V_S \cup V_N, E_{NS})$ 包含鞍点和非鞍点及连接非鞍点和鞍点之间的直接测地路径, 即 NS 边. 第三层子图 $S_3 = (V_N, E_{NN})$ 包含所有非鞍点, 及连接非鞍点之间的直接测地路径, 即 NN 边. Ying 等人已经证明一个连通三角形网格的 SVG 是连通的, 且一个连通三角网格上的全局最短测地线是与其相关联的 SVG 图上的最短路径, 因此, 通过构建 SVG 图, 就可以将网格模型上的离散测地问题转换成 SVG 图中的最短路径问题.

SVG 方法将求解全局测地线的问题有效地分成若干个局部求解测地线的子问题, 从而可以并行计算和重用信息, 而且构建 SVG 图也可以预先计算出来, 从而大大提高了算法的运行效率, 使该算法的时间复杂度达到了 $O(Dn \log n)$, 空间复杂度为 $O(Dn)$. 其中 $n = |V|$ 是网格顶点的数目, $D(D \ll N)$ 度量 SVG 的复杂性, 它是 3 个子图的平均度 D_{SS} , D_{NS} 和 D_{NN} 的最大值.

SVG 算法具有以下优点: (a) SVG 算法非常高效, 因为对一个真实世界的模型, D 远远小于 n , 所以该算法的时间复杂度接近 $O(n \log n)$, 而且由于 Dijkstra 算法不涉及任何数值计算, 因此 SVG 是远远快于快速行进法 (FMM). (b) 该方法计算的测地距离是一个度量, 即满足对称性条件和三角不等式. 由于它不需要任何数值求解器, SVG 方法是数值稳定的和健壮的, 而且用户可以直观地指定一个独立于模型的参数 K , 指定局部覆盖鞍点的范围, 有效地平衡 SVG 的复杂度和测地距离的精度. (c) SVG 可以以一个完全局部和并行的方式构建, 而且 SVG 的平均度取决于模型的详细信息, 它对网格大小不敏感, 具有更多细节和复杂的几何模型, 鞍点数量会越多, 从而 SVG 算法的效率会更高.

SVG 算法的不足之处是, 该算法只适用于三角网格模型上, 不能用于其他类型的网格模型, 因为它强烈地依赖于三角网格的局部结构. 另外, 该算法的复杂度取决于 SVG 图中边的数目, 如果 SVG 图是完全图 (例如凸网格), 会导致复杂度大大增加, 但庆幸的是 SVG 图在一般情况下是一个稀疏图, 因而复杂度大大降低了.

3.2 基于 PDE 方法的近似最短测地线算法

基于 PDE 的方法, 是通过解偏微分方程来得到离散测地线问题的近似解. 该方法相对容易实现, 一般求解速度较快, 但只能得到近似的测地线, 可应用于一些对精度要求不高, 但对速度要求比较高的情况.

3.2.1 Kimmel 和 Kiryati 算法

1995 年, Kimmel 和 Kiryati^[28] 假设一个离散曲面用给定的一个体素表示, 即空间被划分成一个立方体的网格, 每个格点被标记为位于曲面内侧、曲面上、或曲面外侧. Kimmel 和 Kiryati 提出的方法包括从粗略到精细两个阶段. 在第一阶段中, 在曲面体素定义的图上使用三维长度估计量和搜索图相结合的方法, 迅速获取全局近似的 shortest path. 在第二阶段, 进一步使用局部信息将近似的 shortest path 改进成最短测地曲线, 即局部最优路径. 使用一个有效的固定端点曲线进化数值算法, 将 3D 曲线缩短流变换到一个等价的二维形式. 通过这种方式, 可以找到两个网格点之间逼近的 shortest path. 虽然该算法求得的测地线的近似比是否有界未被证明, 但是由于曲面被假定为光滑的, 因此该算法得到的近似比是用现有的体素网格大小计算得到的最好的近似比. 该算法已实现并进行了测试.

3.2.2 Fast marching 算法 (FMM)

1996 年, Sethian^[29] 对一般网格提出了 $O(n \log n)$ 一阶近似快速行进算法 (fast marching method, FMM) 计算测地线. 1998 年, Kimmel 和 Sethian^[30] 将快速行进方法扩展到三角网格上. 2008 年, Weber 等人^[31] 提出了一种基于光栅扫描版的快速行进算法, 在几何图像上逼近测地距离. 由于其并行结构, 该方法可以在现代 GPU 上高效并行.

Fast marching 算法求解 shortest path 的主要思想是模拟波由近及远的传播过程, 从一源点出发单调地进行波的传播, 并在传播过程中存储各点的测地距离. 因为波的传播方向同距离函数变化最快的方向相同, 也是连接两点的最短路径的方向, 因而测地距离的计算可以通过解 Eikonal 方程 $|\nabla T| = F(x, y)$ 来求得. 当 $F = 1$ 时, 测地距离就等于曲面的传播速度时间 T . Fast marching 算法是求解 Eikonal 方程的一种数值解法, 其具体求解步骤为: 首先, 初始化源点的距离为 0, 并将其状态改为 Close (表示测地距离已经计算), 将与源点相邻的点状态改为 Alive (表示测地距离正在更新的点), 其余的点设为 Far (表示将要更新的点). 采用与 Dijkstra 算法类似的方法进行迭代. 每次迭代, 用所有网格中标记为非 Close 的点中最小的距离值更新标记为 Alive 的点的距离, 然后将 Alive 点标记改为 Close, 并将邻近的点的标记更新为 Alive. 整个过程循环进行直到所有点都标记为 Close 为止.

