

基于手画多边形的三维网格模型生成算法

丁剑飞(2007210830), 杨甲东(2008310402), 王丽华(2008310403)
dingjf07@mails.tsinghua.edu.cn

1 前言

借助于用户界面技术,快速、方便、友好地创建三维实体模型是计算机图形学和计算机辅助设计领域的经典问题。文献[1]中设计了一种名为“Teddy”的基于手画图形的三维模型创建与编辑系统。该原型系统能够根据用户创建的二维手画图形,自动地生成三维空间的多面体模型;同时该系统还能够根据用户的要求对生成的三维模型进行修改。相比于其他方法,该原型系统使得用户不必再通过繁琐的操纵控制点的方法就可以快捷的得到所需要的三维模型。

本次实验实现并改进了文献[1]中设计的基于手画多边形的三维网格模型生成算法。该算法根据用户在二维平面输入的封闭的简单多边形,自动的生成三维空间内的网格模型,从而使得用户简单自由地创建所需要的三维实体。相比于文献[1]所提出的算法,本实验对于若干特殊的情况给予了更为充分和细致的分析和处理,从而提高原有方法的适用范围。

本报告的主要内容安排如下:第2部分将介绍对本实验的相关背景;第3部分将详述本次实验的各主要步骤;第4部分将介绍就实验各个步骤中所涉及到的重要的数据结构;第5部分将介绍本次实验中所遇到的若干重要问题及其解决办法;第6和7部分给出了模型实例的生成过程及效果图;第8部分将对总结实验并对今后的改进工作做出展望。

2 相关背景

2.1 Teddy 系统生成三维网格模型的主要步骤

在文献[1]设计的名为“Teddy”的系统中,生成三维网格模型的主要包括以下几个步骤:

- 1) 实体构建 (Object Creating): 以用户在二维平面的手画多边形为输入,通过对该多边形进行约束的 delaunay 三角剖分,提取出其中的骨架,并将其按一定比例在至三维空间内进行提升,从而形成某个三维实体模型,如图 2-1 所示。

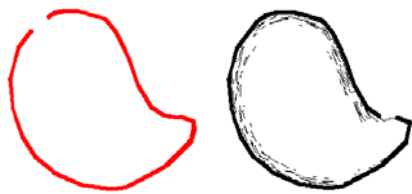


图 2-1 实体构建步骤示意图

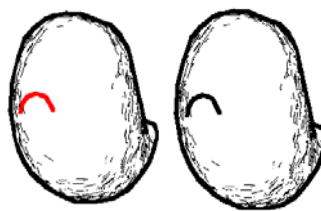


图 2-2 绘制步骤示意图

2) 绘制(Painting): 在获得三维实体模型之后, 用户可以在模型表面进一步以自由的手绘的形式, 编辑实体模型的表面属性。系统将根据用户输入投影到三维空间内, 从而生成更为真实和生动的三维实体表面如图 2-2 所示。

3) 处理凸起变形(Extrusion): 得到三维实体之后, 用户可以在三维实体空间内, 以自由的手绘的形式, 对三位实体模型进行再次编辑, 从而补充在第一次实体模型生成中不易得到的结构。如图 2-3 所示。

4) 处理截断变形(Cutting): 对于三位实体模型进行再编辑, 也可以利用手画的形式添加语义信息, 从而将原有实体中的某个部分截去, 如图 2-4 所示。



图 2-3 凸起变形示意图

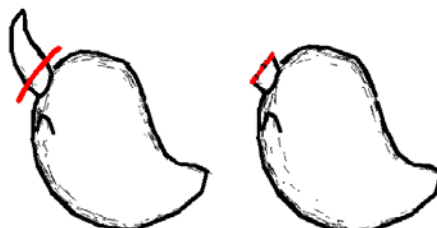


图 2-4 截断变形示意图

5) 光滑表面(Smoothing): 经过多次编辑之后, 三维实体表面的可能会有很多冗余的结构。为了得到最终的理想的绘制效果, 用户可以在在实体上输入的笔画, 对实体表面存在的不平整区域光滑处理。如图 2-5 所示。

