

# 空间网格面片可见边界提取方法研究与应用

郭爱斌 邓家禔

(北京航空航天大学机械工程及自动化学院, 北京 100083)

**摘要** 在计算机视觉和图形学中广泛应用三角网格来表示空间曲面,在许多应用领域基于边缘检测的2维图像处理技术已非常成熟,但空间三角网格面片边界提取,特别是沿某一方向可见边界提取方法还没有得到很好发展。本文提出一种求解三角网格面片可见边界算法,该算法根据光线方向和三角网格对应的法向量判断三角网格的可见性,从而确定三角网格面片的可见边界,再由各个三角网格具有可见边界的数目标识三角网格的类型,接着依据三角网格类型确定面片的可见边界,该算法的时间复杂度为 $O(n)$ 。最后以实例说明了该算法的可行性和有效性,并将该算法应用于卫星构型分析中的太阳翼遮挡分析计算模块,其计算速度、精度能满足工程实践要求。

**关键词** 网格面片 可见边界提取 遮挡分析

中图分类号: TP391.41 文献标识码: A 文章编号: 1006-8961(2007)11-2119-05

## Research & Application of a Visible Mesh Border Detection Algorithm

GUO Ai-bin, DENG Jia-ti

(School of Mechanical Engineering and Automation, Beijing University of Aeronautics and Astronautics, Beijing 100083)

**Abstract** Triangle meshes are widely used in representing surfaces in computer vision and computer graphics. Although a lot of work related to 2D image processing based on edge extraction techniques has been done in many applications, it is not well developed to extract the border of surfaces represented by triangle meshes, especially the visible border along one vector. A detection algorithm of visible mesh borders is presented in the paper. The visibility of a triangle facet in the mesh is determined by the relation between its normal vector and the ray direction. The type of a triangle facet is defined by the number of its visible edges, and it includes 0-type, 1-type, 2-type and 3-type. The visible border is detected according the triangles' type. The algorithm's time complexity is  $O(n)$ . Experimental results demonstrated the feasibility and validity of the algorithm. This algorithm is applied in the shelter calculation for the solar arrays of the satellite configuration analysis. And the requirements on computing velocity and accuracy for engineering application are met completely.

**Keywords** mesh surface, visible border extraction, shelter analysis

## 1 引言

三角网格是表示3维物体常见的数据格式,许多已有的面片重构算法的输出结果都是三角网格面片<sup>[1]</sup>,并且大多CAD软件也都可以把形体的几何信息输出为三角网格面片。曲面网格数据是对曲面网格的网格顶点进行数值采样得到的离散数据<sup>[2]</sup>。而曲面网格数据是3维网格数据,数据分布在曲面上,

由数值数据与几何数据两部分组成<sup>[3]</sup>。文献[2]、[4]中给出了三角网格面片的边界提取方法,但对于连续封闭曲面,如网格化的球面是没有边界的,另外沿某一矢量的可见边界与曲面的实际边界往往并不一致。有些工程问题分析和计算需要提取网格曲面沿某一矢量的可见边界,本文给出由与边相邻三角网格的类型来确定空间网格面片可见边界的算法。并实现了三角网格面片可见边界沿任意矢量在某平面上所得投影边界的求并运算。最后给出采用该算

基金项目:国防基础科研项目(k1203010804)

收稿日期:2006-01-13;改回日期:2006-07-17

第一作者简介:郭爱斌(1977~),男,北京航空航天大学机械制造及其自动化专业博士研究生。主要研究方向为CAD/CAE、复杂系统多学科优化设计。E-mail: abguo@buaa.edu.cn