与 Dijkstra 算法^[32] 相比, Fast marching 算法与之不同的地方主要在于更新策略不同. Dijkstra 算法解出的 shortest path 局限于图的边上, 顶点的距离值每次是由邻接点更新. 而在 Fast marching 算法中, shortest path 可以通过网格中的三角面片, 因此顶点的更新是基于三角形的, 通过两个顶点的距离值更新. 另外, 更新的顺序也不同. Fast marching 算法在更新时总是选取与已有测地距离点相邻的点 (Alive 集合中的点) 中距离最短的进行更新, 这就是所谓的“单调属性”, 也就是说波总是向外扩充, 而不会回溯更新已有测地距离的点. 对于 Dijkstra 算法则是不断选取最短距离的点去更新以前的点. 因而 Fast marching 算法求出的测地线的精确程度要比 Dijkstra 算法高. 二者的异同见表 4.

Fast marching 算法 (FMM) 是一种广泛使用的快速有效的离散测地线近似算法, 许多人在此基础上进行了改进. Kirsanov^[33] 介绍了一种新的在 FMM 行军途中的更新规则, 此更新规则产生的 shortest path 具有较高的数值精度. Martinez 等人^[12] 提出了一种迭代方法, 改善现有的三角网格曲面上的两

表 4 Dijkstra 算法与 FMM 算法的对比
Table 4 The comparison of Dijkstra algorithm and FMM algorithm

	Dijkstra	FMM
Application domain	Graph	Mesh
Complexity	$O(E \log E)$	$O(n \log n)$
Quality	Poor	First-order approximation
Implementation	Easy	Specially handling an obtuse angle

个顶点之间测地路径的估计. 他们给出的迭代方案收敛到一个局部最小值, 但没有给出量化的改进或收敛速度. Xin 和 Wang^[34] 提出了另一个迭代方法改善了快速行进法的路径. 该算法首先提高了初始快速行进方法, 将多面体的边分成不同的类型, 并针对不同类型的边在波前传播时做不同的处理. 其次, 算法迭代提高了所得的最短路径直到找到精确地局部最短路径. 显然, 迭代方法效率在时间上比 FMM 低. Bertelli 等人^[35] 考虑使用 FMM 解决所有点对之间的最短路径问题 (all-pairs shortest path, APSP). 他们的目标不是简单地将每个顶点作为源点运行 FMM 算法 n 次, 而是利用不同的 SSSP 算法的冗余计算获得一个更有效的方法. 虽然算法在实验中取得了更高的效率, 但最坏情况下, 该算法的运行时间仍然是 $O(n^2 \log n)$. Giard 和 Macq^[36] 提出了一种替代方法, 在多面体曲面上解决 APSP 问题. 该方法在预处理步骤中将网格展开为平面, 选择许多参考顶点逐一进行展开. 展开阶段通过一个给定的参考点作为源点, 旋转所有的 Dijkstra 算法遍历的边. 这些边按顺序旋转到参考点的切平面. 在查询阶段, 新的源点位于它最接近的参考点的展开位置. 在此展开平面中用欧氏距离来近似新源点的测地线图. 该算法已实现并在实践中效果良好.

3.2.3 从梯度重建测地线的算法

2012 年, Xin 等人^[37] 提出了从梯度重建测地线的算法. 该算法的主要思想是, 根据曲面上测地距离都有一个唯一的极小点 (即源点), 以及单位梯度 $\|\nabla d\| = 1$ 的性质, 通过最小化能量函数解 Eikonal 方程

$$\min E(d) = \iint_S |\nabla d \cdot \nabla d - 1|^2 d\sigma, \quad (6)$$

其中 $d\sigma$ 是面积元素.

算法在实现时首先初始化距离场 d_0 : 可以使用欧氏距离、测地距离, 甚至一个随机的标量场进行初始化. 对于有裂缝的模型, 使用欧氏距离进行初始化效果较好. 然后通过以下 3 个步骤迭代更新距离场. (a) 给定当前的距离场 d , 通过拉普拉斯平滑 d , 然后归一化其梯度向量场, 对任意的面 $f \in F$, 使得 $\|\nabla d|_f\| = 1$. (b) 固定梯度 ∇ , 通过最小化能量函数 $E(d, \nabla)$ 更新它的距离 d , 相当于解一个稀疏线性系统.

$$\min E(d, \nabla) = \sum_{f_i \in F} A_i \|G_i - \nabla_i\|^2, \quad (7)$$

其中 ∇ 是单位向量, G 是离散的梯度算子. (c) 检查当前距离场是否包含最小点以外的源点. 如果包含, 我们使用拉普拉斯平滑去除这些最小点. 重复上述步骤, 直到超过最大迭代数目 I_{\max} , 或连续两次迭代之间的区别小于用户指定的公差 δ .

该算法最主要的优点是它是一种内蕴的全局算法, 不仅适用于完整网格模型上, 也适用于有缺陷的网格模型上, 对于网格的三角化细分程度和分辨率不敏感, 对噪声具有鲁棒性. 该算法的不足之处

在于采用这种方法求出的测地距离不能保证是度量, 其结果依赖于初始化, 也不能保证都收敛.

3.2.4 基于热流的算法

2013 年, Crane 等人^[38]提出了一种基于热流计算测地距离的新方法, 引入热流方法对给定域的指定子集 (例如点或曲线) 计算测地距离. 热流方法从本质上讲也是一种由梯度重建测地线的算法. 该算法的基本思想是通过热量从一点传递到其他区域求解测地线. 在某一点的热量, 随着时间的推移热量从一点传播到其余的区域, 可以通过一个热核函数 $k_{t,x}(y)$ 来描述, 即热核函数度量从一个源点 x 经过时间 t 后到一个目的地 y 传递的热量. 一个著名的热量和距离之间的关系是 Varadhan 公式, 该公式表明黎曼流形上的任何一对点对 x, y 之间的测地距离, 能够通过一个简单的热核逐点传播获得

