DXF文件中3D网格模型的简化

姓名： 刘清华

学号： PB04203220

导师： 董兰芳

二〇〇六年十月

整体介绍

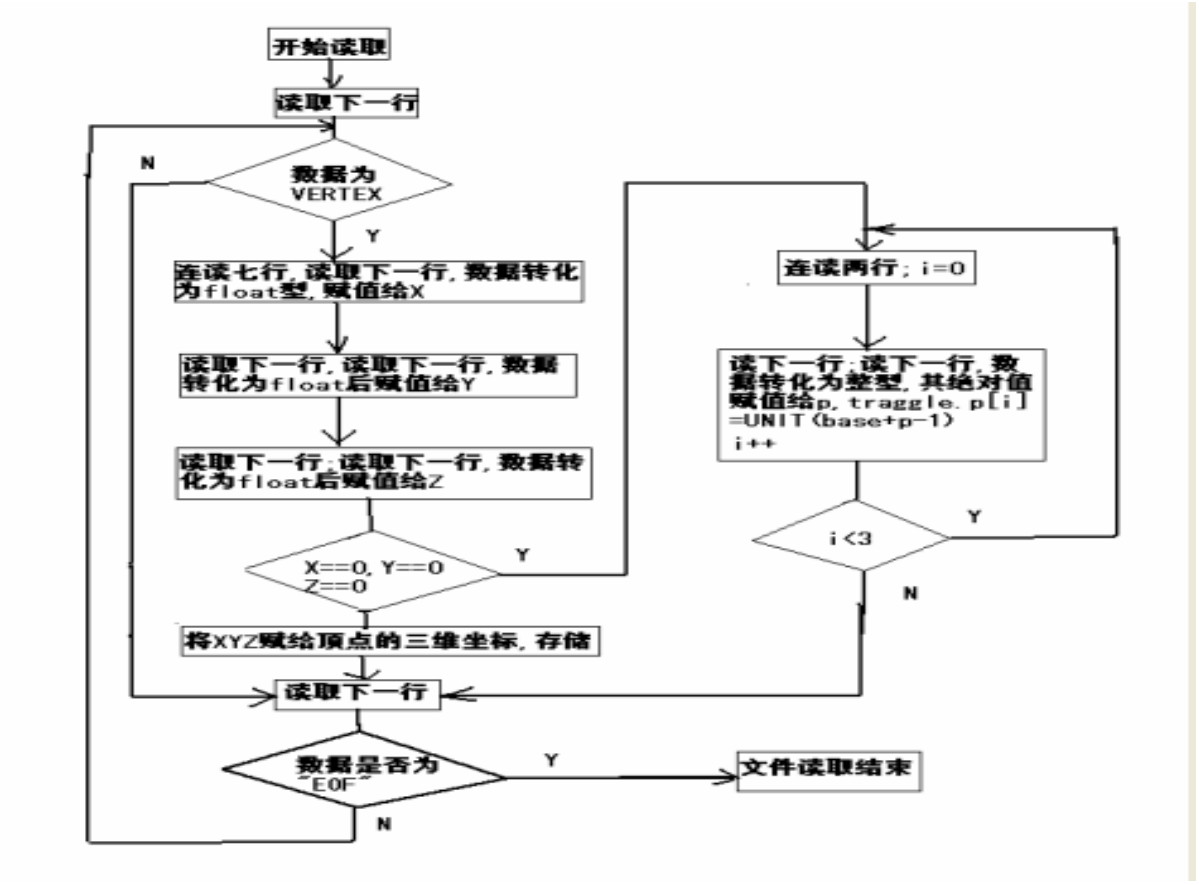
- 主要内容：
  - DXF文件的数据存放格式。
  - 常用的面片简化方法QEM的研究。
  - 基于改进的二次误差的半边折叠,三角形删除以及中值滤波的快速简化方法 。
  - 一个对DXF文件中3D网格模型简化的系统的设计和实现。