法完成卫星构型分析中太阳翼遮挡分析应用实例。

## 2 可见边界提取算法

### 2.1 基本概念及定义

在描述算法前,需在三角网格数据中定义邻接三角形网格和邻接点。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。三角网格面片上一点  $p$  的所有邻接三角网格构成  $p$  的邻接三角网格集合  $T_p = \{T | p \in T, T$  是面片上的一个三角网格 $\}^{[2]}$ 。对于网格数据中的边,如果它是某一个三角网格的一条边,则称这个三角网格是该条边邻接三角网格;三角网格面片上一条边  $e$  所邻接的三角网格集合为  $T_e = \{T | e \in T, T$  是网格面片上的一个三角网格 $\}$ ,  $T_e$  元素的个数是 1 或 2,当边属于三角网格面片的边界时,与其相邻的三角网格数为 1,否则为 2,该边称为内边。

三角形对应法向量与光线方向  $S(nx, ny, nz)$  的夹角的余弦值大于 0 时,称其为可见三角形;否则为不可见三角形。根据网格面片上可见三角网格的情况,网格面片可分为完全可见、完全不可见和部分可见 3 种。

如果三角网格可见,则它的 3 条边均为可见边。如果与边相邻的三角网格只有一个可见,则该边称为可见边界边。

如果可见三角形中 3 条边均为可见边界边,则该三角形为 3-类三角形;有两条边为可见边界边,则该三角形为 2-类三角形;只有一条边为可见边界边,则该三角形为 1-类三角形;不属于 1-类三角形、2-类三角形、3-类三角形的三角形为 0-类三角形(包括不可见三角形和 3 条均为可见内边的可见三角形)。

在图 1 中,假设该三角网格面片为完全可见面片。构成面片的三角网格的边均为可见边,边  $AB$ 、 $BC$ 、 $CK$ 、 $KL$ 、 $LC$ 、 $CD$ 、 $DE$ 、 $EF$ 、 $FG$ 、 $GA$  等为边界边,其余边均为内边。 $\triangle CKL$  为 3-类三角形, $\triangle AFG$  为 2-类三角形, $\triangle ABH$ 、 $\triangle BCI$ 、 $\triangle CDI$ 、 $\triangle DEJ$ 、 $\triangle EFJ$  等为 1-类三角形,其余三角形为 0-类三角形。

### 2.2 三角网格数据存储

存储三角网格数据的数据结构包括点、边、三角网格、法向量等,具体定义如下:

#### (1) 点类型

```
Public Type VertexType
```

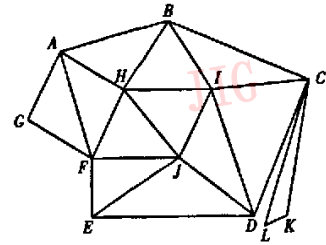


图 1 三角网格面片示例

Fig. 1 Example of mesh surface

```
x As Double; y As Double; z As Double
```

```
EdgePointer() As Long; TriPointer() As Long
```

```
End Type
```

*EdgePointer* 存储与点相连的边在边列表中的位置;*TriPointer* 存储与点相邻的三角形在三角网格列表中的位置。

#### (2) 边类型

```
Public Type EdgeType
```

```
Point1 As Long; Point2 As Long
```

```
MeshPointer(1) As Long
```

```
End Type
```

*Point1*, *Point2* 表示边顶点在顶点列表中的位置;*MeshPointer* 存储与该边相邻的三角网格在三角网格列表中的位置,如果为边界边时, *MeshPointer*(1) 为 -1。

#### (3) 法向量类型

```
Public Type NormalType
```

```
nx As Double; ny As Double; nz As Double
```

```
Visual As Byte; TriPointer() As Long
```

```
End Type
```

*Visual* 标识与法向量相关的三角形网格的可见性,可见为 1,不可见为 0;*TriPointer* 存储与其相关的三角形在三角网格列表中的位置。

#### (4) 三角网格类型

```
Public Type TriType
```

```
p1 As Long; p2 As Long; p3 As Long
```

```
e1 As Long; e2 As Long; e3 As Long
```

```
Normal As Long; Visual As Byte
```

```
End Type
```

