

# 空间三角网格曲面的边界提取方法

张献颖 周明全 耿国华

(西北大学计算机科学系, 西安 710069)

**摘要** 边界是曲面的重要特征之一, 在形状匹配、曲面拼接等方面有着重要作用。由于空间网格数据分布不规则, 不能从点的坐标直接得到曲面的边界, 因此提出了一种空间三角网格曲面的边界提取方法, 通过判断一个点的邻接点是否都能通过三角网格的边组成闭合曲线来获取边界点。该方法易于实现、适应性强, 并应用于计算机辅助文物复原系统, 实验证明, 此算法能正确处理空间三角网格数据, 且效率很高。

**关键词** 计算机图形学(520·6030) 边界提取 三角网格数据 边界曲线 邻接点 边界点  
**中图分类号**: TP391.41 **文献标识码**: A **文章编号**: 1006-8961(2003)10-1223-04

## A Method of Detecting the Edge of Triangular Mesh Surface

ZHANG Xian-ying, ZHOU Ming-quan, GENG Guo-hua

(Department of Computer Science, Northwest University, Xi'an 710069)

**Abstract** Edge contour is an important feature of surface, which plays a special role in shape matching, surface conjunction. Due to the irregular distribution of space mesh data, the contour of surface cannot be gained from coordinate of points directly. So a method to detect edges of triangular mesh surface is brought out in this passage. If the adjacent points of the certain point can be formed a closed curve form ridges of triangular mesh, this point can be determined to a edge point. Scan the adjacent edge points of every edge point to get the sequence of edge points. This method has the advantage of simple and easy to implement. Which has been employed in the computer aided cultural relic restoration system, and has been proved to process tri-dimension data correctly and efficiently. This method is universal, which can also be applied to other mesh data.

**Keywords** Computer graphic, Edge detection, Triangular mash surface, Edge contour, Adjacent point, Edge point

## 0 引言

曲面网格数据是在曲面网格的网格顶点进行数值采样得到的离散数据, 分布在曲面上的三维曲面网格数据, 由数值数据与几何数据两部分组成<sup>[1]</sup>。几何数据即曲面网格, 它决定数据分布的曲面形状及数据采样形式; 数值数据决定曲面上属性值的分布特点<sup>[2]</sup>。网格曲面事实上是数据采样曲面的离散剖分结果, 在实际应用中, 连续曲面一般被剖分为网格, 因此曲面采样数据多为曲面网格数据。空间网格数据是一种非常重要的数据格式, 在科学计算可视化、计算机辅助几何设计、三维建模和重构、有限元

计算等领域有着广泛的应用<sup>[3]</sup>。由于曲面网格数据分布在曲面上, 在 X-Y-Z 坐标空间不规则分布, 而且数据采样空间较大, 因此, 图象处理中大多数特征分析方法无法在曲面网格数据分析中应用, 文献[1]中讨论了对曲面网格数据中特征区域边缘的自动提取与显示方法。但在某些时候, 例如在曲面的形状匹配、曲面拼接时, 关心的只是曲面的外边界轮廓, 甚至曲面网格数据中不包含数值数据, 因此, 提出了网格曲面边界轮廓提取方法。

## 1 边界提取算法

在对网格曲面进行形状匹配和拼接时, 不但要

基金项目: 国家自然科学基金(60072044)

收稿日期: 2002-11-11; 改回日期: 2003-03-30

得到曲面的所有边界点,而且还要使这些边界点按照某个方向顺次排列,依次连接构成一条或多条闭合的曲线.而边界点的判断是整个算法的基础.

### 1.1 边界点的判断方法

由于曲面三角网格数据分布在曲面上,而且采样不规则,因此首先要在三角网格数据中定义邻接三角网格和邻接点.对于网格数据中的点,如果它是某一个三角网格的一个顶点,则称这个三角网格是这个点的邻接三角网格;如果网格数据中的两个点的连接线段为某一三角网格的边,则称这两个点为邻接点.曲面上一点 $P$ 的所有邻接三角网格构成 $P$ 的邻接三角网格集合 $T_P = \{T | P \in T, T \text{ 是曲面上的一个三角网格}\}$ ;  $P$ 的所有邻接点构成 $P$ 的邻接点集合 $V_P = \{Q | T, P \in T \ \&\& \ Q \in T, Q \text{ 是曲面上异于 } P \text{ 的点}\}$ .

对 $P$ 的判断可通过点 $P$ 的邻接点集合 $V_P$ 来判断.如果 $V_P$ 中的点都能通过三角网格的边相连组成闭合的曲线,则点 $P$ 为内点,否则为边界点.如图1所示, $V_H = \{A, I, J, E, F, G\}$ ,因这些点通过三角形的边 $AI, IJ, JE, EF, FG, GA$ 组成闭合的曲线,所以点 $H$ 为内点,不是边界点;而 $V_A = \{B, I, H, G\}$ ,因点 $B$ 和 $G$ 不相邻,所以点 $A$ 为边界点.