$$\phi(x, y) = \lim_{t \rightarrow 0} \sqrt{-4t \log k_{t,x}(y)}. \quad (8)$$

这说明, 热扩散可以看作一个收集大量热粒子的从点 x 开始的随机漫步模型, 任何粒子经过一小段时间 t 后沿着最短路径到达远处的点 y . 因此, 基于热流的方法可以通过 3 个基本步骤求解 (图 S4^[38]): (a) $t = 0$ 时在某一点放一热源, 求解经过很短时间后的热流方程 $\dot{u} = \Delta u$. (b) 归一化向量场 $X = -\nabla u / |\nabla u|$. (c) 求解 Poisson 方程 $\Delta \phi = \nabla \cdot X$. 其中, 用 Δ 表示负的半定的 Laplace-Beltrami 算子作用在黎曼流形可微的实值函数上. 第一步通过 Laplacian 作用于温度场, 将热流方程的解作为测地距离求解的初值. 然后, 根据 Eikonal 方程, 梯度的真实距离是单位长度, 因此计算归一化的梯度场 $X = -\nabla u / |\nabla u|$. 最后通过最小化 $\int_M |\nabla \phi - X|^2$ 找到最接近的标量场 ϕ , 或等价地通过求解相应的 Euler-Lagrange 方程 $\Delta \phi = \nabla \cdot X$. 热量方法依据在于当时间间隔 t 趋于零时, 函数 ϕ 接近于真实的测地距离. 因此可以通过以上 3 步求解出测地距离场. 当求解从一个源点到其他点的测地距离时, 可以使用初始条件 $U_0 = \delta(x)$ (即 Dirac delta); 当计算流形 γ 上任意两点的测地距离时, 可以通过设置 U_0 为 γ 上广义的 Dirac.

该算法实现简单、高效, 收敛于精确距离, 经过一次预处理后, 可以在近线性的时间内计算出结果. 在实践中, 距离更新比最先进的方法快一个数量级的速度, 同时保持水平相当的准确性. 该方法只需要标准的微分算子, 因此可以适用于各种各样的对象, 如网格、三角网格、点云等.

基于热流的方法与从梯度重建测地线的算法, 主要思想都是初始化一个标量场, 然后归一化该标量场, 最后通过解一个线性系统来得到测地距离. 不同之处在于 Xin 等人^[37]的方法使用欧氏距离或测地距离初始化距离场, 然后通过多次迭代最终求得测地距离; 而 Crane 等人^[38]的方法使用热流的梯度场来重建距离场, 不需要进行迭代.

4 有缺陷的网格模型上的最短测地线算法

上述测地线算法通常认为网格具有良好的结构和流形. 然而在实际中, 网格往往含有缺陷如洞、裂缝、退化及非流形的情况, 甚至可能只是一些无组织的多边形 (a soup of polygons). 要应用以往的算法, 必须先修复这些破损的网格, 这往往需要大量的繁琐的手工工作. 于是近几年, 学者开始研究如何直接在有缺陷的网格上计算测地线. 有缺陷的网格模型是指, 对于一个由 n 个面、若干条边和若干个顶点组成的三角网格曲面 M , 不对其完整性做任何假设, 即 M 可能包含洞 (hole)、裂缝 (gap)、非流形 (non-manifold configuration)、退化 (degeneracies) 或缺少连通信息等^[39].

4.1 Azouz 的算法

2007 年, Azouz 等人^[40] 考虑在有部分缺失数据的三角网格流形上, 计算两个顶点 s 和 t 之间的测地距离问题. 该算法不是通过试图重建丢失的数据来估计测地距离, 而是通过多维尺度度量 (multi-dimensional scaling, MDS) 的方法将曲面嵌入到一个低维的欧氏空间, 然后通过最小二乘 MDS 方法用欧氏空间的欧氏距离估计测地距离.

该方法计算近似测地距离主要包括 3 个步骤: (a) 对网格 S 进行均匀分布采样, 采样后的集合为 P , 然后使用快速行进法 (FMM) 计算采样后任意两个顶点 P_i 和 $P_j (i, j \in P)$ 之间的测地距离. 由 FMM 计算的 P_i 和 P_j 之间的测地路径可能通过模型的一个洞, 因此是不正确的. 他们又计算可信度值 $\omega_{i,j} = 1 - \frac{m_{i,j}^h}{m_{i,j}}$ 作为不相似的程度, 其中, $m_{i,j}$ 是由 FMM 计算的测地路径的边数, $m_{i,j}^h$ 是从 P_i 和 P_j 的测地路径上通过 S 上的洞的边数. (b) 使用测地距离 $\delta_{i,j}$ 作为相异性, 可信度值 $\omega_{i,j}$ 作为权重, 通过多维尺度度量 (MDS) 方法将流形 S 嵌入低维的欧氏空间. 这样得到了一个 S 的规范形式, 该规范形式是 Elad 和 Kimmel^[41] 提出的, 使用 MDS 方法计算的从高维空间到低维欧氏空间样本点的一个映射. (c) 通过使用 Gower^[42] 设计的增加一个点到嵌入空间的扩展技术, 将 P_i 和 P_j 投影到标准形式, 使用嵌入点之间的欧氏距离近似估计原始点之间的真正测地距离, 从而计算出 S 上两个任意顶点 P_i 和 P_j 之间真正测地距离的估计值.

该算法计算出了 S 上两个顶点 P_i 和 P_j 之间的近似测地距离, 并说明可以在 $O(n \log n + npt)$ 的时间内预处理不完整的三角网格流形. 这样, 计算网格上的任意两点, 边界误差在一定范围内的测地距离估计值, 可以在 $O(n \log n)$ 的时间内完成. 该算法还给出了一个最坏情况下最优的测地距离误差的上限, 所计算的边界误差的上限是最坏情况下的最佳值.

