



计算机科学

COMPUTER SCIENCE

基于法线迭代的模型中轴生成方法

宗迪迪, 谢益武

引用本文

宗迪迪, 谢益武. [基于法线迭代的模型中轴生成方法](#)[J]. 计算机科学, 2022, 49(6A): 764-770.

ZONG Di-di, XIE Yi-wu. [Model Medial Axis Generation Method Based on Normal Iteration](#)[J]. Computer Science, 2022, 49(6A): 764-770.

相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[基于 GPU 的并行 DILU 预处理技术](#)

GPU-based Parallel DILU Preconditioning Technique

计算机科学, 2022, 49(6): 108-118. <https://doi.org/10.11896/jsjcx.210300259>

[非均匀白噪声条件下的相干 MIMO 雷达角度估计](#)

Angle Estimation of Coherent MIMO Radar Under the Condition of Non-uniform Noise

计算机科学, 2022, 49(5): 262-265. <https://doi.org/10.11896/jsjcx.210300162>

[基于逐次超松弛技术的 Double Speedy Q-Learning 算法](#)

Double Speedy Q-Learning Based on Successive Over Relaxation

计算机科学, 2022, 49(3): 239-245. <https://doi.org/10.11896/jsjcx.201200173>

[基于机器学习的编译器自动调优综述](#)

Survey on Automatic Tuning of Compilers by Machine Learning

计算机科学, 2022, 49(1): 241-251. <https://doi.org/10.11896/jsjcx.210100113>

[基于高斯增强和迭代攻击的对抗训练防御方法](#)

Defense Method of Adversarial Training Based on Gaussian Enhancement and Iterative Attack

计算机科学, 2021, 48(6A): 509-513. <https://doi.org/10.11896/jsjcx.200800081>

基于法线迭代的模型中轴生成方法

宗迪迪 谢益武

大连海事大学信息科学技术学院 辽宁 大连 116026

(zongdizdf@163.com)

摘要 作为模型的降维表示,中轴因具有良好的性能,在许多工程领域得到了广泛应用。目前,模型中轴的生成方法主要基于近似中轴的思想,要么中轴的质量不高,要么计算的时间成本较高。由此,提出了一种基于法线迭代的模型中轴生成方法。法线迭代方法首先将模型离散化为三角网格模型,然后对样本点和三角面片进行基于中轴定义的 GPU 并行跟踪计算,经过多次法线迭代,得到所有样本点对应的中轴点,最后根据样本点的拓扑连接性连接对应中轴点来得到模型的中轴。实验结果表明,不同模型下该方法均可以相对快而精准地生成模型中轴,从而验证了所提方法能有效提升中轴生成的时间效率和精准性。

关键词: 中轴定义;法线;迭代;GPU 并行;中轴质量

中图法分类号 TP391.41

Model Medial Axis Generation Method Based on Normal Iteration

ZONG Di-di and XIE Yi-wu

College of Information Science and Technology, Dalian Maritime University, Dalian, Liaoning 116026, China

Abstract As the dimensionality reduction representation of model, the medial axis has been widely used in many engineering fields because of its good performance. At present, the method of generating the medial axis of the model is mainly based on the idea of approximating the medial axis, or the quality of the medial axis is not high, or the calculation time cost is high. As a result, a method of generating model medial axis based on normal iteration is proposed. The normal iteration method first discretizes the model into a triangular mesh model, and then performs GPU parallel tracking calculations based on the definition of the medial axis on the sample points and triangular faces. After multiple normal iterations, the medial axis points corresponding to all sample points are obtained. Finally, connecting the corresponding medial axis points according to the topological connectivity of the sample points to obtain the medial axis of the model. Experiment results show that the method can generate the model medial axis relatively quickly and accurately under different models, which verifies that the method improves the time efficiency and accuracy of the medial axis generation.

Keywords Definition of medial axis, Normal, Iteration, GPU parallel, Quality of medial axis

1 引言

中轴最早由 Blum 提出^[1],在数学中,中轴可以看作是模型最大内切圆心/球心的集合^[2]。作为物体的一种降维表示,中轴已经在相关工程领域有着具体的应用。在表面重建领域,基于中轴的表面重建高效且稳定^[3]。在路径规划领域^[4],相比传统方法中轴路径抗干扰能力更强,得到的路径更加健壮,因此得到相关学者的广泛关注。传统的碰撞检测代价较高,实时性不好,而基于中轴的碰撞检测是将三维模型转化为中轴球,从而实时地检测碰撞^[5]。随着计算机图形学技术的发展,可以预见的是,中轴在工程领域的应用将不断拓展。

计算模型的精确中轴并不是一项容易的任务,许多研究者进行近似中轴的计算,但是,要么计算代价高,要么获得的中轴不理想。大多数研究工作使用 CPU 作为处理器,影响了中轴生成的时间成本。另外,大多数工作的模型中轴一般是由中轴网格单元、中轴点或者中轴边线等构成,对于模型的

处理不能很好地拟合真实中轴。显然,提出一种高质量的模型中轴生成方法很有必要。

综上,本文提出法线迭代方法有效地生成高质量的中轴。该算法步骤如下:1)将模型离散为三角网格模型。2)将模型的包围盒分割成若干的网格单元,用于并行化处理提高时间效率。3)对法线分情况处理。4)以迭代方式使用数学逻辑来计算每个样本点的相应中轴点。5)通过样本点的拓扑连通性连接中轴点生成中轴。由于法线迭代基于中轴的数学定义,利用了中轴定义的衍生性质,从而能够有效提高模型中轴的质量。

2 相关工作

中轴概念自被提出以来,就一直备受科学界的关注。作为模型的降维表示,中轴在模型检索、路径规划、模型表示、三维打印、有限元分析等^[5]方面获得了广泛的应用。因此在过去几十年里涌现出了大量的中轴提取方法。以往的工作主要

基金项目:国家青年科学基金(61702074)

This work was supported by the National Science Foundation for Youth(61702074).

