

文章编号:1671-8879(2005)05-0038-05

基于改进约束 Delaunay 三角形 算法的公路虚拟环境建模

张景涛

(长安大学 特殊地区公路工程教育部重点实验室,陕西 西安 710064)

摘 要:针对目前公路设计缺乏公路 CAD 系统深层次支持现状,提出公路虚拟环境概念,以提高公路 CAD 系统对设计支持的深度和层次。对公路虚拟环境中的自然环境建模算法进行了研究,分析了基于约束 Delaunay 三角形(CDT)算法建模存在的问题,提出了改进约束 CDT 算法,建立的边结构解决了建模过程中拓扑更新问题。建立了基于删除操作的多层次细节模型简化算法。测试结果表明,该算法具有建模快、健壮性等优点,适合公路虚拟模型的建立。

关键词:道路工程;公路;虚拟环境;CDT 算法;模型;简化

中图分类号:U412

文献标识码:A

Highway virtual environmental model based on improved CDT algorithm

ZHANG Jing-tao

(Key Laboratory for Special Area Highway Engineering of Ministry
of Education, Chang'an University, Xi'an 710064, China)

Abstract: The current CAD systems lack of deep supporting design process. The concept of highway virtual environment is put forward to prompt the depth and level of design supporting. By studying the algorithm of environment model and analyzing the constraint Delaunay triangle algorithm, the improved CDT algorithm is developed. The edge structure is build to solve the topology problem. The level of detail model (LOD) is constructed to solve the model simplification problem. By testing, the algorithms are proved to be faster and stronger and can suit for highway environment model. 6 figs, 6 refs.

Key words: road engineering; highway; virtual environment; CDT algorithm; model; simplification

0 引 言

公路路线设计是集决策性、知识性和实践性为一体的半结构化问题。其影响因素有二维的,也有三维的;有定量的指标,也有定性的指标;有确定性因素,也有不确定性因素;而且各因素之间的关系错综复杂。选线设计的过程就是在自然条件、技术条

件和人所构成的环境中,综合分析和协调各种影响因素,采用特定的方法进行路线设计的创造性思维过程。

目前公路计算机辅助设计(CAD)系统已在中国公路设计领域得到了广泛的应用,但这些系统主要解决在计算和绘图等层次上对设计人员的支持,而把需要经验或知识决策的设计问题留给设计人

员,因而对设计的支持层次较低。由于路线在空间位置的确定依赖于设计人员对自然条件的分析,所以选线设计比其他许多工程设计对环境具有更强的依赖性。为了提高设计质量,提高对设计过程支持的层次和深度,20世纪50年代以来,许多专家学者就开始探讨公路CAD设计理论和模型^[1~3]。但无论那种方法,其本质在于如何将影响选线设计的因素表达成计算机能够识别的形式,并用计算机仿真选线设计方法为设计者提供决策支持。

因此可以说,由计算机、设计模型、应用程序共同构成了一个辅助设计环境,称为公路虚拟环境模型,具体包括虚拟自然环境和智能决策环境。虚拟自然环境是自然环境的计算机再现,包括地形、地质、水文等信息;而智能决策环境是辅助路线工程师进行决策的优化模型、设计模型及其实现程序。建立公路虚拟环境模型,可以将公路设计过程信息进行有机的集成,从而提高公路CAD系统对设计支持的层次和深度。公路虚拟环境模型涉及大量的子模型和算法,本文主要研究公路虚拟自然环境模型及其建模算法^[1]。

1 有关 CDT 算法的研究与分析

在虚拟环境建模中,不规则三角网(triangulated irregular network,简记为 TIN)由于能够通过从不规则分布的数据点生成的连续的三角面来逼近地形表面,因此得到广泛的应用。标准的 Delaunay 三角网是满足无任何约束条件的散点域的三角形剖分,而在实际应用中,部分散点之间常常存在着某种约束关系,如道路边界、山脊线、山谷线、断裂线等。在对此种数据域进行三角形剖分时,三角形剖分的结果应保持原有的约束关系,此即约束数据域下的三角剖分,所形成的三角网称为约束 Delaunay 三角网(constrained delaunay triangulation,简记为 CDT)。由于 CDT 能够有效的处理地形约束的问题,因此在自然环境建模中具有无可比拟的优势。