4.2 Campen 和 Kobbelt 的算法

2011 年, Campen 和 Kobbelt^[39] 提出了在原始网格上以可容忍缺陷的方式, 计算有意义的近似内在距离和测地路径的思路. 其基本思想是抽象网状结构 (及其潜在的缺陷), 将其严格限制到输入元素所占用的空间壳体区域, 离散地执行所有计算. 已经证明, 这样外在壳体的距离在其逐渐增加的紧密性约束下, 统一地收敛于曲面的内在距离场^[43], 其具体步骤为 (图 S5^[39]): (a) 从网格表示抽象一个立方体复形, 即 Cartesian 网格, 严格限制到网格元素所占用的空间区域, 并用八叉树来表示; (b) 运用形态学的膨胀和腐蚀操作, 修补网格所在的八叉树上的洞等缺陷; (c) 使用 MMP 算法在八叉树中高效地生成测地距离场; (d) 将八叉树中的测地距离场投影到原始模型上.

该算法高效速度快, 对于较小的洞、裂缝等有缺陷的网格都适用, 而且由测地距离计算出测地线也很容易. 但其算法的准确性依赖于八叉树的精度和用户指定的阈值, 对于较大的洞、复杂的洞的效果不理想, 而且不能解决短路 (shortcut) 的情况, 内存消耗大, 是一种外在的方法, 很难应用于等距形变的情形.

4.3 边界曲线压平的算法

2012 年, Quynh 等人^[44] 提出了一个内蕴算法, 计算有洞的多边形网格上有意义的近似测地线. 他们的算法基于如下简单的观察: 平面内的最短路径可以通过洞. 因此可以通过修改经典的 MMP 或 CH 离散测地线算法, 仍然采用从源点到目的地保持波的传播方法. 当波达到洞的边界时, 将洞的空间边界曲线压平到平面, 并让波以平面上欧氏距离的度量方式通过洞. 然后, 将波传播到所有目标点.

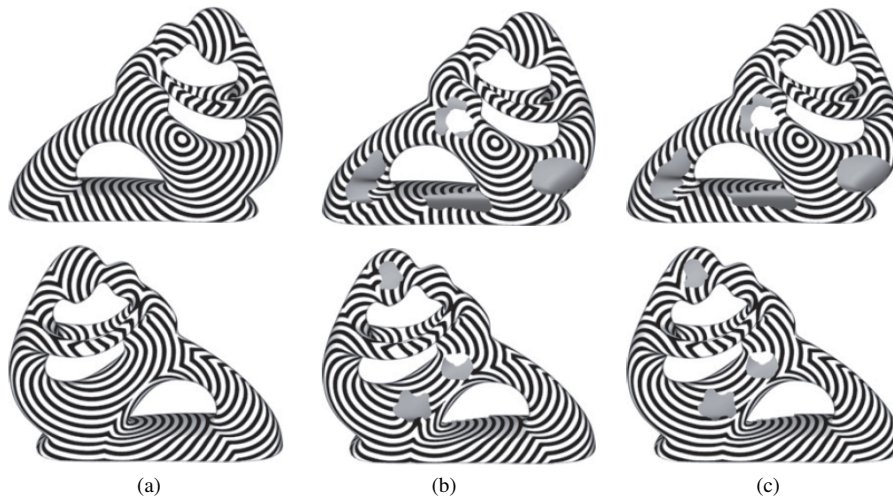


图 9 边界曲线压平算法求得的测地距离效果图 [44]

Figure 9 The geodesic distance computed by the method of unfolding the boundary curve [44], ©Copyright 2012 Elsevier. (a) Original surface; (b) surface with holes; (c) the geodesic distance computed by the method of unfolding the boundary curve

该算法在实现时首先使用现有的离散测地线算法计算网格上的测地距离场. 作为测地线算法的副产品, 将网格顶点划分成两类, 光直接照亮的点集 I 和由于洞而产生的阴影区域的点集 S , 根据它们到源点的距离排序所有洞, 存储到一个优先级队列 Q 中, 然后对每个洞通过求解一个非线性优化方程将边界曲线压扁到平面, 让波以平面上欧氏距离的度量方式通过洞, 即用新的洞边界计算近似的测地距离.

在上述算法中, 关键部分是将洞的空间边界曲线变形为平面曲线, 并尽可能地保持其本质的形状. 该算法采用最小化由长度、测地曲率和面积组成的目标函数

$$\min E(x) = \lambda_s E_s(x) + \lambda_g E_g(x) + \lambda_o E_o(x), \quad (9)$$

其中 E_s 和 E_g 保持曲线的长度和测地曲率, E_o 保持曲线所围成的面积. 采用模拟退火法 (SA) 进行优化, 计算出最优的平面曲线, 使其尽可能与空间曲线相似. 最后通过最小二乘法计算出到平面上伪光源的位置, 使波以欧氏距离的度量方式通过洞, 到达另一边. 边界曲线压平算法求得的测地距离效果见图 9.

与 Campen 和 Kobbelt 的算法使用网格修复技术相比, 该方法并不需要嵌入空间相关的操作 (如形态学运算) 去补裂缝或洞. 由于只度量内在特性, 如测地曲率, 边界三角形的边长和顶点的顺序, 边界曲线压平算法完全是内蕴的, 可以应用到等距形变的物体, 以及嵌入到高维空间的曲面. 如果曲面是可展的, 该算法可以得到准确的结果.

该算法实现时采用改进的 CH 算法 (ICH), 时间复杂度为 $O(n^2 \log n)$, 空间复杂度为 $O(n)$. 该算法没有明确的填洞操作, 而且只度量内蕴特性, 如测地曲率、长度和面积等, 因而是完全内蕴的, 并独立于嵌入空间, 因此, 它对于等距形变的对象和高维空间中的网格有效. 此外, 如果曲面是可展的, 该算法可以保证求得精确的解. 该算法的不足之处在于只适用于网格中有洞的情况, 而不能处理裂缝、短路等其他网格缺陷, 而且在算法实现时需要进行全局优化.