通信作者:谢益武(xieyiwu@dlmu.edu.cn)

可以分为三类,即瘦化、迭代和基于 Voronoi 图的方法^[6]。

瘦化(Thinning)法是在不改变模型拓扑结构的前提下采用迭代递归思想将模型由表及里地删除非核心部分进而得到模型中轴的方法。

Brunner 等^[7]提出了一种基于对象骨架图的网格分割方法,首先将三角面片网格转化为网格单元表示,而后采用瘦化算法提取网格单元骨架,通过迭代思想删除多余满足约束条件的网格单元来保持模型的几何特性以及拓扑特性。Yan 等^[8]引入了一个全局的测度度量,给出了三维空间中厚度侵蚀的精确定义,分析了它的性质,将二维形状内侧轴上的经典侵蚀厚度度量推广至三维,提出了一种在分段线性中轴上存在有界误差的有效逼近算法。Zhang 等^[9]提出了一种基于元胞自动机的并行图像细化算法和骨架化算法,假设对象(白色像素)是被许多蚂蚁(每个黑色像素)包围的猎物,蚂蚁的运动由细胞自动化控制,蚂蚁咬食猎物,直到猎物(物体)变成骨骼,该瘦化算法可以产生一种新型的被称为蚂蚁咬入骨架的骨架。瘦化法能够在一定程度上保证中轴的连通性,但是这种方法也有一些缺点,该方法针对体素进行相应的层层细化的操作,在这一循环迭代的过程中势必会消耗大量的时间和空间。

跟踪(Tracing)法是利用数学知识中法线量、曲率等工具来获得中轴的。

Zhong 等^[10]提出的一种在自由曲面上用跟踪算法进行中轴变换的方法,通过跟踪方法找到平分线,然后修剪这些平分线,找到分支点并构造中轴。Kusmakar 等^[11]提出了一种生成具有自由边界 3D 对象的内部中轴变换的算法,该算法通过沿着对象边界行进的跟踪技术生成中轴。近似级别由跟踪过程中的步长来控制。Makem 等^[12]建立了具有弯曲边界的平面区域的中轴变换的几何模型,基于此模型,生成了一种用跟踪算法计算平面区域曲线边界上的偏移曲线和中轴的方法,使用跟踪算法的结果作为初始值来使迭代过程有效。跟踪法的优点是适用性强,但其对边界的要求很高。

基于 Voronoi 图的方法是利用数学中的维诺图的概念来获得中轴的方法。

Shi 等^[13]利用 Voronoi 图计算原始中轴,使用改进的二次误差度量方法去除噪音。Chen 等^[14]提出了一种利用 Voronoi 图树计算单连通 NURBS 边界中轴的方法,在现有算法的基础上,实现了一种分而治之的竞争中轴计算方法。最后,利用竞争流方法,直观有效地检测出可能存在的分支点位置,并进行中轴的定界。Sun 等^[15]对 Voronoi 图和自由曲线的中轴进行了进一步比较和分析,使其能够适用于任意模型的中轴生成,通过迭代收缩边缘直到近似误差达到预定义的阈值来得到最终结果。Yan 等^[16]通过边界顶点的内部 Voronoi 图(被称为体素核心)近似体素形状的中轴。虽然 Voronoi 图法可以计算出较近似的体素核心,但是计算成本的线性增长使其难以处理更大的体量。

为了降低计算的复杂度,出现了一些比以上方法更高效的并行式中轴生成方法。此类方法将整个模型划分为多个部分,各部分之间并行计算中轴,最后进行组合,得到模型中轴。

Wagner^[17]提出了使用 GPU 有效细化二维和三维图像的新技术,使用编码的二进制图像来同时处理每个线程中的多个点。Saha 等^[18]提出了一个基于 GPU 的框架来提取模型

骨架,基于 GPU 框架法将样本量分为若干样本块,为每一块分配一个线程。Rebain 等^[19]以密集采样定向点集作为输入,为每一个中轴球的优化分配一个线程,计算近远场中有符号距离函数(SDF)的近似,从而得到一个近似的 MAT。这些方法并没有利用双向迭代收敛的方式,因此并行式方法的时间性能有待提高。Zhu^[20]等将双向迭代收敛用于二维形状的中轴生成,带来了相应的效率提升,但是无法适用于三维模型的中轴生成。

