

文章编号: 1003-5850(2006)12-0044-03

一种 3D 草图设计的实现和研究

Implementation and Research of 3D Sketching Design

王 鹤

(上海交通大学汉芯科技有限公司 上海 200030)

【摘 要】基于自由手绘的 3D 草图工具极大地简化了设计。对这种草图进行直接模型创建是计算机图形学的一个重要研究方向。实现了一个简单的 3D 物体的创建工具。系统实现基于 Teddy, 在此基础上, 在一些地方的算法作了多种尝试和优化。

【关键词】自由手绘, 3D 草图, 优化设计

中图分类号: TP391.72

文献标识码: A

ABSTRACT 3D Freeform sketching greatly simplifies the process of design. Modeling directly on sketch is an important in computer graphics. This paper implements an easy-handling tool of 3D objects construction. It's based on Teddy and did some noticeable improvement.

KEYWORDS Freeform based system, 3D sketching, optimize design

1 模型建立的实现

系统参考了 Teddy 中的算法原型。实现过程中, 发现了很多需要商榷的地方。同时, 在具体的步骤上, 对一些算法作了比较和分析, 有的还作了改动。

整个过程可分为以下几个阶段:

采集用户输入作为侧面轮廓, 用离散的点来代替轮廓。然后连接这些点, 得到的多边形将是最终生成的三维多边形表面的侧面轮廓。

找到其中的骨架。

根据骨架点的周边情况完成膨胀。

1.1 轮廓取点

1.1.1 算法介绍

首先接受一条没有自交叉的闭合曲线作为基础轮廓。这里, 把整个轮廓取点的工作分为几个步骤:

首先把用户输入的二维曲线用一些离散点来代替。

用直线把相邻点连接起来, 把最后一个点与第一个点连接起来, 强制闭合。

1.1.2 算法研究

用离散点形成的多边形最大程度地拟合外轮廓。

在此有三种可选方案:

根据一定长度取点。设置最短长度, 每到一个点与上一个点之间的距离超过这个长度, 则记录这个点。此方案便于控制精度, 同时去掉输入时的“噪音”, 即因抖动而造成的参差不齐。缺点是对于轮廓精细和粗略

的地方采集精度是相同的, 造成了浪费。

根据时间片取点。设置一个时间片, 每隔一段时间取一个点。由于用户在画精细部分时必然比画粗略部分时时长, 这样在精细部分采点多, 粗略时采点少, 精度根据轮廓精细程度而变, 更好描述轮廓的同时还节约资源。缺点是由于用户输入速度的不同, 时间片难以把握; 再者, 这样的采集方法不能去除“噪音”。

根据鼠标速度取点。这是结合前两个因素考虑的方法。鼠标速度小的时候取点密集, 而鼠标速度快的时候取点稀疏。这个方案的优点跟时间片的方法是相同的, 并且更加合理。但仍然不能去除输入当中的“噪音”。

综合比较后, 笔者选择了第一种采集方法。因为它能“去噪”, 而且便于膨胀。

1.2 寻找骨架

1.2.1 Constrained Delaunay Triangulation

Constrained Delaunay Triangulation (CDT) 算法是一种对多边形三角化的算法。对一个多边形的 CDT 就是把这个多边形退化成为一个个三角形。其中跟任何一个三角形任意两顶点连接的原多边形的顶点, 都不能出现在这个三角形的外接圆中。

然后对边和三角形分类。轮廓点称为外点, 轮廓边称为外边; 剩下的边称为内边。对于有两条外边的三角形, 称为 T 三角形, 有一条外边的三角形称为 S 三角形, 没有外边的三角形称为 J 三角形。

* 2006-06-07 收到, 2006-10-10 改回

** 王 鹤, 男, 1982 年生, 2005 年毕业于浙江大学, 研究方向: 软件工程。



1.2.2 找到基本骨架

多边形被CDT化之后,开始寻找骨架。遍历三角形,在T三角形中连接内边中点和其相对顶点;在S三角形中,连接两条内边中点;在J三角形中,是锐角三角形,则连接三条边的中点到它的外心,否则,则连接最长边的中点到其他各边的中点。连接完毕后,基本的骨架就出现了,如图1所示。

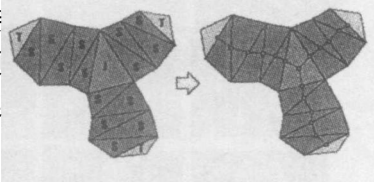


图1 左边:CDT化之后 右边:找到骨架

1.3 剪枝

膨胀前,把不重要的末端骨架支剪除掉:

设X是T三角形。以X的内边作直径在同侧作半圆。检查是否有X的顶点在半圆外。有,则停止剪枝;没有,则删除这条内边,把X和其内边另一侧的三角形合并。

如果新合并的三角形是一个S三角形,则X拥有三条外边和一条内边。按上述方法作半圆。检查是否有X的顶点在这个半圆外。有则停止,否则继续这样的过程。直到新合并的三角形是一个J三角形。

如果X有顶点落在半圆外,则以此时的内边中点为中心,分别连接X的各个其他顶点,形成像扇子一样的形状;如果新合并的三角形是一个J三角形,那么就以此三角形的外心为中心连接X的各个其他顶点,也形成一个像扇子一样的形状。