$p_1, p_2, p_3$  为三角网格的 3 个顶点在面片顶点列表中的位置; $e_1, e_2, e_3$  为三角网格的边在边列表中的位置;*Normal* 表示其对应法向量在法向量列表中的位置;*Visual* 标识三角网格可见性:不可见为 0,可见为 1。

另需定义数组 *VertexList*、*EdgeList*、*TriList* 和 *NormalList*, 它们分别为 *VertexType*、*EdgeType*、*TriType* 和 *NormalType* 类型,用来存储三角网格面片

中点、边、三角网格和法向量。为了数据存取方便,定义 Byte 类型数组 *Tag* 存储网格面片中三角网格对应的三角形类型,数组 *Tag* 与 *TriList* 长度相同。

### 2.3 算法实现

(1) 从 3 维 CAD 模型中提取三角网格面片数据并存储在数组 *VertexList*、*EdgeList*、*TriList* 和 *NormalList* 中。

(2) 由给定光线方向 *S* 判断并标记 *TriList* 中所有三角形网格的可见性。

(3) 判断三角网格类型并将其类型存储在对应的数组 *Tag* 中 (*Tag*(*i*) 的值为 0、1、2、3, 分别表示第 *i* 个三角形网格的类型)。

(4) 求解三角网格面片的可见边界,若三角网格面片中三角形均为 0-类三角形,则转入第 8 步;否则,从非 0-类三角形中任取一个三角形的一条可见边作为面片一条可见边界的起始边,并保证可见三角形在该起始边的左侧(边界有向性),之后并将该三角形标记为 0-类三角形。

(5) 如果该三角形为 1 类三角形,查找当前可见边界终点相邻的 1-类三角形、2-类三角形,并将该三角形标记为 0-类三角形,之后转入第 7 步。

(6) 如果该可见三角形为 2-类或 3-类三角形,接着将该三角形的另一顶点添加到正在求解的可见边界中,并作为可见边界新的终点,如果该可见三角形为 3-类三角形,这时一条可见边界已找完,返回第 4 步查找其他可见边界;如果该可见三角形为 2-类三角形,执行和 1-类三角形相同的操作,并转入第 7 步。

(7) 如果即将处理的可见三角形的可见边上另一点与当前可见边的起点重合,则找到一条可见边界,返回第 4 步查找其他可见边界;否则,将该点添加到当前求解的可见边界,并作为可见边界新的终点,返回第 5 步。

(8) 将所有找到的可见边界向指定平面投影,如果可见边界只有一条,则投影即为所求,如果不止一条,将得到的投影(平面多边形)采用文献[5]中的方法求并,得到投影边界。