3 方法概述

基于法线迭代的模型中轴生成方法的整体流程如图 1 所示,主要包括预处理、法线迭代和中轴计算三大部分。

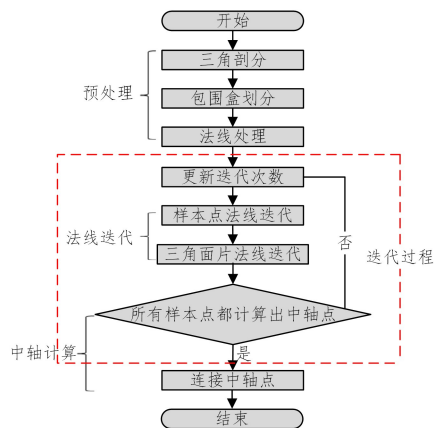


图 1 方法概述

Fig. 1 Method overview

下面针对每一小部分进行具体介绍。

(1)三角剖分:将模型离散为三角网格模型,得到若干样本点和三角面片。

(2)包围盒划分:将模型的包围盒划分为大小相等的(立方体)网格单元。

(3)法线处理:对样本点以及三角面片的法线进行估计,防止法线的不连续改变对后续中轴计算的影响。

(4)样本点法线迭代:通过迭代跟踪样本点的法线,将中轴点的计算范围限制在网格单元的范围。样本点的相互独立使得并行迭代跟踪成为可能。

(5)三角面片法线迭代:迭代跟踪三角面片的法线柱段,从而得到同一网格单元内若干样本点三角面片信息对,用于后续的中轴计算。三角面片的相互独立使得并行迭代跟踪成为可能。

(6)中轴计算:基于样本点以及三角面片的法线迭代跟踪结果,通过基于中轴定义的中轴计算样本点的中轴点。

(7)更新迭代次数:每一轮的迭代结束后,将迭代次数加 1,同时上一轮计算出中轴点的样本点不参与下一轮的迭代计算过程。

(8)连接中轴点:计算出所有样本点的中轴点之后,通过对样本点的拓扑连接性连接所得中轴点,得到模型中轴。

4 预处理

4.1 三角剖分

模型的三角剖分是为 GPU 并行法线迭代作准备,模型的表面是由各种曲面和平面组成,样本量、三角面片(三角

网格)的质量将直接影响后续生成中轴的质量。综上,首先遍历模型,对得到的平面或曲面进行三角剖分,得到若干样本点和三角面片,用于后续的法线迭代。

目前,对参数曲面的网格生成方法的研究主要集中在 Delaunay 方法和波前法 (Advancing Front Method, AFM)^[21]。二者时间复杂度相当,但 AFM 生成的网格质量高并且可控性好,表现出良好的曲面适应能力。AFM 采用递归的方法对区域进行分割、搜寻和判断,自动化程度高,并且在二维平面以及三维曲面中的应用已比较成熟。因此,本研究中选择使用 AFM 来实现模型表面的三角面片化,图 2 给出了一个环形结模型三角剖分示例。

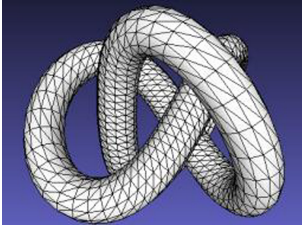


图 2 AFM 三角网格剖分
Fig. 2 AFM triangulation

4.2 包围盒划分

包围盒划分是为了在一定程度上缩小中轴点的搜索范围,进一步提高算法的时间效率。

包围盒划分的思想类似于三维模型的体素化,表示模型的空间体素跟表示图像的二维像素比较相似,差别仅在于二维的点扩展为三维的单元块。本文对中轴点计算采用的是样本点与三角面片的联合计算,所以每一个样本点的中轴点的计算都要先找到其对偶三角面片,模型的空间划分是将模型的包围盒划分成等大的网格单元块。包围盒的范围是通过模型在空间中 3 个方向的坐标范围来确定的。为了确保划分之后的网格单元数为整数,必要时可以对包围盒进行小范围调整,划分大小由 3 个方向的最小值和分辨率来确定。

4.3 法线处理

法线是方法的核心,因此需要对法线的处理进行一定的规范处理说明。

基于中轴的数学定义,中轴点一定位于距离最近的两个切点的法线上。一个三角网格上点的法线是统一的,它们构成了一个无限高度的法线三棱柱,简称为法线柱。如图 3 所示,网格 T 的法线柱用一个底面是三角面片 T 且无限延伸的三棱柱表示。

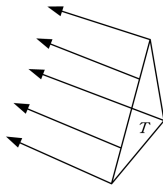


图 3 三角面片 T 的法线柱

Fig. 3 Normal column of triangular patch T

需要注意的是,网格点的法线可能存在不一致的现象,这将影响中轴生成。因此,为了避免法线的不连续改变在中轴生成中引起的不确定性,需要进行法线处理来保证法线的一致性。

4.4 三角面片的法线处理

在一个线段上,假设有端点 A, B , 其上任一点为 C , C 到

A, B 的距离分别为 $del(AC), del(BC)$, C 点的坐标可以表示为: $del(AC) * A + del(BC) * B$, 顶点 A, B, C 的法线表示为 N_A, N_B, N_C 。如果已知两个端点的法线,那么可以得到: $N_C = del(AC) * N_A + del(BC) * N_B$, 将其思想应用于面,同理,三角面片内点的法线是由三角面片 3 个顶点的法线共同决定的。对于三角面片来说,可以将区域大小对应长度关系。如图 4 所示,点 P 位于一个三角面片, P 将面分成了 3 个区域,区域大小分别记为 x, y, z , 三角面片区域比表示为: $S(PBC) : S(PAC) : S(PAB) = x : y : z$, 由此有 $x + y + z = 1$ 成立。区域面积类比线段长度,同理,三角面片上点 P 的坐标可以用点 A, B, C 表示为: $x * A + y * B + z * C$ 。顶点法线分别记为 N_A, N_B, N_C , 由此推知点 P 的法线 N_P 估计如下:

$$N_P = x * N_A + y * N_B + z * N_C \quad (1)$$

基于极限思维,随着 P 点的移动,如果点 P 移动到 3 个顶点处,那么, x, y, z 的值分别为 1, 其余两个值均为 0, 即 P 位于顶点处,则 N_P 等于 N_A, N_B 或 N_C , 经此处理便可以保证三角面片内点法线的连续性。同样利用极限思维,假设点 P' 是靠近点 P 的边界点,点 P' 的坐标可以表示为式(2):

$$P' = (x + \epsilon x) * A + (y + \epsilon y) * B + (z + \epsilon z) * C \quad (2)$$

其中 $\epsilon x, \epsilon y, \epsilon z$ 均为接近于 0 的极限值。基于前文,点 P' 的法线方向表示为:

$$N_{P'} = (x + \epsilon x) * N_A + (y + \epsilon y) * N_B + (z + \epsilon z) * N_C \quad (3)$$

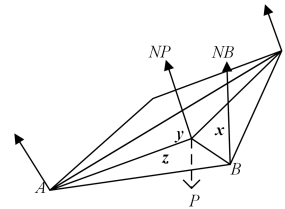


图 4 三角面片点的法线估计

Fig. 4 Normal estimation of triangular patch points

根据式(3),运用极限定理可知 P' 接近其邻近点 P 的方向。综上所述,基于本节针对法线的处理,使得三角面片内的法线可以保持一致。

4.5 相邻边的法线处理

除了三角面片上样本点 P 的法线需要处理,对于一个模型来说,相邻三角面片的相邻边上的法线如果不进行估计,也可能出现法线不唯一的情况,这种情况会影响三角面片内点的法线估计结果的不一致性。基于上述考虑,需要对三角面片上相邻边的法线也进行相应处理。

三角面片的相邻边可以分为两种情况讨论。

- (1) 两个三角面片 M 和 N 的相邻边为凸边。
- (2) 两个三角面片 M 和 N 的相邻边为凹边。

针对上述两种情况进行相应处理,图 5 给出了凸边和凹边条件下的法线处理结果。

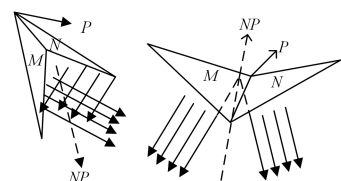


图 5 凸边/凹边的法线估计

Fig. 5 Normal estimation of convex/concave edges

5 法线迭代

法线迭代是方法的核心思想,法线迭代方法涉及多轮迭代。在第 i 轮中(i 从 0 开始),其内侧半径在 $(i * gl, i * gl + gl)$ 范围内的样本点将通过两次法线迭代生成它们的中轴点,该范围称为第 i 轮的当前范围。在每轮迭代之后,所涉及的样本点的数量将变得更少,因为在上一轮迭代中已经计算了一些样本点的中轴点,所以它们不需要参与下一轮迭代。在执行若干轮迭代之后,所有样本点都将获得其中轴点,算法结束。对于每一轮迭代,这两次法线迭代分别被称为基于样本点的第一次法线迭代和基于三角面片的第二次法线迭代。

算法 1 法线迭代算法

输入:模型的信息包括三角面片 T 和样本点 P 的信息,网格单元的长度 gl

输出:样本点 P 的中轴点 M

```

1. int i=0//初始化迭代次数从 0 开始迭代
2. while(p.hasMA=false 的数量 >= 1)//至少有 1 个样本点还没有得到对应的中轴点
3. set 范围为  $[i * gl, (i+1) * gl]$ //初始为  $[0, gl]$ //将首轮法线迭代范围设为一个网格单元的范围
4. for 每个样本点在当前范围内进行法线迭代得到相交网格
5. end for
6. for 每个三角面片在当前范围内进行法线迭代得到相交网格  $v$ 
7.   for 对于与三角面片相交的每个  $v$ ,遍历样本点信息。
8.   for 每个三角面片和样本点信息对通过 5.3 计算
9.     if 样本点所在面为平面,若  $r$  在当前范围内,  $p.hasMA=true$ //将  $P$  标记为已经获得中轴点,不进行下一次的迭代。
10.    end if
11.   if 样本点所在面为曲面
12.     if  $r1=r2$  唯一在当前范围内,  $p.hasMA=true$ 
13.     end if
14.     if  $r1 \neq r2$  均在当前范围内,  $r = \min(r1, r2)$ ,  $P.hasMA=true$ 
15.     end if
16.   end for
17. end for
18. end for
19. end for
20. 更新  $i$  值为  $i+1$ //迭代轮数
21. end while//直至所有样本点的中轴点计算结束,结束整个迭代过程。