在标准 Delaunay 三角网建模方面,目前基本算法为逐点插入算法、分而治之算法和三角网生长算法。

从时间复杂度看,分而治之算法最好,其缺点也是显而易见的,表现在:

(1)需要占用较大的内存空间。其原因主要是分治算法需一次性读入所有参加构网的数据。

(2)数据预处理工作量较大。由于在构建三角网前要对数据进行排序,故分治算法的预处理工作

量较大。

(3)优化工作量较大。

逐点插入算法实现比较简单,占用内存小,但时间复杂度较差,主要制约因素在于点在三角形中的判断、LOP 局部优化过程的拓扑更新以及空外接圆检测。

三角网生成算法速度最慢,目前已很少使用,但刘学军提出的 TBP 算法可以提高算法的效率,但并未得到广泛的应用^[2,3]。

目前研究较多的算法是合成算法,即将逐点插入法与分而治之法融合,即对数据进行分块,子网建立采用逐点插入法,子网间合并采用凸包合并算法,从而解决二者不足。

实际上,逐点插入法在解决了以上问题后,其速度基本与合成算法接近。合成算法较为复杂,实现比较困难,因此,本文在标准 Delaunay 三角网建立上采用逐点插入算法。

综合有关文献,CDT 的构造算法可归纳为

(1)约束图法。Lee 和 Lin 在 1986 年提出了基于约束图计算 CDT 构造算法。他们首先用 $O(N^2)$ 的时间计算约束数据域的可见性图,然后用 $O(N^2)$ 的时间检测优化可见性图并形成 CDT。

(2)分割-合并算法。Lee 和 Schachter 在 1981 年提出了非约束数据域 Delaunay 三角剖分的分割-合并算法,Chew 将其推广至约束数据域,Joe 和 Wang 也提出了相类似的算法。分割-合并算法虽执行效率较高($O(N\log N)$),但约束线段的存在使得分割过程的实现比较困难。

(3)加密算法。Boissonnat 在进行 CDT 构造的研究中,将约束数据转换为非约束数据,即对约束线段按空外接圆规则进行加密,然后对加密后的数据进行非约束数据域的 Delaunay 三角剖分。Boissonnat 认为加密过程的时间复杂度为 $O(N\log N)$,虽然该算法简单易实现,但加密点的存在不但加大了数据量,而且改变了原来的数据集。

(4)Shell 三角化算法。该算法是 Piegl 和 Richard 在 1993 年提出的,其实质是非约束数据域的三角网生长算法的推广。算法由任一约束线段开始,以可见性和空外接圆特性为准则寻找可扩展的第三点。该算法关键是要解决好第三点的查询问题,以提高算法效率。

(5)两步法。这是目前采用较多的一种方法,即首先对约束数据集建立非约束 Delaunay 三角网(初始三角网),然后引入约束线段以嵌入之(调整过

程)。这方面的算法以 Bernal、Sloan 和 Floriani 为代表。Bernal 和 Sloan 的算法在初始三角网过程中不区分约束与非约束点,而在调整过程中使之满足约束线段条件;Floriani 的算法更具灵活性,他只建立非约束数据点的初始三角网,然后在初始三角网中插入约束线段。Sloan 利用连续的对角线交换法实现约束线段的嵌入,要求可交换对角线的四边形是严格凸的,而这一点在实际应用中较难满足。Floriani 的算法核心是简单多边形的 Delaunay 三角剖分,但其实现过程是递归执行,递归过程虽执行效率较高,但内存开销较大。

目前算法多基于两步法,如刘学军提出的约束线段嵌入的迭代算法, Marc Vigo 提出的约束线段嵌入的递归算法。刘学军算法比较简单,算法效率较高,但没有考虑约束边界存在重叠的情况。Marc Vigo 算法是递归算法,考虑到了边界重叠情况,执行效率较高,内存开销较大。

综合以上情况,本文采用两步法进行建模,初始三角网采用逐点插入算法,约束线段的插入则采用局部递归算法,并设计边结构来维护拓扑关系。

2 两步法建立 CDT 的基本步骤

2.1 建模思路

设当前处理的约束线段(有向)为 $P_1, P_2, \dots, P_N (N \geq 2)$, 在一已存在的 CDT 中加入该连续线段,这一过程可分为两步完成,即:

- (1) 在 CDT 中加入约束线段顶点 P_1, P_2, \dots, P_N ;
- (2) 对三角网进行局部调整以满足 $P_i P_j (i, j < N)$ 的可见性及空外接圆法则。

利用逐点插入算法,可完成步骤(1)。但需要注意的是,首先要保证 P_i 点不重合,如重合就不插入。算法如下:剔除坐标一致或相近的点;设当前点为 P_i , D 为给定的距离阈值, G 为 p_i 所在的网格,依次提取网格内的其他点,若有下式成立,则 P_i 点应赋值为 P_i 点。

$$\text{dist}(p_i, p_j) \leq D, p_i, p_j \in G, i=1, 2, \dots, n$$

对于步骤(2),基本思路为

设当前处理的线段为 $P_i P_j$, 由与 $P_i P_j$ 相交的三角形组成的区域称为 $P_i P_j$ 的三角形影响域 $MT = \{T_1, T_2, \dots, T_N\}$, 而由 MT 中三角形的外围边组成的多边形称为影响多边形 $Q = \{P_i, 1, 2, \dots, P_j, \dots, P_i\}$ 。线段 $P_i P_j$ 将 Q 分为 QL 和 QR 左右两部分,如图 1 所示,可分别对 QL 和 QR 进行三角形剖

分和优化。

注意到影响域 Q 并不是严格意义上的多边形,影响域边界可能会包含重复的顶点(图 2)。

在这种情况下,对于悬边两侧的三角形都需要重建,不像一般的边界边,只需要一边重建即可。这种情况的存在并不影响算法的整个过程,但在建立三角形拓扑关系的时候需要小心处理。

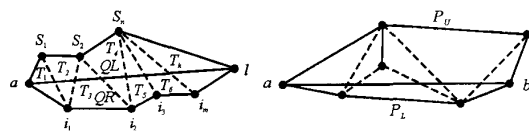


图 1 影响域

图 2 边界边重复的多边形

2.2 边插入算法描述

按照上述基本思路,边插入算法可描述为

- (1) 由包含 P_i 或 P_j 的任一三角形开始,在 CDT 中查找与 $P_i P_j$ 相交的三角形,形成三角形影响域 $MT = \{T_1, T_2, \dots, T_N\}$ 。

- (2) 如三角形影响域为 $MT = \{T_1\}$, 则 $P_i P_j$ 为 CDT 三角形中一条边,则退出算法。如 $P_i P_j$ 与三角形的交点除 P_i, P_j 外,还有一点 P_k , 则将 $P_i P_j$ 分割为 $P_i P_k$ 和 $P_j P_k$, 按边插入算法,重新进行。

- (3) 由 MT 中的三角形组成 $P_i P_j$ 的影响多边形,并把 Q 以 $P_i P_j$ 为边界分成 QL 和 QR 2 个多边形域(注意组成影响域多边形的时候,悬边需要重复计入)。

- (4) 对 QL 和 QR 做如下三角形剖分(以 QL 为例): ① 如 QL 点集 P 不空,以 $P_i P_j$ 为基边,按空外接圆法则或最大最小角法则,在 QL 点集 P 中寻找合适的点 C , 构成三角形 $P_i C P_j$; ② 三角形将 P 划分为 PE 和 PD , 如图 3 所示。以 $P_i C$ 和 $C P_j$ 为基边,

图 3 迭代算法图示

执行步骤 ①,直至点集为空为止。

- (5) 对三角形拓扑关系进行更新。

3 CDT 算法的实现与拓扑关系的维护

3.1 边结构设计

为了维护三角网重构后的拓扑关系,本文设计了边结构,以辅助维护拓扑关系,描述为

```
struct EDGE{
    int BPNuM; //边起点;
    int EPNuM; //边终点;
    int LTriNuM; //左侧三角形号;
    int RTriNuM; //右侧三角形号;
    int LTriPos; //左侧三角形在右侧三角形中拓
```

扑位置;

int RTriPos; //右侧三角形在左侧三角形中拓扑位置;

int InfoFlag; //边信息更新标志,初值为-1,如左侧更新则为1,右侧更新为0,对悬边,在查找影响域时,并构造边数组时,令边信息更新标志为-1;

int EdgeInfo; //悬边标识,当为0时,非悬边,为其他值时则是悬边;