4.4 从梯度重建测地线的算法

2012年, Xin 等人^[37]提出了一种在破损网格上计算容忍缺陷的测地距离全局算法(详见 3.2.3 小节), 使用该算法对有缺陷的网格求得的测地距离如图 S6, S7^[37]所示.

该算法最主要的优点是它是一种内蕴的全局算法, 适用于等距形变的物体. 由于其全局性, 因此适用于有洞、裂缝、短路等有缺陷的网格, 对于网格的三角化细分程度和分辨率不敏感, 对噪声具有鲁棒性. 该算法的不足之处在于采用这种方法求出的测地距离不能保证是度量, 其结果依赖于初始化, 且不能保证结果都收敛. 该算法要求破损网格是连通的, 破损的部分不包含有意义的几何或拓扑特性, 而且拓扑短路的面积是相对较小的.

5 网格模型上的最直测地线算法

关于测地线的研究大量集中在最短测地线的研究上, 自 1998 年, Polthier 和 Schmies^[9]提出了最直测地线后, 人们开始了对最直测地线的研究.

Polthier 和 Schmies 在讨论多边形曲面的曲率的基础上, 定义了离散最直测地线, 并给出了离散最直测地线的初值定理: 假设 S 是一个多边形曲面, p 是 S 上的一点并具有切向量 $v \in T_p S$, 那么存在唯一的一条最直测地线 γ , 使得 $\gamma(0) = p$ 且 $\gamma'(0) = v$. Polthier 和 Schmies 还定义了离散的测地曲率, 并指出最直测地线平分顶点周角, 其测地曲率为 0 等性质, 以此定义了多边形曲面上向量的平行移动和向量域上积分的四阶 Runge-Kutta 法. 该方法需要使用网格上的方向域. 1999 年, Polthier 和 Schmies^[45]在上述基础上提出了各向相同的纹理映射技术, 可视化“测地圆”, 并模拟了最直测地线在网格曲面上一点开始向各个方向发展的情况, 包括分支和互相覆盖的情况. 尽管这个结果很漂亮, 但是该工作并未涉及直接在网格模型上求解测地线的算法. 2000 年, Hotz 和 Hagen^[46]使用直接取当前切方向在曲面上投影的方法, 求解参数曲面上的最直测地线, 这相当于求解测地线微分方程的一阶欧拉法. 而求解测地线偏微分方程一般采用 Runge-Kutta 法, 但是, 在离散网格上, 四阶方法的每一步都至少需要估计 6 个 Christoffel 符号中的 4 个.

Kumar 等人^[47]介绍了 3 种求解测地线的方法, 第一种方法是直接在 NURBS 曲面上求解; 第二种方法是基于最直测地线的定义, 在 NURBS 曲面的三角表示上直接求解; 另外还考虑了三角表示的法向, 比第二种方法更准确, 更接近第一种方法的结果. Lee 等人^[48]提出了基于切平面计算最直路径的算法. 该算法首先指定源点 S , 并构建通过源点和两个相邻的三角网格边界点的虚拟平面. 然后, 由虚拟平面的所有法向的平均确定源点的法线方向, 用这个点的法向计算出源点的基准平面 B . 最后, 该算法求垂直于 B 且通过源点 S 和目标点 D 的平面与三角网格的边的交线, 将该交线作为近似的最直路径. 该算法的优点是, 时间复杂度是线性的, 但求出的最直路径对源点和边界点敏感. 2012 年, Li 等人^[49]在 Lee 等人^[48]算法的基础上, 通过构建最佳切平面, 近似计算网格上两个顶点之间的最直路径. 首先通过源点和目标点的法线信息构建初始的切平面, 求出这个平面与网格边的交点, 这个交点在从源点到目标点的近似的路径上. 然后通过与以往的路径距离比较, 反复迭代创建一个最佳的切平面, 从而优化这个近似路径. 该研究表明, 最终基于最佳切平面的最直路径更准确, 而且对网格边界不敏感, 并将这个结果运用于网格参数化. 2009 年, Balasubramanian 等人^[50]将极小测地线定义为经过的网格顶点和边处测地曲率都是零的曲线. 他们提出了在多边形曲面上计算极小测地距离和最短路径的 LOS 算法, 该算法展开测地线经过的三角形链, 是最直测地线定义的左右周角相等的另一种实现. 该算法的时间复杂度为 $O(n^3)$. 2010 年, 程鹏等人^[11]提出了法截面方法和切向法向调整法两个

线性时间的算法, 求解三角网格上一点开始沿给定切方向的最直测地线. 法截面方法, 从几何的角度解释了算法的有效性和误差来源, 该算法具有一阶截断误差, 相当于连续曲面上求解测地线的欧拉法. 切向法向调整法, 在粗糙的网格上取得了连续曲面上的四阶 Runge-Kutta 法的准确性, 很好地解决了已有算法的累积误差问题, 对法向具有鲁棒性. 他们的算法不需要单独处理钝角三角形的情况, 在网格的顶点和网格边处都有统一的计算, 高效准确, 对累积误差问题有较好地解决.

目前, 最直测地线的求解算法比较少, 精度也比较低, 在求解时不能充分利用几何信息, 累积误差较为严重. 求解的思路也都局限在利用最直测地线基本的几何性质在局部求解, 误差较大, 还有待于进一步深入研究.

6 总结

本文在探讨光滑表面上的测地线和离散网格上测地线概念的基础上, 归纳总结了网格模型上的离散最短测地线 and 最直测地线的定义、性质及算法, 并重点讨论了网格模型上离散最短测地线的相关算法, 包括完整网格和有缺陷网格上最短测地线的精确算法和逼近算法. 详细论述了每个算法的基本思想与实现方法, 从多个角度分析了其优缺点, 并对比了其时间复杂度、空间复杂度及适用范围等, 有利于后续对测地线算法的深入研究.