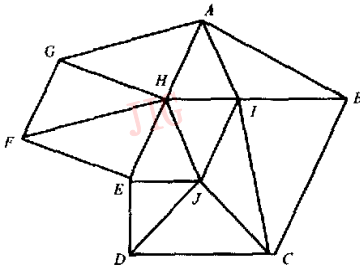


图1 空间三角网格曲面

在实际应用中,由于空间曲面上的点分布不规则,对点进行排序比较困难,因此用点的邻接三角网格集合来对点进行判断.基本步骤如下:

①建立一个临时点数组 $Temp$ ,并置为空.

②对当前点的所有邻接三角网格上异于当前点的点作如下操作:如果 $Temp$ 包含此点,则从 $Temp$ 中删除此点,否则,把此点加入到 $Temp$ 中.

③检查 $Temp$ ,如果 $Temp$ 为空,则当前点为内点,否则为边界点,并且, $Temp$ 中的点是和当前点的邻接边界点.

### 1.2 边界提取算法描述

在一般情况下,空间曲面上点的存储是随机的,所以得到的所有的边界点也是无序的,前面已经提到,对曲面上的点进行边界点判断的时候,可以得到与当前边界点的邻接边界点,利用上面的边界点的判断方法,对空间曲面上的点依次作边界点的判断,得到与当前边界点的邻接边界点.基于此,即可对所有边界点排序,得到完整的边界曲线.

在完整描述算法之前,先定义算法中要用到的数据结构.设空间曲面上有 $N$ 个点,组成 $M$ 个三角网格.点的存储结构为

```
struct Point
{
    double x, y, z;
    int * NeighborTriangles;
    int NumAdjPoint;
    int * NeighborPoints
}Points[N]
```

其中, $x, y, z$ 存放点的3个坐标值, $NeighborTriangles$ 指向一个整数数组,这个数组存放这个点的邻接三角网格的索引值, $NumAdjPoint$ 存放这个点的邻接边界点的个数, $NeighborPoints$ 指向一个整数数组,这个数组存放这个点的邻接边界点的索引值,结构体数组 $Points[N]$ 存放所有的点.三角网格的存储结构为

```
struct Triangle
{
    int A, B, C;
}Triangles[M]
```

其中, $A, B, C$ 分别存放三角网格的3个顶点的索引值,结构体数组 $Triangles[M]$ 存放所有的三角网格.

边界提取算法的完整步骤如下:

(1)对曲面上的每个点建立它的邻接三角网格链表 $NeighborTriangles$ .具体做法是遍历 $Triangles[M]$ ,把当前三角网格在数组中的索引值分别加入到3个顶点的邻接三角网格链表中.

(2)对曲面上的每个点作边界点判断.用1.1节中边界点判断方法对每个点作判断, $NumAdjPoint$ 纪录 $Temp$ 中点的个数,如果当前点为内点,则 $NumAdjPoint$ 为0,数组 $NeighborPoints$ 存放 $Temp$ 中的点的索引值.

(3)从 $Points$ 中选取 $NumAdjPoint$ 值最大的一个点 $Points[i]$ 作为当前点,新建一个边界曲线,把此

点加入曲线作为起点,继续搜索此边界上的其他点,如果  $\text{Points}[N].\text{NumAdjPoint}$  均为 0,则转到第 6 步。

(4) 从当前点的  $\text{NeighborPoints}$  中取出一一点  $\text{Points}[j]$ ,并在  $\text{NeighborPoints}$  中把它删除,当前点的  $\text{NumAdjPoint}$  减 1。

(5) 从  $\text{Points}[j]$  的  $\text{NeighborPoints}$  中找出当前点的索引,将其删除,  $\text{Points}[j].\text{NumAdjPoint}$  减 1。如果  $\text{Points}[j]$  不等于边界曲线的起点,则把它加入当前的边界曲线中,并把它作为当前点,转到第 4 步,否则转到第 3 步。

(6) 结束。

### 1.3 边界提取算法的简化

边界提取算法有很好的适应性,对于曲面有多条边界线以及曲面上有洞的情况都能做到很好的处理。

本文所述的边界提取算法可能会出现图 2 所示的两种不理想情况,一种是在对形如  $A, C, G, H$  4 点进行剖分时出现的不理想情况,理想的剖分应为  $ACH$  和  $CGH$  两个三角形,或者  $ACG$  和  $AGH$  两个三角形,但剖分结果却为图 2 中所示的  $ACH, CGH$  和  $ACG$  3 个三角形。这时  $C$  也成为边界点,点  $A, C, G$  组成一条边界线,然后再对曲线处理;另外一种是一个或多个三角网格通过一个顶点和别的三角网格相连的不理想情况,如点  $F$  在边界处出现,所述的边界提取算法若从  $F$  点出发开始搜索边界,就会提取出多条边界线。为此,我们提出如下判别规则:任何一个点的邻接点只能属于这个点的 1 个或 2 个邻接三角网格,任何一个边界点的邻接边界点只有 2 个。按此判别规则,就可避免出现两种不理想的曲面剖分,由于我们在实际处理三维扫描的数据时,发现这种不理想的剖分还大量存在,故采用该判别规则,使得边界提取算法得到简化,效率大大提高。