}

另外,实际上不必分左右,因为通过两侧三角形号和位置,就可以对边两侧的三角形拓扑关系进行更新。

3.2 影响三角形的确定

(1) 由 P_i, P_j 的点号,可以很快的找到 P_i, P_j 所在的三角形,设 P_i 所在的三角形为 t 。

(2) 由三角形的拓扑信息,由 t 开始以 P_i 为顶点逆时针方向寻找到三角形 T_1 。如图4(a)所示。

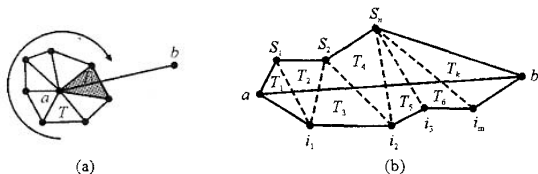


图4 影响域的确定

(3) 从 T_1 开始,由三角形拓扑信息依次找到影响域三角形 T_2, T_3, \dots, T_k ,存入影响域三角形数组中,同时生成 QL 和 QR 的影响域边界数组,其边界点具有拓扑性,如上所述。同时构造边结构数组,对于悬边,需进行标识,即

EDGE E[];

If(E[i-1].BPNUM=E[i].EPNUM &&

E[i-1].EPNUM=E[i].BPNUM) then

E[i-1].EdgeInfo=i-1;

E[i].EdgeInfo=i-1;

EndIf

delete E[i]; //不需要重复记录。

QL, QR 的确定可根据下述公式确定

如果 $x_{p_j} \neq x_{p_i}$ 则 $F(G, G \in Q) = y_G - Ax_G - B$

如果 $x_{p_j} = x_{p_i}$ 则 $F(G, G \in Q) = x_{p_i} - x_G$

其中, $A = (y_{p_j} - y_{p_i}) / (x_{p_j} - x_{p_i})$

$B = (y_{p_j}x_{p_i} - x_{p_i}y_{p_j}) / (x_{p_j} - x_{p_i})$

如果 $F(G) < 0$, 则 $G \in QL$; 如果 $F(G) \geq 0$, 则 $G \in QR$ 。

4 局部三角形重构的递归算法

伪代码描述为

Procedure TriangulatePseudopolygonDelaunay
(P:VertexList, ab, Edge, T:CDT)

If P has more than one element then //如果 P 不空;

c := First vertex of P //设 c 为 P 中第一点;

//对 P 中每一点做空外接圆检测,直到找到一点 c, $\triangle abc$ 中不包含其他点。

For each vertex v \in P do

If v \in Circumcircle(a, b, c) then

c := v

EndIf

EndFor

//将 P 分为点集 PE 和点集 PD, 其中 $P = PE + \{c\} + PD$;

TriangulatePseudopolygonDelaunay(PE, ac,

T) //对 bc 和 ca 做递归调用

TriangulatePseudopolygonDelaunay(PD, cb, T)

EndIf

//如果 P 不空,从影响域堆栈中弹出三角形空间,将 a, b, c 信息读入;

//判断 ab 边的更新记录,如为-1,则更新 ab 边左侧信息 LTriNum, RTriPos;

//否则更新右侧信息。

If P is not empty then

t = Pop(stack);

RefreshTri(t, a, b, c);

RefreshEdge(ab);

EndIf

EndProc

三角形局部重构完毕后,由于构造了边数组,因此其拓扑信息都已写入边数组中,拓扑更新的工作,就是遍历所有的边,将边中拓扑信息写入相应的三角形中即可。

5 LOD 模型简化算法

层次细节模型是 Clark 在 20 世纪 70 年代提出的,是一种模型简化技术,是目前模型简化方法中研究最热的一个领域。LOD 模型又称多分辨率模型,由一组对原始对象进行不同精度逼近的模型组成。在应用中,根据对象和视点的距离,选择合适的分辨率模型来进行绘制^[4~6]。

表面模型的简化中,生成 LOD 模型的主要方法为细分方法和删减方法。

细分方法从一个由很小的数据集构成的简单三

角形网格开始,逐步添加进一些符合某种规则的点,从而改善分辨率,直到获得需要的层次细节。最典型的是称为贪婪插入的算法。

表面模型简化算法的主体是删减方法。删减方法从包含整个数据的原始三角形网格开始,逐步地删除那些符合一定判别规则的顶点、边或者面,直至粗化到设定的细节水平为止。改变网格内几何实体数量的局部删除操作可分为点删除、边折叠、点对合并操作等。