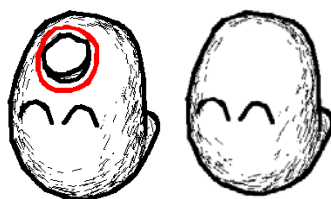


图 2-5 光滑表面示意图

2.2 本文的主要工作

上述的五个基本步骤构成了 Teddy 原型系统最为核心的功能。在这些步骤中, 第一步, 即根据用户在二维平面上的手画多边形生成三维网格实体模型, 无疑是整个系统中最基础、最核心的步骤。鉴于此, 本次实验将完整地实现该步骤。同时, 对于原文中没有考虑到的若干特殊情况也给予了扩展。

3 算法基本流程

根据用户在二维平面上的手画多边形生成三维网格实体模型主要包括以下几个步骤：

3.1 约束的 delaunay 三角剖分

delaunay 三角剖分将原有的多边形所围成的平面区域剖分为若干个三角形，使得这些每一个三角形内的三个角的角度尽可能接近。但是，一般意义的 delaunay 三角剖分结果不能直接作为后续步骤地运算基础。因此，需要对 delaunay 三角剖分的结果施加一些约束条件。其中的细节，详见本报告第 5 部分。

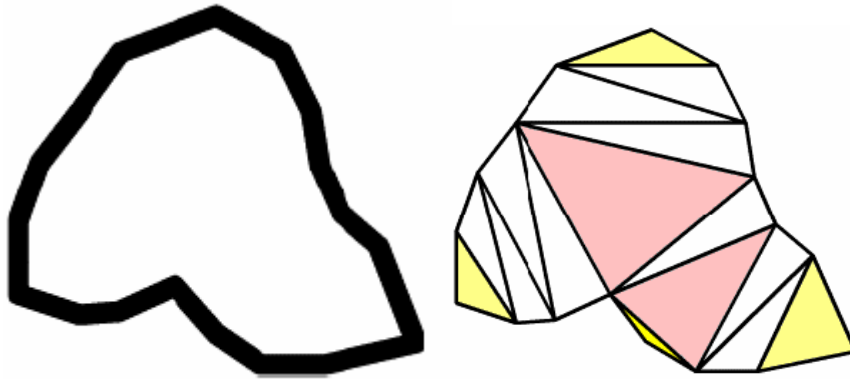


图 3-1 约束的 delaunay 三角剖分

3.2 骨架生成

在三角剖分的基础上，可以对其中的三角形分成三类[2]：terminal 三角形、sleeve 三角形以及 junction 三角形。其中，terminal 三角形，指该三角形有两条边在原多边形的边界上，如图 2-1 中黄色三角形所示；sleeve 三角形，指该三角形有一条边在原多边形的边界上，如图 2-1 白色三角形所示；junction 三角形，指该三角形没有一条边在原多边形的边界上，如图 2-1 黄色三角形所示。

根据三角形不同种类，连接生成相应的边，从而获得初步的骨架。具体的方法为：对于 junction 三角形，将各边中点连接至该三角形的外心；对于 sleeve 三角形，直接连接两个非边界边的中点；对于 terminal 三角形，连接该三角形中非边界边的中点和其所对的顶点。如图 3-2 所示；