```

5.1 样本点法线迭代

样本点的法线迭代是通过迭代来追踪样本点的法线,这个过程中样本点之间彼此相互独立,因此为每一个样本点分配一个 GPU。

根据中轴的几何定义,其中轴点一定位于其中轴球的球心,而中轴球球心与样本点的连线与 P 点切线为垂直关系,因此,中轴点一定位于点 P 的法线 N_P 上。在第 i 轮,如果一个样本点 P 的中轴点 M 在该轮迭代得到,那么 P 的中轴半径一定在当前范围内。以图 6 为例,假设 P 为模型上任意一点, P 的中轴点 M 一定位于 P 的法线上,若该点的中轴点可以在当前范围内被计算出,那么点 M 一定位于法线段 ES 上。

每个样本点的中轴点所在范围都是未知的,为了避免在

模型的整个空间对中轴点进行搜素,基于网格单元的范围,采用局部搜索的方式,将样本点可能的中轴点的范围控制在一些网格单元中,这也正对应了法线迭代的核心思想。为了节省空间,设置链表来保存网格单元范围内的样本点信息。后续在计算中轴点时,直接以网格单元的链表信息计算即可。由前文所述,样本点 P 的中轴点位于 ES 上, ES 长度为 gl ,体素化过程中的网格单元的长度也为 gl ,因此,若网格单元与样本点在当前范围内的法线相交,则认为该网格单元可能包含样本点 P 的中轴点 M 。

需要说明的是,这里与 ES 相交的网格情况需要分类讨论。

(1) ES 均落在同一个网格单元范围内。

(2) ES 在面相邻的两个网格上, ES 横跨两个网格面,此时两个网格均将 P 保存在链表中。

(3) ES 在边相邻的 3 个网格上, ES 穿过了 3 个网格单元,此时 3 个网格均将 P 保存在各自的链表中。

(4) ES 在点相邻的 4 个网格上,此时 4 个网格均将 P 保存在各自的链表中。

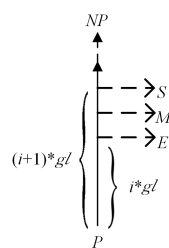


图 6 样本点的法线迭代

Fig. 6 Normal iteration of sample points

5.2 三角面片法线迭代

样本点法线迭代后的结果使得一些网格单元有了一些候选样本点,但是后续的中轴点计算需要相应的三角面片来联合计算。为了得到相应的三角面片,需要对三角面片也进行法线迭代,使得其法线能落在一些网格单元范围内,并且使得同一个网格单元中可能有若干个样本点和三角面片对,从而为后续的中轴点计算奠定基础。依据 5.1 节所述,模型上任意点的中轴点一定位于其法线上,对于三角面片来说,中轴点也一定位于由 3 个顶点围成的无线延伸的棱柱上。同理需要得到与三角面片相交的网格单元,思想与第一次法线迭代类似,这里不再赘述。

需要说明的是,判断样本点以及三角面片的法线是否与某一网格相交的准则均为:法线段/法线柱被当前范围 $[i * gl, (i+1) * gl]$ 切割后的柱段是否在某一网格单元坐标范围内。

各个三角面片的法线迭代也可以并行实现,为每个三角面片均分配一个 GPU,在当前迭代结束后,网格单元便会产生若干样本点以及三角面片,至此根据网格单元的数据对(由候选网格和候选样本点组成)可以计算中轴点,5.3 节将给出计算方法和具体计算过程。

5.3 中轴点计算

中轴计算从中轴定义出发进行中轴获得过程的描述。为了得到更精准的中轴,不同于以往的近似中轴的思想,本文方法从中轴的数学定义出发,中轴的数学定义为内切球的球心的集合。基于球的性质,那么球心到球边界的距离必定为

中轴半径 r , 即中轴点到模型边界的距离必定为中轴半径 r , 中轴计算即是基于此推理来计算中轴。离散之后的模型随着样本点数量的增多, 离散样本点拟合出的模型会更接近原模型, 因此模型中轴的精准性会随之提高。

法线迭代后, 每个网格单元中都会含有若干样本点和三角面片对, 基于所在面, 通过求解方程组的方式来求解中轴点, 样本点的内切球称为样本点的中轴球, 球心称为中轴点, 记作 M , 球的半径称为中轴半径, 记作 r 。中轴点即为内切球的球心, 而法线必过样本点与球心的连线, 由此, 便可推知有下列等式成立:

$$M = P + N_P * r = (M_X, M_Y, M_Z) \quad (4)$$

其中, M_X 是 M 在 X 轴的坐标, M_Y 是 M 在 Y 轴的坐标, M_Z 是 M 在 Z 轴的坐标。基于样本点所在面信息可表示 P 点坐标以及法线信息。由此, 只有 r 是未知的, 因此, 关键在于求得相应的 r , 将其代入(4)式即得 M 。

$$M_X = P_X + r * N_{P_X}$$

$$M_Y = P_Y + r * N_{P_Y}$$

$$M_Z = P_Z + r * N_{P_Z}$$

其中, P_X 是点 P 在 x 轴的坐标, P_Y 是点 P 在 y 轴的坐标, P_Z 是点 P 在 z 轴的坐标, $N_{P_X}, N_{P_Y}, N_{P_Z}$ 分别是点 P 的法线在 x 轴、 y 轴、 z 轴的坐标。一般地, 平面、曲面的一般方程分别可表示为式(5)和式(6):