(9) 结束。

算法实现流程图如图 2 所示。

### 2.4 算法分析

本文分析结果的误差来源于 CAD 软件划分三角网格的质量,在此基础上分析,该算法不会增加分析结果的误差。

设所处理的面由 *n* 个三角形组成,对应  $k(1 \leq$

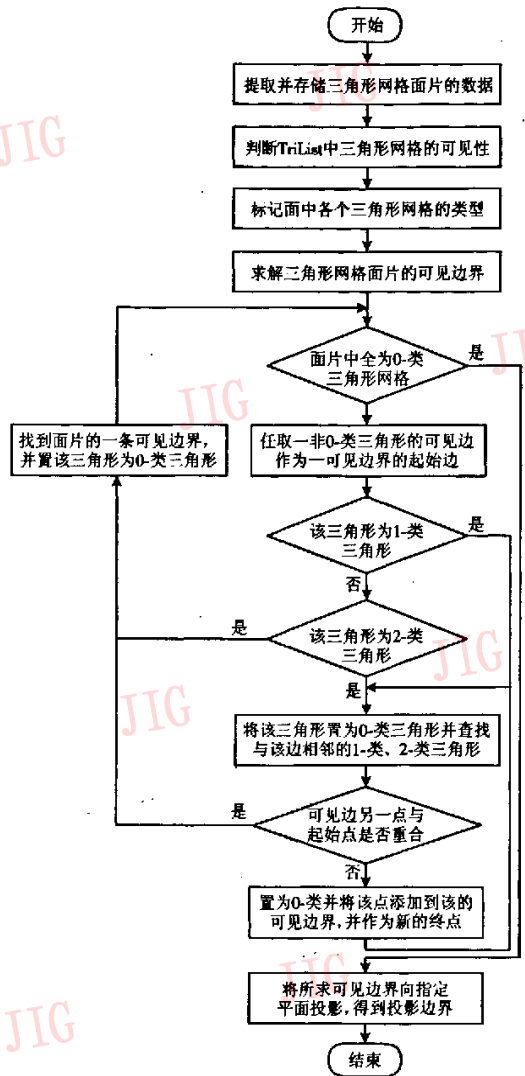


图 2 求解可见边界算法实现流程图

Fig.2 Detecting visible border flow chart

$k \leq n$ ) 个法向量,当面片为单一平面时 *k* 取 1,当面片为连续曲面时 *k* 取 *n*。在第 2 步中要进行 *k* 次法向量比较,每次法向量比较需要做 3 次乘法运算、2 次加法运算和 1 次比较运算。在第 3 步中要做 3*n* 次比较和 *n* 次赋值运算。在第 4 步中每次判断顶点在起始边的左侧时需要做 19 次加法、14 次乘法、两次比较和 1 次赋值运算。提取面片所有可见边界在第 5 步、第 6 步、第 7 步中完成,对于不含有图 1 中所示的孤立三角形(文献[4]中称为错误三角剖分)的情形,查找全部可见边界时,需要查找与当前顶点相连的三角网格最多次数为 2*n* 次,每次做的 6 次

比较和 2 次赋值运算;对于含有孤立可见三角形的情形,在第 6 步中进行了特殊处理,即 3-类三角形对应的边为该面片的一条可见边界。综上,该算法的复杂度为  $O(n)$ 。

实际分析中,如果网格面片为平面,在第 2 步中只需做一次法向量比较。该面片可见时,其可见边界与面片的实际边界一致,不可见时,无可见边界。对于一些比较规则的连续曲面的可见边界提取,可不采用该方法,如球面,无论光线怎样变化,它的可见边界始终是一个圆,该圆所在平面与光线方向垂直。这样可以进一步提高计算速度。

### 3 实例验证

实例的 3 维实体图如图 3 所示,  $S$  为光线方向(图中箭头指向),矢量为  $(1, 0, 0)$ ,沿光线的可见面如图 3 中所示的灰色曲面。图 4 为该曲面三角网格表示,该面片由 58 个三角网格组成,包括 60 个顶点和 117 条边。图 4 中粗实线为沿光线方向  $S$  的可见边界。该面片沿  $S$  方向在平面  $x = -50$  上的投影边界为点  $(-50, 500, 0)$ 、 $(-50, 500, 901.11)$ 、 $(-50, -500, 901.11)$ 和  $(-50, -500, 0)$ 连成的四边形。该面片可见边包括 38 条可见边界边,在已知该面片的点、边和三角网格相邻关系的前提下,求解可见边界用时 250ms。计算环境为 Pentium4 处理器,CPU 2.00GHz,内存 512MB。

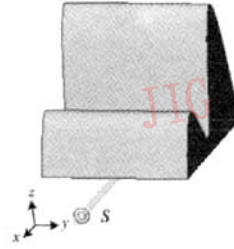


图 3 3 维实体图

Fig.3 3D solid part

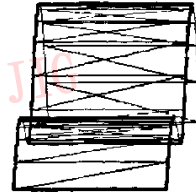


图 4 网格面片示意图

Fig.4 Mesh surface sketch

### 4 应用

运用本文提出的空间网格面片可见边界提取算法可实现不同类型卫星的太阳翼遮挡分析计算。太阳翼遮挡分析是指在给定太阳光方向和卫星构型的前提下,分析卫星本体对两翼电池阵的遮挡影响。该分析模块为用 VB6.0 开发的一个 SolidWorks 插件。图 5 为结果查看界面,该界面显示受晒面积变