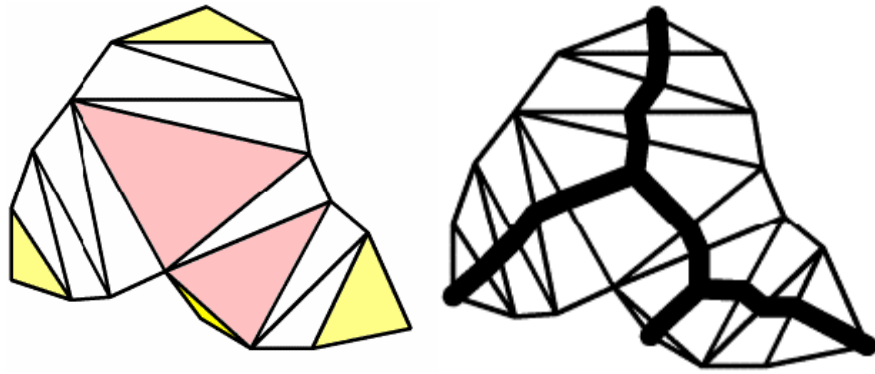


图 3-2 骨架初步生成

3.3 连线退化

为了使各个顶点在从二维多边形生成三维实体模型的过程中明确各自的空间支撑结构关系，需要对多边形内部的各条连线进行合并退化。合并退化的主要方法是，首先尝试将每一个 **terminal** 三角形与邻近的三角形合并形，如图 2-2 所示。在合并后的区域中，以非边界的那条边为直径作半圆，如果半圆将该区域完整包括，则继续合并邻近区域。如图 3-3 所示。

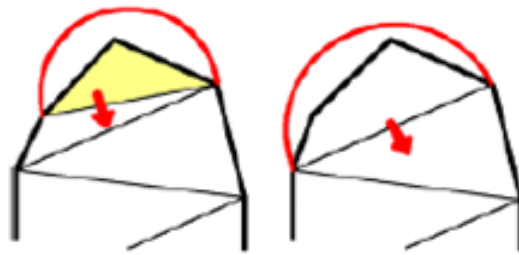


图 3-3 连线退化的基本方法

连线退化过程的终止条件有两类。A.当退化到 **sleeve** 三角形时，生成的区域不能被完全包括在半圆之中，如图 3-4(a)所示；B.一直退化到 **junction** 三角形，如图 3-4(b)所示。对于第一种情况，将该区域内所有的多边形边界的顶点同内部边的中点连接起来，如图 3-4(c)所示；对于第二种情况，将该区域内所有的多边形边界的顶点同该 **junction** 三角形的外心用线段连接起来，如图 3-4(d)所示。

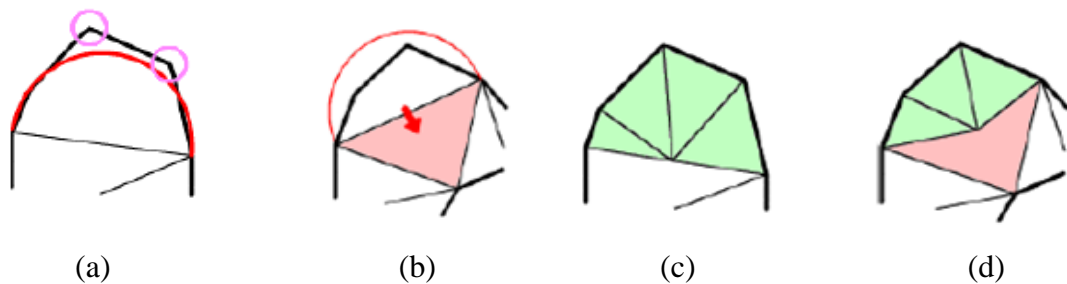


图 3-4 连线退化过程结果

3.4 骨架再生成和细分

在连线退化结果的基础上，连接所有 sleeve 三角形和 junction 三角形边的中点，即形成了该多边形的脊轴线，如图 3-5(a) 所示，其中各个边的中点称为骨架点。在生成骨架线的基础上，还需要对于其中的 sleeve 三角形和 junction 三角形再进行划分。具体的说，就是在 sleeve 三角形和 junction 三角形中，将骨架点同三角形的未连接的顶点连接起来，如图 3-5(b)所示。