DXF数据存放格式介绍

- 一个由AutoCAD生成的DXF文件由七段组成：HESDER段，CLASS段，TABLES段，BLOCKS段，ENTITIES段，BJECT段和THUMBNAILIMAGE段
- 我们最关心的是ENTITIES段下OLYLINE中的VERTEX图元部分和3DFACE图元部分，存放有顶点信息和面片信息

包含VERTEX图元数据的读取

- 数据读取流程图



网格模型DXF文件输出

- 对于开头的几个字段可以按原文件录入，同时可以根据需要部分省略。
- 对于顶点和面的信息按内存中以适当格式输出
  - 可以只将所有的点和面保存为一个图层，也可以保存为多个。
  - 对于原有的尺寸，摄像机位置等不做改动。

网格模型简化的概念

- 三角形网格与多边形网格
- 网格简化的意义和实际应用
- 网格简化的概念
  - 全局简化与局部简化
  - 静态简化与动态简化
  - 网格简化与LOD模型
  - 简化的原则与简化误差
    - 两模型的几何相似测度（最大与平均）

常用的几种网格简化算法

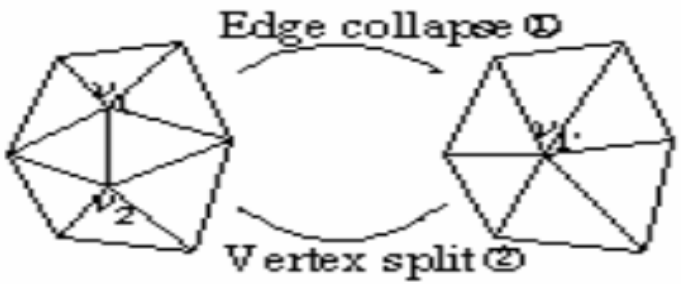
- 顶点聚类方法
- 区域合并法
- 重新布点法
- 逐步求精法
- 边折叠法
- 三角形删除法
- 小波分析法等

由于网格特征的复杂性与多样性，目前仍然没有一个通用的方法

常用的几种网格简化算法

- QEM算法
  - 边折叠方法

边折叠	边折叠的关键是折叠的次序以及边折叠后新顶点的位置
点分裂	HOPE采用能量优化的方式来确定折叠次序和新顶点的位置



①边折叠,②点分裂.

Garland和Heckbert提出了一种基于二次误差测度(quadric error metric,简称QEM)的化简算法

常用的几种网格简化算法

- QEM算法
  - 二次误差测度

在三维欧氏空间，一个平面可以表示为 $ax+by+cz+d=0$，而平面法矢为 $n=(a,b,c)^T$,且有 $a^2+b^2+c^2=1$.点 $v=(x,y,z)^T$ 到该平面距离为

$$D^2(v)=(n^Tv+d)^2=(ax+by+cz)^2.$$

定义该顶点的二次误差为该点到各平面距离平方之和：

$$E_{plane}(v)=\sum D_i^2(v)=\sum_i (n_i^Tv+d_i)^2.$$

$$D^2(v)=v^T(nn^T)v+2(dn)^Tv+d^2.$$

常用的几种网格简化算法

- QEM算法
  - 二次误差测度

    定义三元组其中 $Q=(A,b,c);A=nn^T;b=dn;c=d^2.$

    Q称作二次误差测度或者二次矩阵，二次误差测度值为 $Q(v)=D^2(v)$ 也叫做二次误差，而且误差可以直接累加：

$$Q_i(v)+Q_j(v)=(Q_i+Q_j)(v).$$

$$Q_i+Q_j=(A_i+A_j,b_i+b_j,c_i+c_j).$$

常用的几种网格简化算法

- QEM算法
  - 二次误差测度实质

它相当于求解以下关于x，y，z的方程组：

$$\begin{cases} a_1x+b_1y+c_1z+d_1=0 \\ \cdots\cdots\cdots \\ a_nx+b_ny+c_nz+d_n=0. \end{cases}$$