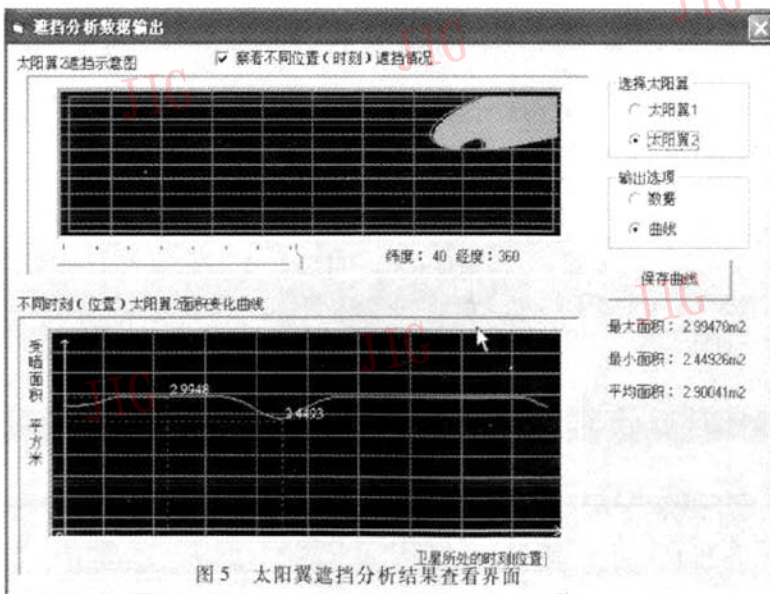


Fig.5 Browsing solar array shield analysis result interface

化曲线和数据以及任一时刻遮挡示意图、受晒面积和遮挡深度等。结果表明,计算速度、精度均能满足工程实践需要。

## 5 结 论

本文把三角网格面片边界与可见方向关联起来,给出可见边界概念,并根据三角网格具有可见边界边的数目将其定义为 0、1、2、3 共 4 类三角形,进而由三角网格类型确定面片的可见边界。该算法时间复杂度为  $O(n)$ 。最后以实例说明了该算法的可行性和有效性,并将该算法应用于卫星构型分析中的太阳阵遮挡分析模块,其计算速度、精度能满足工程实践要求。

### 参考文献 (References)

1 Sun Y, Page D L, Palk J K, *et al.* Triangle mesh-based edge detection and its application to surface segmentation and adaptive

surface smoothing [A]. In: Proceedings 2002 International Conference on Image Processing [C], Rochester, New York, USA, 2002; 825 ~ 828.

- 2 Tan Zheng, Wang Li-sheng. Detection of edge of surface mesh data [J]. Journal of Computer Aided Design and Computer Graphics, 2000, 12(8): 580 ~ 584. [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 Hagen H, Muller H, Nielson G M. Focus on Scientific Visualization [M]. New York: Springer-Verlag, 1993: 73 ~ 91.
- 4 Zhang Xian-ying, Zhou Ming-quan, Geng Guo-hua. A method of detecting the edge of triangular mesh surface [J]. Journal of Image Graphics, 2003, 8(10): 1223 ~ 1226. [张献颖,周明全,更国华. 空间三角网格曲面的边界提取方法[J]. 中国图形图象学报, 2003, 8(10): 1223 ~ 1226.]
- 5 Yu Lei-yi, Bian Fu-ling, Wan Feng. An efficient algorithm for intersection, union and difference between polygons [J]. Geomatics and Information Science of Wuhan University, 2003, 8(10): 615 ~ 618. [于雷易,边福苓,万丰. 一种多边形交、并、差运算的有效算法[J]. 武汉大学学报(信息科学版), 2003, 8(10): 615 ~ 618.]