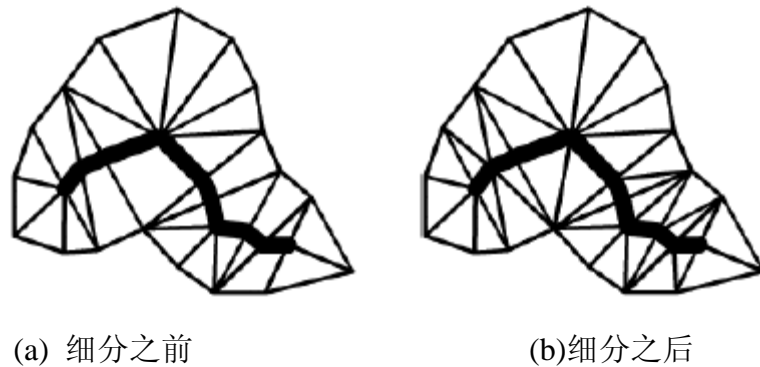


图 3-5 骨架生成示意图

3.5 提升骨架并生成空间网格

在前几步的基础上，本步骤将骨架线从原多边形所在的二维平面提升出来。骨架线各个关节点，即骨架点所提升的高度与其到相邻边界点的平均距离成正比，如图 3-6(a)(b)所示。在提升骨架点的同时，与该点相连接的直线将转换为相对应的空间椭圆曲线，如图 3-6(c)所示。在此基础上，还需要对椭圆曲线所围成的空间曲面进行细分，即对该区面进行三角剖分，从而形成最终的三维网格实体模型，如图 3-6(d)所示。

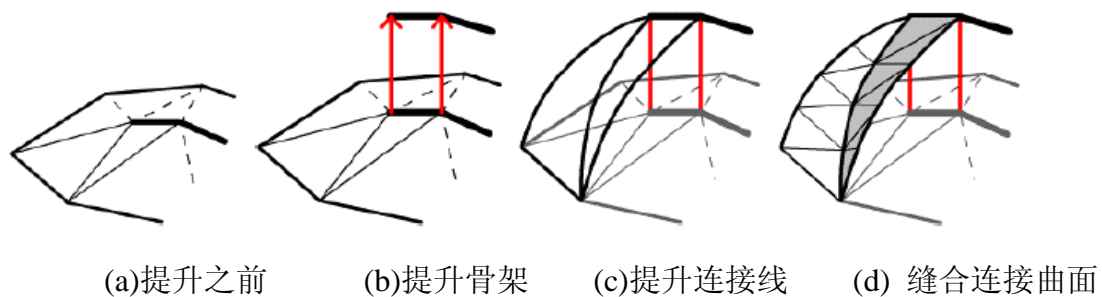


图 3-6 骨架提升

4 数据结构

记录三角面片时采用了普通数据结构和半边数据结构相结合的方式：

普通数据结构：只记录面的三个顶点

半边数据结构包括一下三个结构：

1)面结构：逆时针顺序记录了面的三条边，并存储了面的类型。

- 2) 顶点结构：存储了顶点的坐标和经过该顶点的线段集。
- 3) 边结构：存储了边的起始点、终止点、对称边及逆时针顺序的下一条边。按逆时针顺序，记录面的三条边。并记录每一条边的对称边。

另外为了保证平面和空间中每次加入新顶点的唯一性，用到了标准 C++ 的 map 结构和 pair 结构。

5 解决的难点问题及创新

1) 针对简单多边形 delaunay 三角剖分的约束方法：将三角边的中点在多边形外或者边与多边形相交的三角面去除掉。得到了简单多边形的三角剖分结果。

2) 针对连接三角形 (junction 三角形) 要根据三角形的形状采用不同方法取骨架：对于锐角三角形：取三角形的外心，然后依次与每一条边的中点相连构成骨架。对于钝角和直角三角形：取最长边的中点然后依次与其他两条边的中点相连接构成骨架，如图 5-1 所示。