本文使用的是基于三角形移去的网格简化算法,描述为

(1)对三角形网格 TM 中的每个三角形 T_i ,求与 T_i 相关的三角形板 P_i 。

(2)求出三角形板 P_i 的平均平面,计算三角形板 P_i 中所有顶点到平均平面的距离,然后求出评价距离 d 。

(3)若平均距离 d 小于指定的简化标准阈值,那么置该三角形的移去标志为真;否则,置该三角形的移去标志为假。

(4)对移去标志为真的三角形 T_i ,根据其相关三角形板 P_i 求出其相应的边界多边形 PBP_i ,对其进行带约束的局部三角形优化。

(5)若三角形网格中,三角形的数目仍大于所要删减到的数目,可以相应地调整简化标准阈值,重复以上步骤,直到终止条件满足为止。

利用该算法对某地形进行模型简化的结果如图 5 所示。利用上述算法生成的公路虚拟自然环境如图 6 所示。

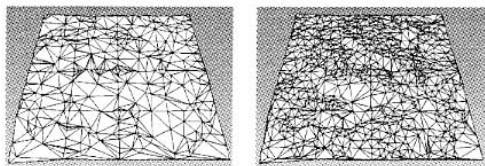


图 5 多层次细节模型

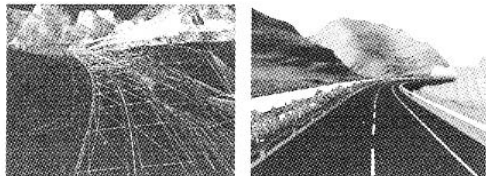


图 6 公路虚拟自然环境模型

6 结 语

(1)本文提出了公路虚拟环境的概念,对建立公路虚拟自然环境的 CDT 算法进行了回顾,完善了两步法构造 CDT 的算法,并提出边结构以实现拓扑关系的快速维护。

(2)建立的基于删除操作的 LOD 简化算法,可以快速进行模型简化。

参考文献:

References:

- [1] 易思蓉. 虚拟环境铁路选线设计系统的理论与方法研究[D]. 成都: 西南交通大学, 2000.
YI Si-rong. Study on the theories and methods of virtual environment railway location system[D]. Chengdu: Southwest Jiaotong University, 2000.
- [2] 刘学军, 符铎砂. TIN 数模的点单位算法及网形优化[J]. 中国公路学报, 1997, 10(2): 24-31.
LIU Xue-jun, FU Xin-sha. An algorithm of triangulation based points for constructing TIN and it's adjusting on feature-line[J]. China Journal of Highway and Transport, 1997, 10(2): 24-31.
- [3] 朱 庆, 陈楚江. 不规则三角网的快速建立及动态更新[J]. 武汉测绘大学学报, 1998, 23(3): 204-207.
ZHU Qing, CHEN Chu-jiang. Quick generation of TIN and its dynamic updating[J]. Journal of Wuhan Technical University of Surveying and Mapping, 1998, 23(3): 204-207.
- [4] 蒲 浩, 宋占峰, 郑顺义, 詹振炎. 道路三维场景的实时动态显示技术[J]. 交通运输工程学报, 2003, 3(1): 52-56.
PU Hao, SONG Zhan-feng, ZHENG Shun-yi, ZHAN Zhen-yan. Real-time dynamic display technology of road 3D scene[J]. Journal of Traffic and Transportation Engineering, 2003, 3(1): 52-56.
- [5] 许金良, 杨宏志. 公路视景仿真模型[J]. 长安大学学报(自然科学版), 2004, 24(2): 37-40.
XU Jin-liang, YANG Hong-zhi. Highway scene simulation model[J]. Journal of Chang'an University (Natural Science Edition), 2004, 24(2): 37-40.
- [6] 杨宏志, 许金良, 李建士. 基于计算机仿真的公路线形评价[J]. 中国公路学报, 2005, 18(1): 14-17.
YANG Hong-zhi, XU Jin-liang, LI Jian-shi. Evaluation of highway route design based on computer simulation[J]. China Journal of Highway and Transport, 2005, 18(1): 14-17.