$$A * x + B * y + C * z + D = 0 \quad (5)$$

$$F(x, y, z) = 0 \quad (6)$$

遍历模型过程中, 通过 opencascade(Coefficient 函数)可得前面的表示参数, 基于式(5)、式(6), 记点到平面的距离为 D_P , 点到曲面的距离为 D_F , 点到平面的距离为 D_P , 可以由式(7)求得。为了求得中轴点 M 到曲面的距离, 设曲面上一点的坐标为 (x_0, y_0, z_0) , 则中轴点 M 到平面和曲面的距离分别为:

$$D_P = \frac{|A * M_X + B * M_Y + C * M_Z + D|}{\sqrt{A^2 + B^2 + C^2}} \quad (7)$$

$$D_F = \sqrt{(M_X - x_0)^2 + (M_Y - y_0)^2 + (M_Z - z_0)^2} \quad (8)$$

为了求得 D_F , 引入了拉格朗日乘数的思想, 将式(8)作为目标函数, 根据拉格朗日乘子法有下式成立:

$$D_F + \lambda * F(x, y, z) = 0 \quad (9)$$

依次用式(9)对 x, y, z, λ 求偏导得:

$$\frac{\partial D_F}{\partial x} + \lambda \frac{\partial F}{\partial x} = 0 \quad (10)$$

$$\frac{\partial D_F}{\partial y} + \lambda \frac{\partial F}{\partial y} = 0 \quad (11)$$

$$\frac{\partial D_F}{\partial z} + \lambda \frac{\partial F}{\partial z} = 0 \quad (12)$$

$$F(x, y, z) = 0 \quad (13)$$

把 r 当作常数, 基于式(10)~式(13)可以求解出 $Q(x_0, y_0, z_0)$, 点 Q 为样本点 P 的对偶边界点。根据中轴的定义, 样本点和对偶边界点必定位于中轴球的不同位置, 由此可知中轴半径 r 与中轴点到对偶三角面片的距离必定相等, 将中轴定义经过简单转换, 则式(14)、式(15)成立:

$$r = D_P \quad (14)$$

$$r = D_F \quad (15)$$

式(14)、式(15)分别为关于 r 的一元一次和一元二次方程, 容易求得解。

若式(14)求解出的 r 值在当前范围内, 则认为该样本点的中轴点可在当前范围内求得, 否则将 P 进行下一轮迭代。若式(15)求解出的 r 值在当前范围内, 则认为该样本点的中轴点可在当前范围内求得; 若求解出的 r 为互不相等的两个值, 则将较小值作为有效 r , 这样处理是为了尽量控制后续参与计算的样本量, 减少不必要的时间消耗, 否则将 P 进行下一轮迭代。最后, 将求出的有效 r 代入式(4)即可得到在当前范围内的样本点 P 的中轴点 M 。

本文方法从中轴数学定义出发进行中轴计算, 避免了不必要的处理且基于数学推理结果较为严谨, 至此, 根据样本点的拓扑连通性对应样本点各自的中轴点, 即可连接中轴点得到模型中轴。

6 实验结果与分析

为了进一步验证算法的可行性及高效性, 进行了模型的模拟实验。实验平台是 Visual Studio 2010 和 opencascade 7.3.0, 处理器 i7-9750, 显卡为 GTX1660Ti, 使用 3dMax 构建基本模型, 使用 MeshLab 来渲染展示结果, 文件模型数据格式以“.obj”为后缀名, 图7~图10给出了4个模型使用本文所提算法进行实验所得中轴点结果。

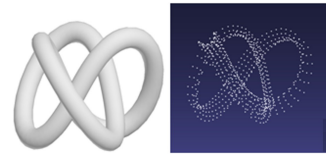


图7 模型 a 及其轴点

Fig. 7 Model a and its medial axis point



图8 模型 b 及其轴点

Fig. 8 Model b and its medial axis point

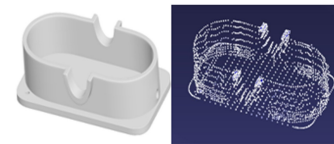


图9 模型 c

Fig. 9 Model c

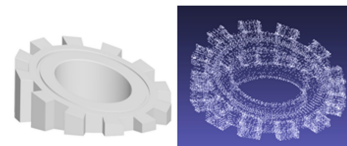


图10 模型 d

Fig. 10 Model d

基于上述实验结果, 对于不同模型, 所提算法均可以很好地计算得到其中轴, 由此验证了其有效性。

为了验证所提算法在时间性能上的提升, 首先将其与传统的方法进行了对比试验, 实验结果如表1所列。

表1 时间成本对比

Table 1 Time cost comparison

Model	Algorithm	Processor	Sample size	Time/ms
Model a	Thing	CPU	5 188	5 031
	Tracing	CPU	5 188	4 097
	Normal iteration	CPU	5 188	932
Model b	Thing	CPU	4 001	4 044
	Tracing	CPU	4 001	3 625
	Normal iteration	CPU	4 001	612
Model c	Thing	CPU	34 704	28 509
	Tracing	CPU	34 704	29 117
	Normal iteration	CPU	34 704	16 729
Model d	Thing	CPU	53 062	47 713
	Tracing	CPU	53 062	49 001
	Normal iteration	CPU	53 062	28 900