最短测地线的精确算法从凸多面体上的求解算法发展到一般多面体网格上的求解算法, 主要是基于点光源光线直线传播原理求解三角网格表面上的测地线, 以 MMP 算法最为经典, 其他算法均是在其基础上对窗口的生成、扩展等方式进行改进, 来不断提高算法效率.

关于最短测地线的近似算法, 本文主要讨论了基于 PDE 的方法, 其中 Fast marching 算法是应用较多的近似算法, 速度快, 而且精度也较高. 基于热流的算法和梯度重建算法是目前最新的近似算法, 具有广泛的应用前景.

对于有缺陷的网格模型的最短测地线算法, 是最近几年研究的热点问题, 不同的学者使用各种方法来探索有效地解决该问题. Azouz 等人^[40] 使用多维尺度度量 MDS 的方法, 将曲面嵌入到一个低维的欧氏空间, 用欧氏距离估计测地距离. Campen 和 Kobbelt^[39] 用形态学的膨胀和腐蚀操作修补网格上的洞等缺陷, 近似内在距离和测地路径, 解决了在较小的洞、裂缝等有缺陷的网格上计算测地线的情况, 但不能解决短路的情况, 是一种外在的方法. Quynh 等人^[44] 及 Xin 等人^[37] 先后提出了两个内蕴的算法, 其中边界曲线压平的算法只能解决有洞的情况, 而梯度重建的算法则可以解决在有洞、裂缝、短路等各种缺陷的网格上计算测地线.

从以上分析可以看出, 关于网格模型上的离散测地线算法的研究主要集中在求解最短测地线方面, 这类算法相对较多, 而且研究较为深入, 研究方向一方面追求求解速度, 向测地线求解算法的理论下限 $O(n \log n)$ 的方向发展. 从目前精确算法的实现效率来看, 虽然在完整网格上求解最短测地线的 Schreiber 和 Sharir 算法已达到了理论下限 $O(n \log n)$, 但其只能适用于凸多面体, 对一般多面体曲面, 这个最佳时间仍然是一个公开的难题. 对于近似的最短测地线算法, 虽然大部分算法都能达到 $O(n \log n)$ 的理论下限, 但求解的精度还有待进一步提高. 另一方面最短测地线的求解算法, 追求算法的鲁棒性, 向破损网格中如何求解测地线方面发展, 目前对于有洞、裂缝、短路等缺陷的网格, 已有了初步的解决方案, 对于其他情况的缺陷网格还有待进一步探讨.

目前, 对于求解网格模型上的最直测地线的研究还较少, 求解算法的精度也比较低, 在求解时不能充分利用几何信息, 累积误差较为严重. 求解的思路也都局限在利用最直测地线基本的几何性质在局部求解, 误差较大, 还有待于进一步深入研究.

7 挑战与展望

基于上述关于网格模型上离散测地线的综述, 展望未来关于离散测地线的如下研究方向:

(1) 流形网格模型上提高离散测地线计算效率的研究. 随着目前三维模型的获取越来越方便, 模型数量越来越多, 精度越来越高, 如何在高维数据模型上提出高效快速、健壮、精确的离散测地线算法, 是今后进一步的研究方向. 一方面, 研究离散测地线计算的理论复杂度与实际运行时间之间差异的原因, 有助于降低离散测地线算法的时间复杂度. 如在之前的研究中提到过理论上 MMP 算法的时间复杂度高于 CH 算法, 但在实际中 MMP 的运行效率却显著高于 CH 算法, 这其中的原因值得深入研究. 另一方面, 也可以直接从如何提高离散测地线算法的计算效率入手, 研究现有算法可以改进的地方, 如 MMP, CH 算法中波的传播方式的改进等. 此外, 模型三角化的质量是否影响测地线算法的效率也有待研究.

(2) 非流形网格模型上的离散测地线算法研究. 目前大部分离散测地线算法对模型的要求比较高, 而很多由三维扫描仪获取的三维模型大部分都存在洞、非流形、退化的三角形等问题, 如何在这些非流形网格上高效快速地计算测地线是今后要解决的重要问题.

(3) 基于离散测地度量的各种三角网格模型高效处理与编辑技术研究. 由于测地线是曲面的内蕴量, 在等距形变的情况下是不变的, 可以应用于许多领域, 因而基于离散测地度量的各种三角网格模型高效处理与编辑技术也是未来的研究方向. 研究测地 Voronoi 图, 将 2D Voronoi 图的研究推广到三维网格模型上, 可应用于三维模型的重新网格化、采样等, 是计算几何和图形学中的重要研究问题^[51, 52]. 研究利用测地线建立的指数映射, 用于纹理映射、贴图合成、泊松圆盘采样等. 研究利用测地度量建立的姿态无关的形状描述符, 进行三维模型形状检索等. 还可以研究基于测地线的三维模型分割、动画变形等技术.

补充材料 图 S1~S7. 本文的补充材料见网络版 info.scichina.com. 补充材料为作者提供的原始数据, 作者对其学术质量和内容负责.