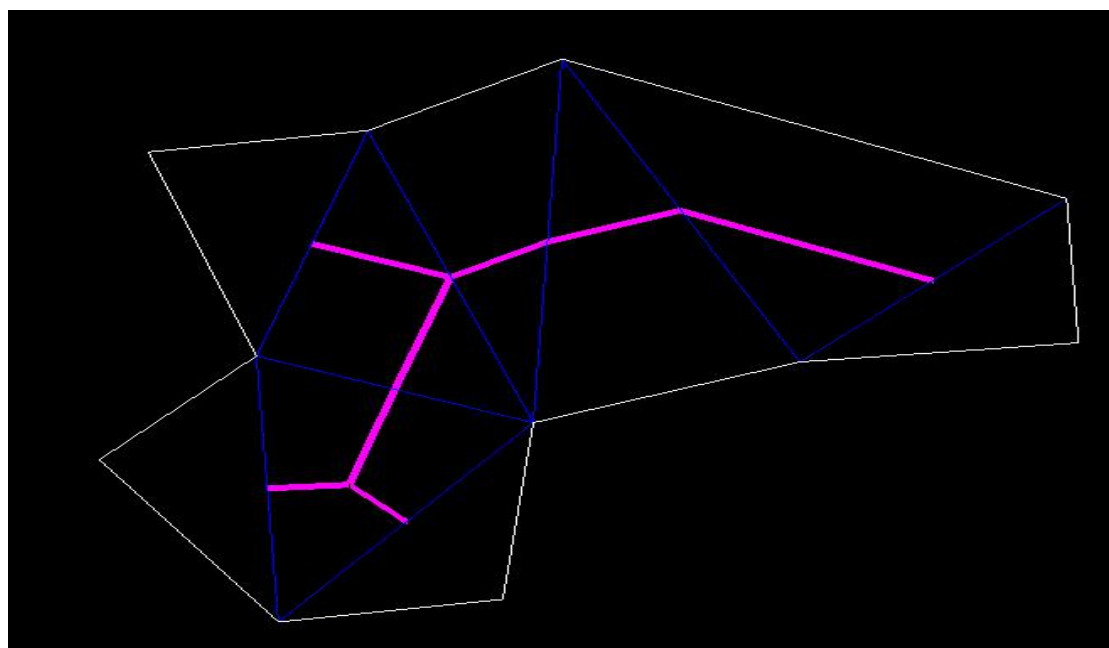


图 5-1 两个 junction 三角形不同的取骨架方式

3) 加入顶点唯一性保证：在进行退化和细分处理时，会产生新的顶点插入。同样，在提升生成空间网格时也会有新的顶点生成，为了避免同一点重复添加，我们利用了标准 C++ 的 map 结构中的关键值的特性，配合 pair 结构实现了加入顶点的唯一性保证。

在每次加入顶点前我们都先把顶点试图存到全局的 map 结构中，如果存入 map 成功同时进行编号则说明，还未加入该顶点。否则说明已有该顶点，通过 map 的返回值我们可以获取该顶点的编号。