具体做法是在对点进行边界点判断时即完成排

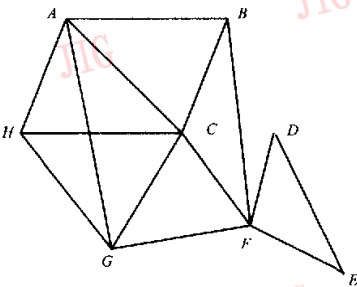


图 2 含有错误剖分的空间三角网格曲面

序,如果能进一步判断知道曲面只有一条边界线,则可不必对所有的点作判断。点的存储结构可改为

```
struct Point
{
    double x, y, z;
    int * NeighborTriangles;
    bool checked;
}Points[N]
```

其他变量意义不变,布尔型变量  $\text{checked}$  记录当前点是否作过判断,如果曲面只有一条边界,这个变量也不需要。简化算法的具体步骤如下:

(1) 对表面上的每个点建立它的邻接三角网格链表  $\text{NeighborTriangles}$ ,具体方法如前所述,把每个点  $\text{checked}$  变量初始化为假。

(2) 点数组中  $\text{checked}$  为假的点依次作边界点判断,直到遇到一个边界点,把此点的  $\text{checked}$  置为真,新建一个边界曲线,把此点加入曲线作起点,把它的一个邻接边界点当作当前点,继续搜索此边界上的其他点,否则转到第 4 步。

(3) 如果当前点等于边界曲线的起点,则转到第 2 步。对当前点进行作边界点判断,把当前点的  $\text{checked}$  置为真,把当前点加入曲线中,把异于曲线上上一个点的当前点邻接边界点作为当前点,继续执行这一步骤。

(4) 结束。

如果只有一条边界,则第 2 步只需执行一次,所以一般情况下大部分的曲面内点都不必作边界点判断,而在曲面上,边界点在所有点中占的比例比较小,所以效率也会明显提高。

## 2 结论

空间三角网格曲面的边界提取方法,通过判断一个点的邻接点是否都能通过三角网格的边组成闭合曲线来获取边界点,并在判断边界点的同时完成对边界点的排序,获得边界曲线。该方法已应用于计算机辅助文物复原系统,图 3 是部分实验结果图。

在图 3(a)中,由于手柄部分扫描不全,曲面上留下一个洞,试验的结果得到两条边界,右上角向里突出一块,边界也相应地突出一段,试验结果与理论完全一致。该算法的思想,也可用于处理其他网格类型的空间曲面。实验证明该算法能正确处理空间三角网格数据,而且效率很高。

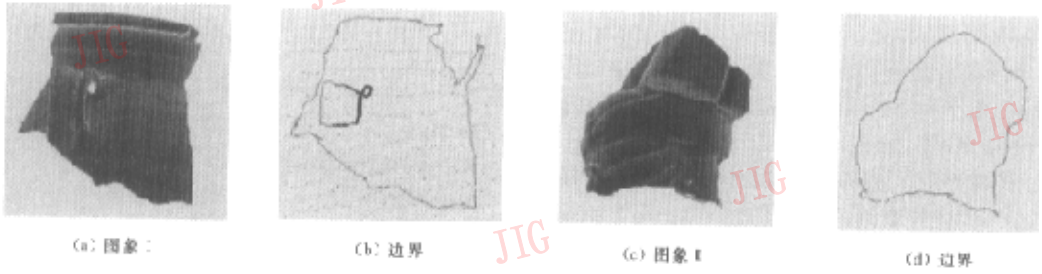


图3 边界提取实验结果图

参 考 文 献

- 1 Hagen H, Muller H. Focus on scientific visualization[M]. New York: Springer-Verlag, 1993:73~91.
- 2 谈正, 王利生. 曲面网格数据的边缘提取与显示[J]. 计算机辅助设计与图形学学报, 2000, 12(8): 580~584.
- 3 Moorhead R J, Zhu Z F. Signal processing aspects of scientific visualization [J]. IEEE Signal Processing Magazine, 1995, 12(5): 20~41.



**张献颖** 1978年生, 现为西北大学可视化研究所硕士研究生. 主要研究领域为图形图像处理 and 模式识别.



**周明全** 1954年生, 现为西北大学计算机科学系教授, 博士生导师. 主要从事图形图像处理与可视化技术等方面的教学科研工作. 主持完成了国家自然科学基金、“863”等多项相关课题.



**耿国华** 1955年生, 现为西北大学计算机科学系教授, 博士生导师. 主要从事模式识别, 工程数据库等方面的教学科研工作, 发表科研论文60余篇, 出版教材4部, 学术专著1部.