以上方程组试图求解位于n个平面上的点（该点到所有平面的距离均为0）。我们知道，当n>3时，以上方程一般无解。但可以采用最小二乘法来计算其最优近似解。所以二次误差测度就相当于用最小二乘法计算该最优近似解。

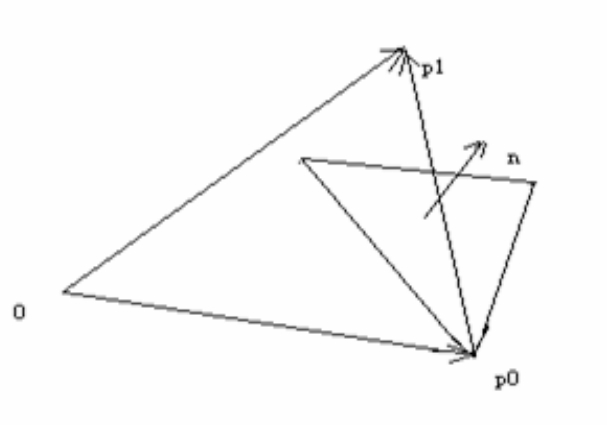
常用的几种网格简化算法

- QEM的一些不足
  - 进行平面和矩阵的运算
  - 尖端特征消失
  - 局部过度简化
  - 边界收缩
  - 狭长三角形产生
  - 三角形面积差异大

需要做局部的改进

我的二次误差计算方法

- 向量计算
  - 从定义出发计算点到面的距离

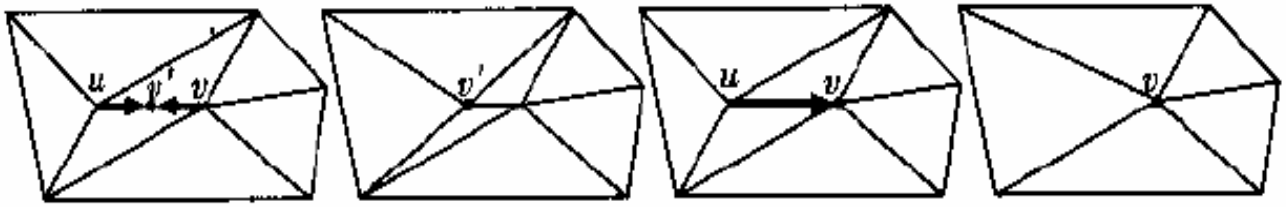


$$d = (\overrightarrow{p_1} - \overrightarrow{p_0}) \bullet \vec{n} = \overrightarrow{p_1} \bullet \vec{n} - \overrightarrow{p_0} \bullet \vec{n}$$

$$Q = \sum d \times d$$

半边折叠方法

- 为了便于生成LOD模型
  方便计算与存储
- 操作
  - 边折叠的新顶点为原来中的一个
  - 将误差记录到边上，$\xi = \min\{\xi_u, \xi_v\}$