参考文献

- Mitchell J S B. Geometric shortest paths and network optimization. *Handbook Computat Geom*, 2000, 334: 633–702
- Bose P, Maheshwari A, Shu C, et al. A survey of geodesic paths on 3D surfaces. *Comput Geom*, 2011, 44: 486–498
- Wang Y N, Liu J Z. *Lectures on Differential Geometry*. Beijing: Beijing Normal University Publishing Group, 2003. 149–150 [王幼宁, 刘继志. 微分几何讲义. 北京: 北京师范大学出版社, 2003. 149–150]
- Chen W H. *Differential Geometry*. Beijing: Beijing University Press, 2006. 241–242 [陈维桓. 微分几何. 北京: 北京大学出版社, 2006. 241–242]
- Peng J G, Chen Q. *Differential Geometry*. Beijing: Higher Education Press, 2002 [彭家贵, 陈卿. 微分几何. 北京: 高等教育出版社, 2002]
- Belton A C R. *Geometry of Curves and Surfaces*. Lancaster University MATH329, 2012. 41–42
- Sharir M, Schorr A. On shortest paths in polyhedral spaces. *SIAM J Comput*, 1986, 15: 193–215
- Mitchell J S B, Mount D M, Papadimitriou C H. The discrete geodesic problem. *SIAM J Comput*, 1987, 16: 647–668
- Polthier K, Schmies M. Straightest geodesics on polyhedral surfaces. In: Hege H C, Polthier K, eds. *Mathematical Visualization*. Berlin: Springer, 1998. 135–150
- Xin S -Q. From the Discrete Geodesic Problem to the Dynamic Ordered Set Problem. Dissertation for Ph.D. Degree. Hangzhou: Zhejiang University, 2009 [辛士庆. 从离散测地问题到动态有序集. 博士学位论文. 杭州: 浙江大学, 2009]
- Cheng P. Computing Straightest Geodesics on Triangle Mesh. Dissertation for Master Degree. Jinan: Shandong University, 2010 [程鹏. 计算三角网格模型上的最直测地线. 硕士学位论文. 济南: 山东大学, 2010]

- 12 Martínez D, Velho L, Carvalho P C. Computing geodesics on triangular meshes. *Comput Graph*, 2005, 29: 667–675
- 13 Chen J, Han Y. Shortest paths on a polyhedron. In: *Proceedings of the 6th Annual Symposium on Computational Geometry*. New York: ACM, 1990. 360–369
- 14 Schreiber Y, Sharir M. An optimal-time algorithm for shortest paths on a convex polytope in three dimensions. *Discret Comput Geom*, 2008, 39: 500–579
- 15 Mount D M. On finding shortest paths on convex polyhedral. Maryland Univ College Park Center for Automation Research, 1985
- 16 Mount D M. Storing the subdivision of a polyhedral surface. *Discret Comput Geom*, 1987, 2: 153–174
- 17 Hershberger J, Suri S. An optimal algorithm for Euclidean shortest paths in the plane. *SIAM J Comput*, 1999, 28: 2215–2256
- 18 Schreiber Y. Shortest paths on realistic polyhedra. In: *Proceedings of the 23rd Annual Symposium on Computational Geometry*. New York: ACM, 2007. 74–83
- 19 O'Rourke J, Suri S, Booth H. Shortest paths on polyhedral surfaces. In: *Proceedings of STACS 85 2nd Annual Symposium on Theoretical Aspects of Computer Science*. Berlin: Springer, 1985. 243–254
- 20 Zhou J W, Cheng Z Q, Jin S Y. Dijkstra-based pruning geodesic algorithm. *J Syst Simul*, 2009, 21: 141–144 [周竞文, 程志全, 金士尧. 基于 Dijkstra 距离剪枝的测地线求解算法. *系统仿真学报*, 2009, 21: 141–144]
- 21 Surazhsky V, Surazhsky T, Kirsanov D, et al. Fast exact and approximate geodesics on meshes. *ACM Trans Graph*, 2005, 24: 553–560
- 22 Liu Y-J, Zhou Q Y, Hu S M. Handling degenerate cases in exact geodesic computation on triangle meshes. *Vis Comput*, 2007, 23: 661–668
- 23 Bommers D, Kobbelt L. Accurate computation of geodesic distance fields for polygonal curves on triangle meshes. In: *Proceedings of the 2007 Vision, Modeling, and Visualization Conference*, Saarbrücken, 2007. 151–160
- 24 Liu Y-J. Exact geodesic metric in 2-manifold triangle meshes using edge-based data structures. *Comput-Aided Des*, 2012, 45: 695–704
- 25 Xin S-Q, Wang G J. Improving Chen and Han's algorithm on the discrete geodesic problem. *ACM Trans Graph*, 2009, 28: 104
- 26 Ying X, Xin S-Q, He Y. Parallel Chen-Han (PCH) algorithm for discrete geodesics. *ACM Trans Graph*, 2014, 33: 9
- 27 Ying X, Wang X, He Y. Saddle vertex graph (SVG): a novel solution to the discrete geodesic problem. *ACM Trans Graph*, 2013, 32: 170
- 28 Kimmel R, Kiryati N. Finding shortest paths on surfaces by fast global approximation and precise local refinement. In: Melder R A, Wu A Y, eds. *SPIE 2356, Vision Geometry III*. Boston: SPIE, 1994. 198–209
- 29 Sethian J A. A fast marching level set method for monotonically advancing fronts. *Proc Natl Acad Sci*, 1996, 93: 1591–1595
- 30 Kimmel R, Sethian J A. Computing geodesic paths on manifolds. *Proc Natl Acad Sci*, 1998, 95: 8431–8435
- 31 Weber O, Devir Y S, Bronstein A M, et al. Parallel algorithms for approximation of distance maps on parametric surfaces. *ACM Trans Graph*, 2008, 27: 104
- 32 Dijkstra E W. A note on two problems in connexion with graphs. *Numer Math*, 1959, 1: 269–271
- 33 Kirsanov D. *Minimal Discrete Curves and Surfaces*. Cambridge: Harvard University Cambridge, 2004
- 34 Xin S-Q, Wang G J. Efficiently determining a locally exact shortest path on polyhedral surfaces. *Comput-Aided Des*, 2007, 39: 1081–1090
- 35 Bertelli L, Sumengen B, Manjunath B S. Redundancy in all pairs fast marching method. In: *Proceedings of 2006 IEEE International Conference on Image Processing*. Piscataway: IEEE, 2006. 3033–3036
- 36 Giard J, Macq B. From mesh parameterization to geodesic distance estimation. In: *Proceedings of 2009 European Workshop on Computational Geometry*, Brussels, 2009
- 37 Xin S-Q, Quynh D T P, Ying X, et al. A global algorithm to compute defect-tolerant geodesic distance. In: *Proceedings of SIGGRAPH Asia 2012 Technical Briefs*. New York: ACM, 2012. 23
- 38 Crane K, Weischedel C, Wardetzky M. Geodesics in heat: a new approach to computing distance based on heat flow. *ACM Trans Graph*, 2013, 32: 152
- 39 Campen M, Kobbelt L. Walking on broken mesh: defect-tolerant geodesic distances and parameterizations. *Comput Graph Forum*, 2011, 30: 623–632