表2所列法线迭代算法分别在CPU、GPU环境下的时间对比,虽然理论上可以引入的GPU线程可以多达数百个,但GPU的特性只能应用于图形计算这种可以并行的任务。由于GPU线程运行速度实际上远远不如CPU,这里引入GPU之后的实际提升效率并不是很高,在7倍左右。

表2 法线迭代方法不同处理器对比

Table 2 Comparison of different processors for normal iteration method

Model	Processor	Load model	(单位:ms)	
			Generation of axis point	
Model a	CPU	182	893	
	GPU	182	141	
Model b	CPU	169	612	
	GPU	169	109	
Model c	CPU	306	16 729	
	GPU	306	951	
Model d	CPU	533	28 900	
	GPU	533	1 454	

为了进一步验证本文方法的双向并行法线迭代思想相比现有并行中轴方法对时间效率的提升,将其与代表性的并行模型中轴生成方法——中轴点逼近法、基于GPU框架法进行相同并行度下模型中轴的时间对比,以此来验证法线迭代方法对并行中轴生成方法的时间效率的提升。对比结果如表3所列。

表3 与高效方法的时间成本对比

Table 3 Time cost comparison with efficient methods

Model	Algorithm	Sample size	Degree of parallelism	Time/ms
Model a	Medial axis point approximation	5 188	768	335
	GPU-based framework method	5 188	768	208
	Normal iteration	5 188	768	141
Model b	Medial axis point approximation	4 001	768	234
	GPU-based framework method	4 001	768	147
	Normal iteration	4 001	768	109
Model c	Medial axis point approximation	34 704	768	2 923
	GPU-based framework method	34 704	768	1 341
	Normal iteration	34 704	768	951
Model d	Medial axis point approximation	53 062	768	3 729
	GPU-based framework method	53 062	768	2 166
	Normal iteration	53 062	768	1 454

由表3可知,相同条件下,法线迭代方法的时间成本相对

较低,而基于GPU框架法相比中轴点逼近法的效率提升在2倍左右,本文方法较GPU框架法的时间效率的提升在1.5倍左右,这主要是因为法线迭代方法引入了双向迭代并行机制,同时借助网格单元将中轴点的搜索范围限制在网格单元范围内,提高了中轴生成效率。

需要说明的是,因为每次的运行时间不会完全相同,为了保证结果的可靠性,表中时间结果均为多次运行结果的平均值取整之后的结果。

方法的精度指通过方法生成的中轴与符合中轴定义的结果偏差,而本文方法从中轴定义出发进行中轴计算,从根本上保证了所得中轴的质量。为了进行误差比较,将其量化为Hausdorff距离。Hausdorff距离是描述两组点集之间相似程度的一种量度,它是两个点集之间距离的一种定义形式,它度量了两个点集间的最大不匹配度。本文同样将Hausdorff距离作为误差逼近的度量。Hausdorff距离的计算时间代价较高,因此仅以模型a、c、d为例,根据模型上采样点的拓扑关系,对应连接相应的中轴点,结果借助MeshLab来渲染,如图11所示。

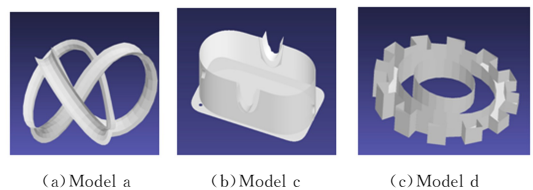


图11 法线迭代

Fig. 11 Normal iteration

将法线迭代方法与目前代表性的缩放因子(SAT)和基于最近邻点的中轴点逼近方法在同一样本条件下进行误差对比,以Voronio的结果为初始中轴,计算同一样本量不同方法的中轴与初始中轴的Hausdorff距离。为了方便比较,将结果进行归一化处理,这里使用模型包围盒对角线的长度为归一化标准,对比结果如表4所列。

表4 误差对比(以模型a、c、d为例)

Table 4 Error comparison(taking model a,c and d as examples)

Model	Algorithm	Sample size	Error
Model a	Normal iteration	5 188	0.0141
	SAT	5 188	0.0392
	Medial axis point approximation	5 188	0.0431
Model c	Normal iteration	34 704	0.0102
	SAT	34 704	0.0213
	Medial axis point approximation	34 704	0.0247
Model d	Normal iteration	53 062	0.0092
	SAT	53 062	0.0201
	Medial axis point approximation	53 062	0.0112

由表4可知,随着样本量的增多,3种方法的误差均呈降低趋势。同时可以看出,双法线跟踪方法相比SAT和中轴点逼近方法的精度提升大概在2~3倍。这是因为本文方法在中轴计算时从中轴的具体数学定义出发,所得到的中轴比较精准。

综上,可以看出,法线迭代方法生成的中轴既保持了原有基本拓扑关系,又大大降低了几何复杂性,能够得到有效的中轴。一般地,当样本点的数量足够多时,样本点会密集地分布

在模型表面。离散后的模型与原模型相似,用本文提出的中轴点计算方式进行处理,从中轴本身定义出发进行计算,从根本上保证了中轴质量的有效提升,因而其生成的中轴与真实中轴非常接近,由此可知本文方法生成的中轴在时间效率和中轴质量上均有所提升。