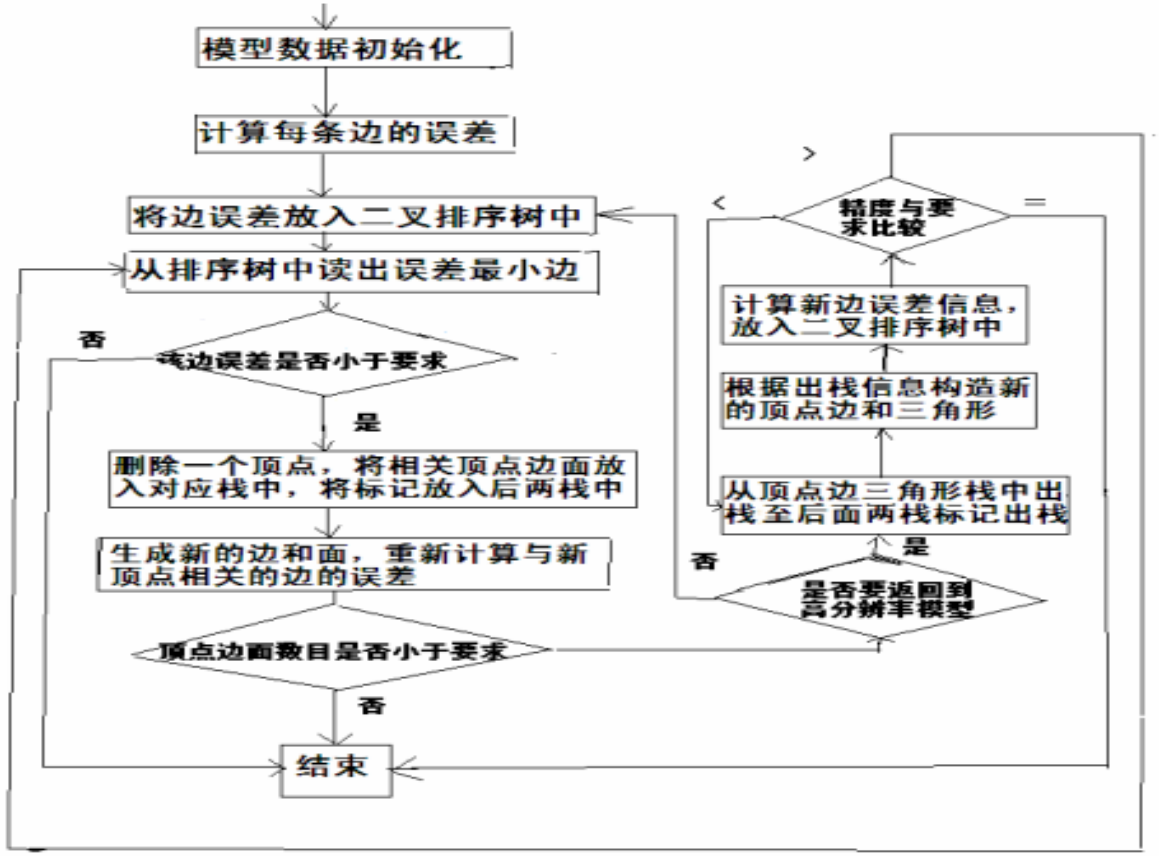


二次误差半边折叠实现

- 局部保护方法
  - 面积加权　　$Q = \Sigma d \bullet d \bullet s$
  - 新删除顶点的保护　　**标记**
  - 尖端保护　　$Q=Q \bullet (1+c \bullet m)$
  - 边界边处理　　**判定**
  - 狭长三角形的检测　　**计算判定**
- 使得结果保持原来拓扑结构