- 40 Ben A Z, Bose P, Shu C, et al. Approximations of geodesic distances for incomplete triangular manifolds. In: Proceedings of the 19th Annual Canadian Conference on Computational Geometry, Ottawa, 2007
- 41 Elad A, Kimmel R. On bending invariant signatures for surfaces. *IEEE Trans Pattern Anal Mach Intell*, 2003, 25: 1285–1295
- 42 Gower J C. Adding a point to vector diagrams in multivariate analysis. *Biometrika*, 1968, 55: 582–585
- 43 Mémoli F, Sapiro G. Fast computation of weighted distance functions and geodesics on implicit hyper-surfaces. *J Comput Phys*, 2001, 173: 730–764
- 44 Quynh D T P, He Y, Xin S-Q, et al. An intrinsic algorithm for computing geodesic distance fields on triangle meshes with holes. *Graph Models*, 2012, 74: 209–220
- 45 Polthier K, Schmies M. Geodesic flow on polyhedral surfaces. In: Proceedings of the Joint EUROGRAPHICS and IEEE TCVG Symposium on Visualization, Vienna, 1999. 179–188
- 46 Hotz I, Hagen H. Visualizing geodesics. In: Proceedings of the 2000 Conference on Visualization. Piscataway: IEEE, 2000. 311–318
- 47 Ravi Kumar G V V, Srinivasan P, Devaraja Holla V, et al. Geodesic curve computations on surfaces. *Comput Aided Geom Des*, 2003, 20: 119–133
- 48 Lee S, Han J, Lee H. Straightest paths on meshes by cutting planes. In: Kim M S, Shimada K, eds. *Geometric Modeling and Processing*. Pittsburgh: Springer, 2006. 609–615
- 49 Li Z, Jin Y, Jin X, et al. Approximate straightest path computation and its application in parameterization. *Visual Comput*, 2012, 28: 63–74
- 50 Balasubramanian M, Polimeni J R, Schwartz E L. Exact geodesics and shortest paths on polyhedral surfaces. *IEEE Trans Pattern Anal Mach Intell*, 2009, 31: 1006–1016
- 51 Liu Y-J, Chen Z Q, Tang K. Construction of iso-contours, bisectors and Voronoi diagrams on triangulated surfaces. *IEEE Trans Pattern Anal Mach Intell*, 2011, 33: 1502–1517
- 52 Liu Y-J, Tang K. The complexity of geodesic Voronoi diagrams on triangulated 2-manifold surfaces. *Inf Proc Lett*, 2013, 113: 132–136

A survey on the computing of geodesic distances on meshes

ZHAO JunLi^{1,2}, XIN Shi-Qing³, LIU Yong-Jin⁴, WANG Xing-Ce^{1*}, WU ZhongKe¹,
ZHOU MingQuan¹ & HE Ying⁵

1 *College of Information Science and Technology, Beijing Normal University, Beijing 100875, China;*

2 *College of Software and Technology, Qingdao University, Shandong 266071, China;*

3 *College of Information Science and Engineering, Ningbo University, Zhejiang 315211, China;*

4 *Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China;*

5 *School of Computer Engineering, Nanyang Technological University, Singapore 639798, Singapore*

*E-mail: wangxingce@bnu.edu.cn

Abstract Geodesic, the shortest path between two points on a three-dimensional surface, is analogous to a straight line between two points on a plane, and is an important concept in differential geometry. It is utilized extensively in computer graphics, image processing, computational geometry, computer vision, and other fields. Geodesic algorithms have also been studied extensively since the 1980s, with many researchers proposing various practical algorithms. This paper summarizes the definition, property, and algorithms associated with the shortest geodesic and straightest geodesic on a mesh after introducing the concept of geodesic on smooth and polyhedral surfaces. The main algorithms discussed are discrete geodesic algorithms on polyhedral surfaces, including the exact shortest geodesic algorithms and the approximate shortest geodesic algorithms on integral meshes and defective meshes. Various algorithms are also analyzed in depth, with the basic underlying idea and method of realization of each algorithm discussed in detail, and the merits and demerits of each algorithm compared from different perspectives. Further, their time complexity, space complexity, and fields of application are also

compared. Finally, the prospects for discrete geodesic research are discussed with a view towards deeper study of geodesic.

Keywords geodesic, geodesic distance, shortest geodesic, straightest geodesic, mesh, algorithm



ZHAO JunLi was born in 1977. She is a Ph.D. candidate at Beijing Normal University and also a lecturer at Qingdao University. Her research interests include computer graphics, discrete differential geometry, and virtual reality.



XIN Shi-Qing received a Ph.D. degree from Zhejiang University. He was a research fellow at Nanyang Technological University 2009-2012, and is now a member of the faculty at Ninbo University. His research interests include algorithm research, computer graphics, and computational geometry.



LIU Yong-Jin is an associate professor at Tsinghua University. He obtained a Ph.D. degree in 2003 from Hong Kong University of Science and Technology. His research interests include computational geometry and computer graphics. Dr. Liu is a member of ACM, IEEE, and a senior member of the China Computer Federation (CCF).



WANG Xing-Ce was born in 1977. She is an associate professor at Beijing Normal University, where she majors in 3D modeling and 3D visualization. Her current research interests include computer graphic, medical imaging, artificial intelligence, and machine learning. Dr. Wang is a senior member of the China Computer Federation (CCF) and China Society of Image and Graphics.