4) Mesh 时出现空洞的问题：是因为在提升之前未对要提升成椭圆曲线的边进

行排序，造成不相邻的两条曲边构成了曲面，因此就会出现空洞现象。

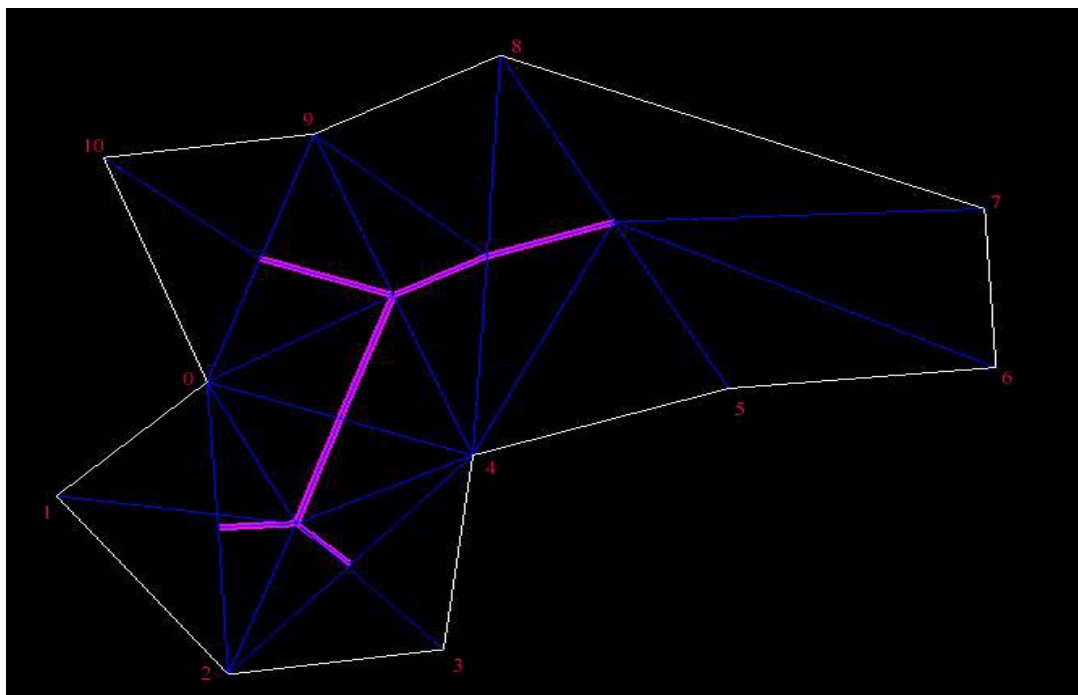


图 5-2 外顶点在输入记录时，就已按顺序排列。

解决思路：因为我们在输入外顶点时，必定是按顺时针或逆时针依次添加的如图 5-2 所示。因此我们只要对每一个外顶点指向骨架顶点的边的集合进行一致顺序的局部排序，就可以使所有外顶点指向骨架顶点的边的集合进行整体排序。

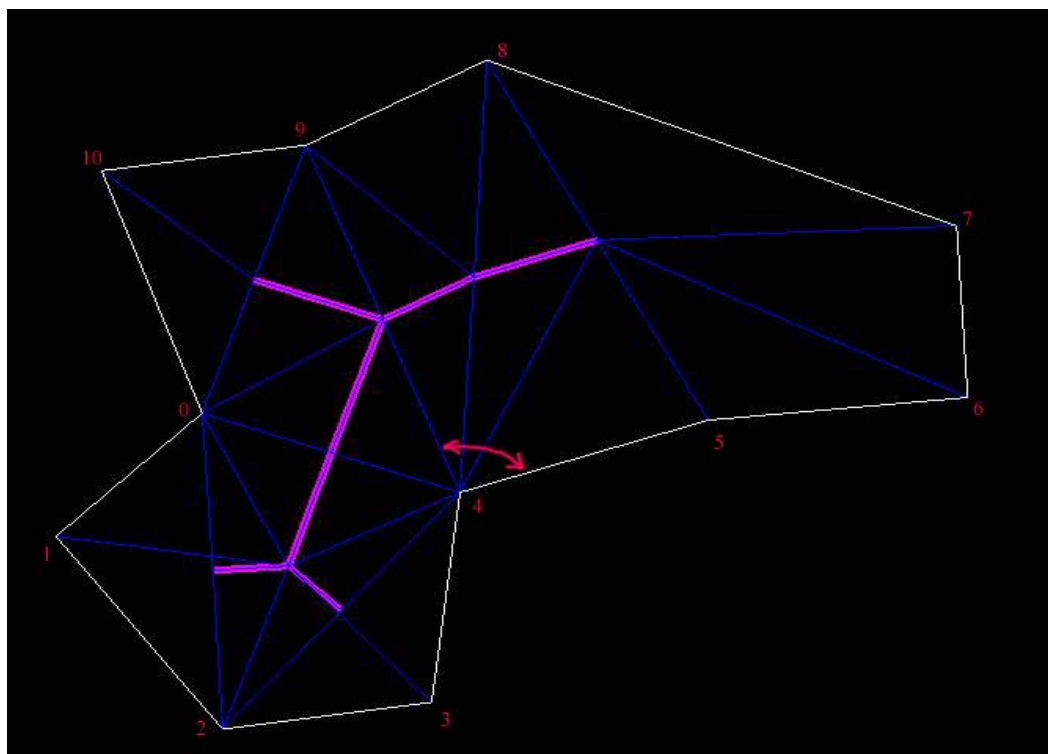


图 5-3 计算内部边与外部边的夹角，根据夹角大小排序

具体方案：每个外顶点只有唯一一条从该顶点发出的外边（多边形的外边），通过计算该顶点发出的其他内部边与该外边的夹角，如图 5-3 所示，根据夹角的大小对所有内部进行排序就可以得到。对每一个顶点都进行这样的排序就最终得到按顺序给出的边。