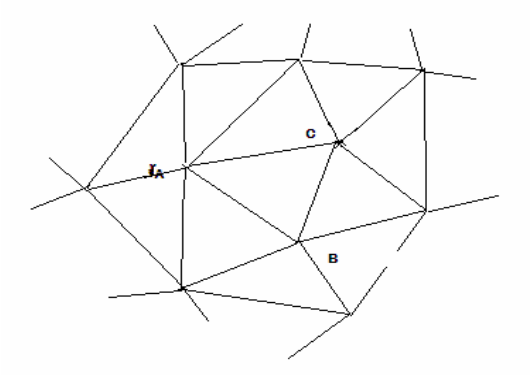
二次误差半边折叠实现

- 流程

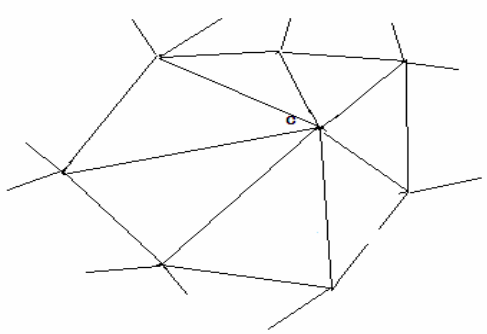


二次误差三角形删除实现

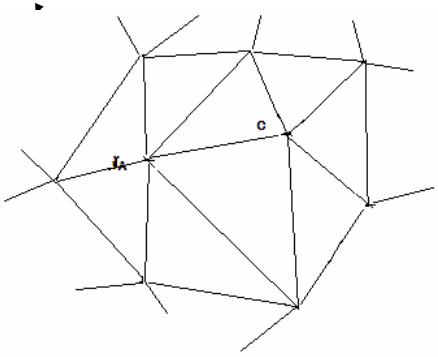
- 三角形删除操作
- 二次误差:将误差记录到三角形



原始网格



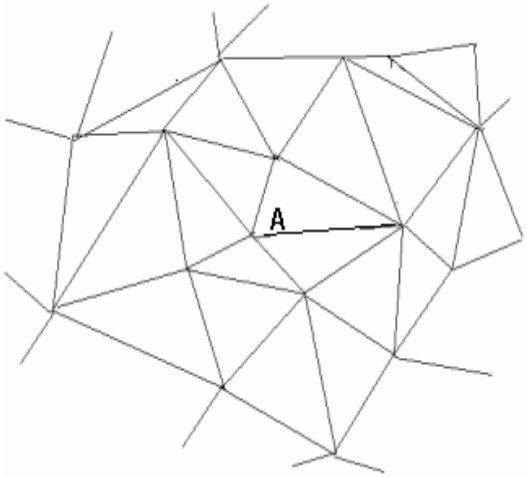
保留顶点



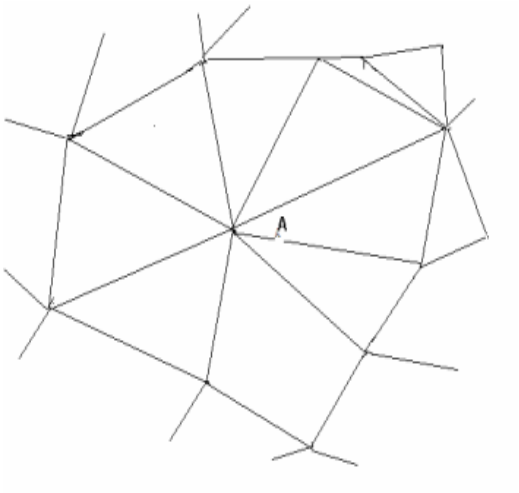
保留边

中值滤波处理方法

- 操作方法



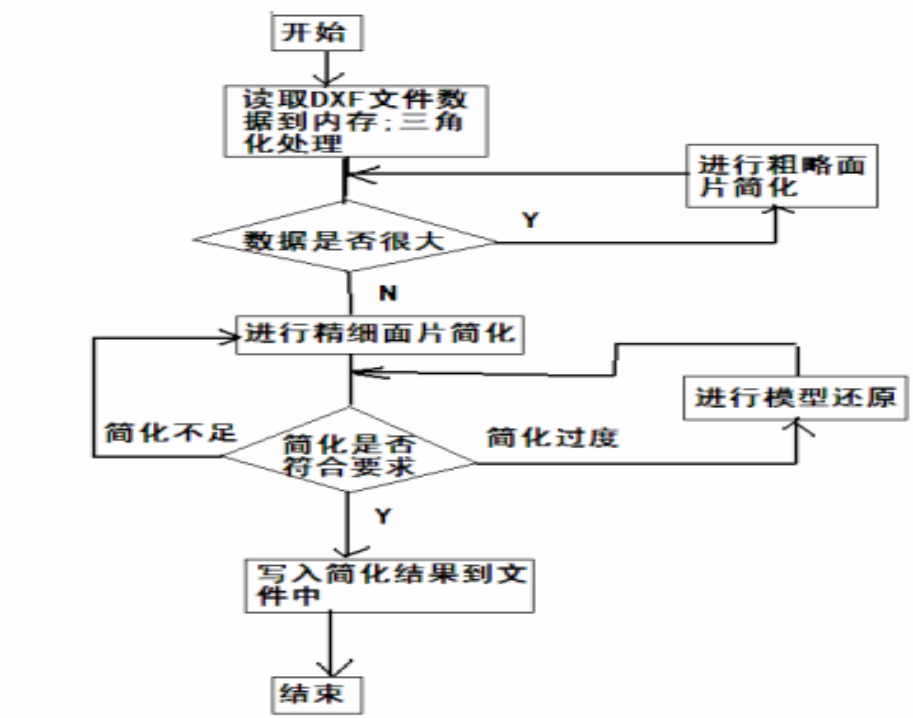
原始网格



简化后

DXF文件中3D网格模型简化系统的实现

- 流程

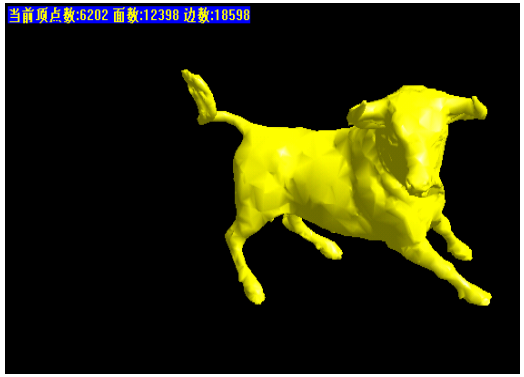