结束语 传统方法基于近似中轴的思想,中轴质量得不到保证。本文提出了一种基于中轴定义的法线迭代中轴生成方法。该方法首先将模型离散为三角网格模型,对网格模型进行 GPU 的并行法线迭代,然后使用中轴定义的衍生性质进行中轴计算,最终得到模型中轴。本文的主要贡献可以概括为:1)从数学角度出发,引入了法线迭代并将其与 GPU 并行思想进行融合,有效减少了对重复样本点的计算,提高了时间效率。2)利用网格单元思想将模型的中轴搜索范围缩小,减少了不必要的时间消耗。3)基于中轴的数学定义,利用中轴定义的衍生性质,提出了有效的中轴计算方法,提高了方法的精准性,从而有效提升了中轴的质量。

未来的工作将会关注:1)研究方法的时间成本与中轴质量之间的关系。2)将本文方法用于水下机器人路径规划方面的模拟,研究其规化出的路径安全性、健壮性等方面的性能。3)中轴本身存在不稳定性,易受噪音的影响,这使得中轴的应用代价较大,处理起来相对复杂,因此将在关注抗干扰能力的前提下研究新的中轴生成方法,寻求一种更稳定的类中轴的模型特征表示方式,拓展中轴的适用性。

参 考 文 献

- [1] LIANG Y, HU P, WANG S, et al. Medial axis extraction algorithm specializing in porous media [J]. Powder Technology, 2019, 343: 512-520.
- [2] MIAO Y W, CHENG C, SUN Y L, et al. Skeleton extraction of mesh model based on maximum inscribed sphere fitting [J]. Chinese Journal of Computer Aided Design and Graphics, 2018, 30(10): 1801-1809.
- [3] SPADAFORA J B, GOMEZ-FERNANDEZ F, TAUBIN G. Fast Non-Convex Hull Computation [C] // 2019 International Conference on 3D Vision (3DV). IEEE, 2019: 747-755.
- [4] MING L U O, CE H A H, HAFEEZ H M. Four-axis trochoidal toolpath planning for rough milling of aero-engine blisks [J]. Chinese Journal of Aeronautics, 2019, 32(8): 2009-2016.
- [5] CHANG J S. Representation and application of axis transformation for 3D deformable objects [D]. Ningbo: Ningbo University, 2017.
- [6] ZHONG Y J. Summary of axis extraction methods [J]. Journal of Computer Aided Design and Graphics, 2018, 30(8): 14-32.
- [7] BRUNNER D, BRUNETT G. Mesh segmentation using the object skeleton graph [EB/OL]. [2017-07-15]. <https://www.mendeley.com/research-papers/mesh-segmentation-using-object-keleto>.
- [8] YAN Y, SYKES K, CHAMBERS E, et al. Erosion thickness on medial axes of 3D shapes [J]. ACM Transactions on Graphics (TOG), 2016, 35(4): 1-12.
- [9] ZHANG F, CHEN X, ZHANG X. Parallel thinning and skeletonization algorithm based on cellular automaton [J]. Multimedia Tools and Applications, 2020, 79(43): 33215-33232.
- [10] ZHONG Y, CHEN F. Computing medial axis transformations of 2D point clouds [J]. Graphical Models, 2018, 97: 50-63.
- [11] KUSMAKAR S, MUTHUGANAPATHY R. Skeletal approach to mandible reconstruction represented as an image [J]. Computer-Aided Design and Applications, 2015, 12(5): 639-650.
- [12] MAKEM J E, FOGG H J, MUKHERJEE N. Medial Axis Based Bead Feature Recognition for Automotive Body Panel Meshing [C] // International Meshing Roundtable. Cham: Springer, 2018: 109-128.
- [13] SHI C W, ZHAO J Y, CHANG J S. Skeleton feature extraction algorithm based on central axis transformation [J]. Computer Engineering, 2019(7): 242-250.
- [14] CHEN X D, MA W. A competition flow method for computing medial axis transform [J]. Journal of Computational and Applied Mathematics, 2018, 340: 342-359.
- [15] SUN F, CHOI Y K, YU Y, et al. Medial meshes—a compact and accurate representation of medial axis transform [J]. IEEE Transactions on Visualization and Computer Graphics, 2015, 22(3): 1278-1290.
- [16] YAN Y, LETSCHAR D, JU T. Voxel cores: Efficient, robust, and provably good approximation of 3d medial axes [J]. ACM Transactions on Graphics (TOG), 2018, 37(4): 1-13.
- [17] WAGNER M G. Real-time thinning algorithms for 2D and 3D images using GPU processors [J]. Journal of Real-Time Image Processing, 2019: 1-12.
- [18] SAHA P K, BORGEFORS G, DI BAJA G S. A survey on skeletonization algorithms and their applications [J]. Pattern Recognition Letters, 2016, 76: 3-12.
- [19] REBAIN D, ANGLES B, VALENTIN J, et al. LSMAT least squares medial axis transform [J]. Computer Graphics Forum, 2019, 38(6): 5-18.
- [20] ZHU H S, BAO X S, ZHU C Y, et al. Parallel shape axis extraction algorithm based on binormal tracking [J]. Computer Engineering and Design, 2021, 42(1): 175-181.
- [21] LI Z. Mesh generation and planarization of quadrilateral building with free form surface [D]. Hangzhou: Zhejiang University, 2019.



ZONG Di-di, born in 1995, postgraduate. Her main research interests include computer-aided design and graphics.



XIE Yi-wu, born in 1965, associate professor. His main research interests include database and information system, and data mining.