6 实例生成过程

以下以橄榄球三维模型的生成过程为例，说明本算法所涉及的主要步骤，包括：1) 绘制多边形；2)取骨架；3)退化处理；4)再取骨架及细分；5)提升生成空间网格；6)生成模型实现。如图 7-1 所示。

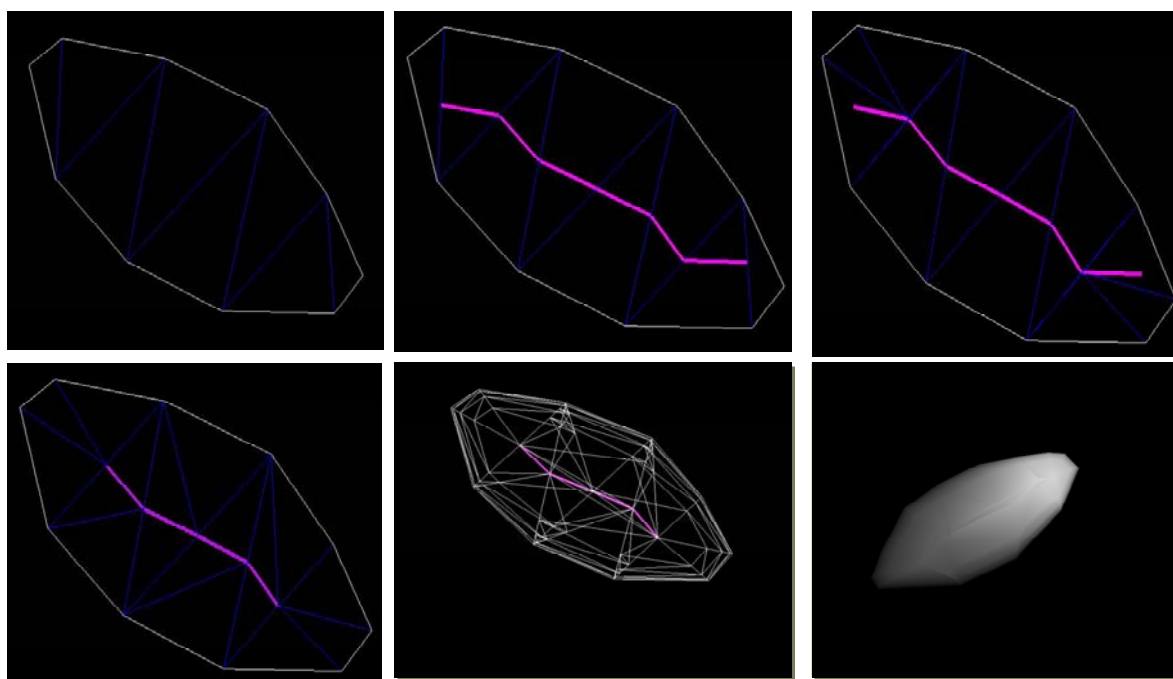
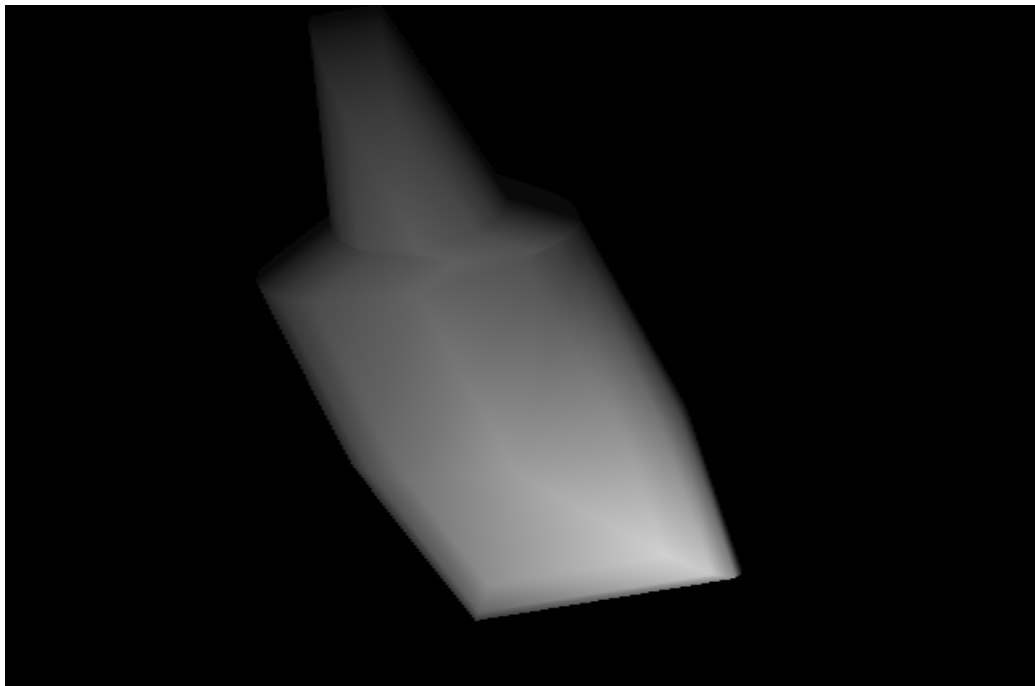
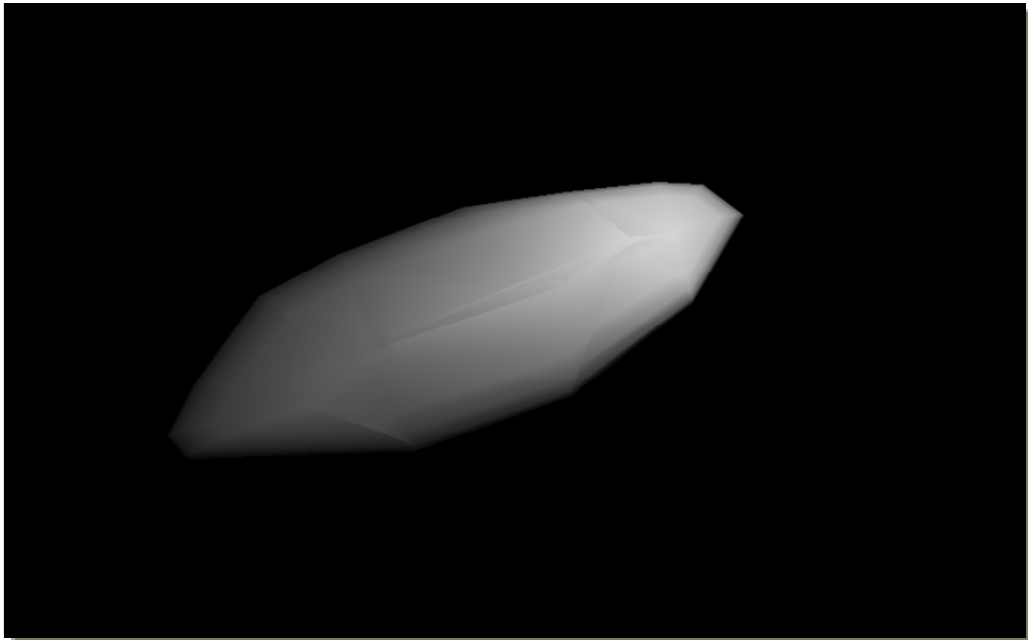


图 7-1 橄榄球橄榄球三维模型的生成过程

7 生成的三维模型图



8 未来工作

1) 提出基于骨架位置变化的新的提升方法。整个升起的骨架线是水平的，在很多情况下，是不合理的。首先，我们可以考虑把骨架线做成中心高，两侧下滑的样式。其次，提升的椭圆曲线改为多种可选择的曲线方式。

2) 对面积太大的三角面片进行细分。三角面片太大就会使生成的三维模型很粗糙，细分会使模型更加细致。

参考文献

- [1] Takeo Igarashi, Satoshi Matsuoka, Hidehiko Tanaka. Teddy: A Sketching Interface for 3D Freeform Design. ACM SIGGRAPH 1999,409-416
- [2] L. Prasad. Morphological analysis of shapes. CNLS Newsletter, 139: 1-18, July 1997.