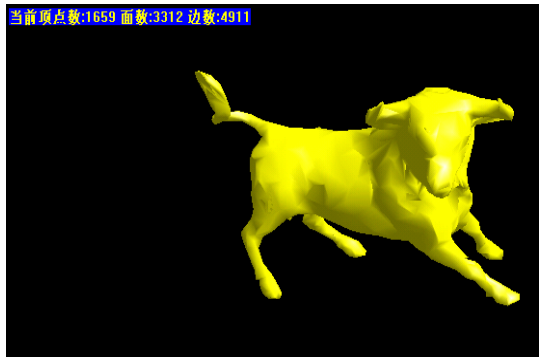
开发工具　VC++ 6.0　＋　DerictX API

DXF文件中3D网格模型简化算法实现

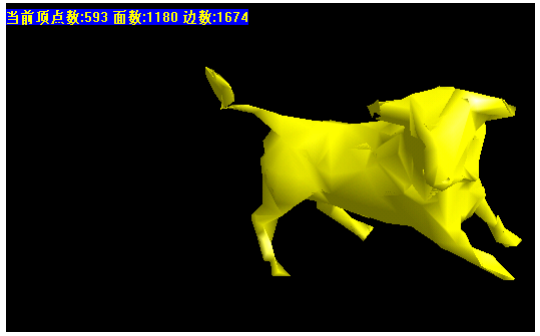
- 中值滤波与半边折叠方法

简化前

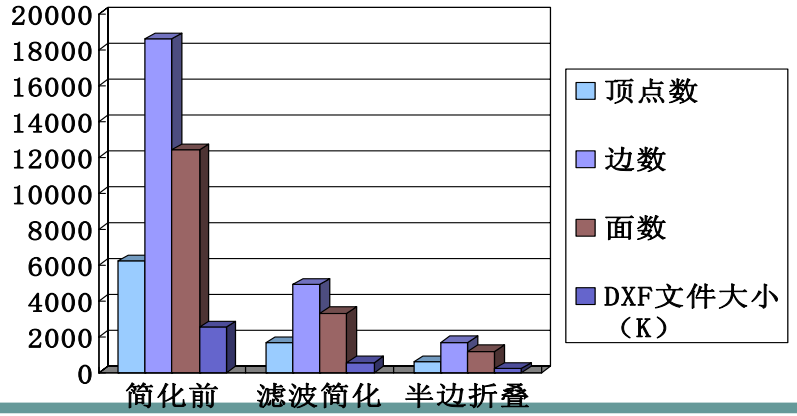




中值滤波简化后

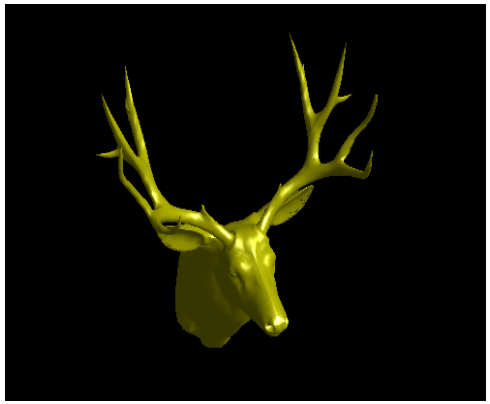


半边折叠最后完成

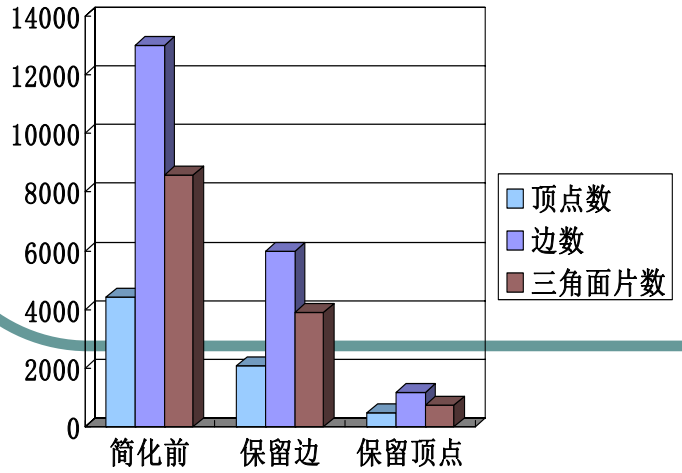


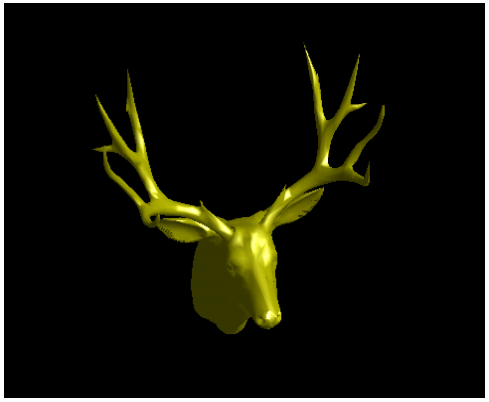
DXF文件中3D网格模型简化算法实现

- 三角形删除简化方法

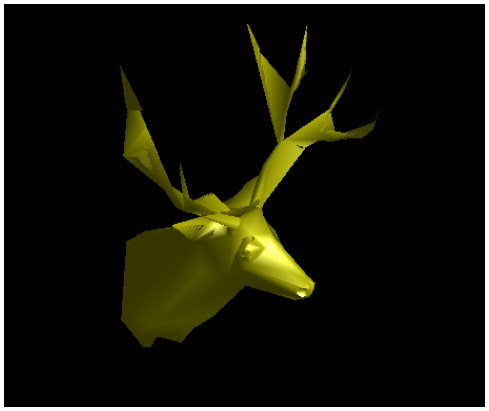


简化前





保留为边的三角形删除



保留为顶点的三角形删除

DXF文件中3D网格模型简化的应用

- 虚拟现实
- 3D游戏
- 手机3D显示.....

模型简化后在**J2me**平台的手机模型器上显示

DXF文件中3D网格模型简化实现

- 算法实现总结和思考
  - 从运行的结果来看，对于大多数DXF文件中的3D网格模型，都可以得到较好的简化结果。
  - 在高度简化时，通常各种简化算法都会生成一些不可预料的边或者删除一些关键的顶点。失去了原来的拓扑结构。
  - 难以处理好整体外观和局部特征的一对矛盾

工作总结

- 大研期间的主要工作
  - 学习并总结比较了常用的面片简化算法
  - 提出了方便的二次误差计算方法，结合实际问题对原有边折叠方法的一种基于局部特征保持的改进方法，还尝试了一些其它的快速简化的方法。
  - 设计并实现对DXF中3D网格模型自动简化的程序,对多个模型进行测试,取得了较好的效果.

谢 谢！