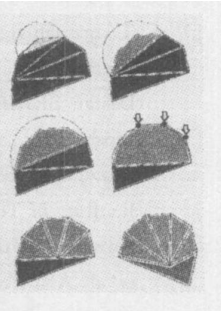


图2 剪枝过程

对每一个T三角形进行同样的处理过程,效果如图2所示。再对骨架和其周边进行一次三角化。整个过程如图3所示。

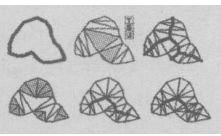


图3 从输入到剪枝完成

1.3.1 算法研究

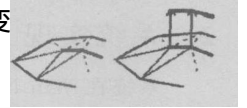
Teddy剪枝算法是遇到J三角形,则用三角形的外心连接X各顶点。而实现中发现如果这J三角形为钝角三角形,则取点会造成骨架点在这种三角形中过于密集,并使对应外心膨胀高度不够。所以本系统多加了一次判断:如果这个J三角形为锐角三角形,则算法不变;否则,则用最长边的中点来代替三角形的外心。

1.4 膨胀

1.4.1 算法介绍

每个骨架点的周围各点,称作周边点。根据骨架点

与其周边点的平均距离在Z轴方向上抬起这个骨架点,同时,在对称的另一边也是如此。新的两个点称为膨胀点。这样,骨架就从二维初步变成了三维的了,如图4所示。



1.4.2 算法研究

Teddy描述的膨胀算法,效果并不好。这一点,也在多篇其他论文中得到了印证。这个算法会在轮廓比较空的区域,膨胀陡然增高。

所以本系统加了一个调整算法。在第一次膨胀之后,再次把每个骨架点的膨胀点的高度与其父节点与子节点的平均值作一个比较。给定一个最大差值,如果高度与平均值的差值大于这个最大差值,就要进行一些“截长补短”的工作。这样有所改观,但还不能完全达到平滑的效果。

1.5 缝合表面

1.5.1 算法介绍

对每一个骨架点选其中一个膨胀点,然后用四分之一椭圆弧线来连接这个膨胀点与其骨架点的每个周边点。再对相邻的两条弧线之间进行“缝合”。然后把这些表面映射到另一边,就是一个完整的多边形表面了。



图5 缝合表面

效果如图5所示。

1.5.2 算法研究

对于这些四分之一椭圆弧线,尝试过三种实现:

按照长度取点。设定最小长度。当计算到弧线上的某一点跟前一点的距离到达这个长度,就采集该点。此方案使弧线上点的分布均匀,很好的拟合弧线;缺点是,弧度大的地方,在平面方向上就会显得点采集很密集,造成浪费;算法的开销也很大;且不同长度的边点采集数量不同,会给缝合带来不便。

按照角度取点。设定一个最大角度。对于每一个椭圆弧线,其中心点就是对应的骨架点。每当找到一个点,与中心点连线与前一个被采集点与中心点的连线之间的夹角达到这个角度,就采集该点。这样,弧线上点分布相对均匀;缺点是如果弧线本身长度很长,则不能很好的拟合。

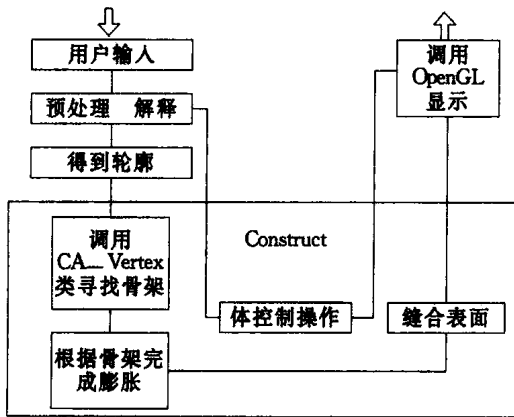
按照固定数量取点。设定一个固定数量,在XY平面上平均取值。对于每一条弧线,对其在XY平面上的投影平均分成所设的数量来取点,再取相对应的弧线上的点。这样一来精度可以自己控制,并且在算法上的开销也不大。缺点是对于长的弧线只能用增加精度的办法来拟合,同时在短一些的弧线上就造成了采点



过于密集。本系统在分别比较了效果之后最后选择了第三种。

2 系统实现

系统在 Win NT 环境下,用 OpenGL 作为三维引擎,C++ 实现。系统流程图如图 6 所示:



2.1 使用说明

用户在“画布”上任意画一条没有自交叉的闭合曲线。系统会自动根据用户所画轮廓来建造三维模型如图 7。

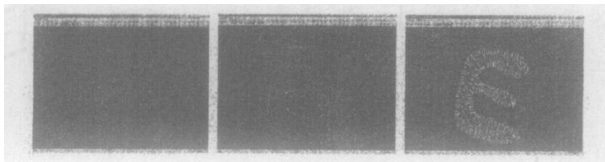


图 7 创建物体过程

用户也可以通过按住右键来看从不同角度来模型如图 8。

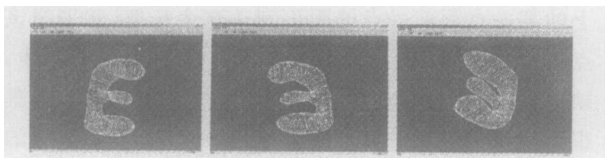


图 8 旋转

系统也提供了移动与放大缩小的功能如图 9。

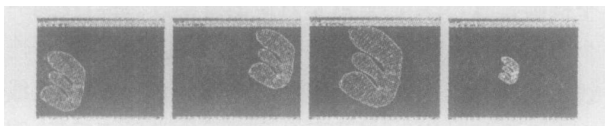


图 9 移动与放大缩小

利用现有简单的功能,只要发挥创意,也能做出一些简单但是有趣的模型。下面是一些用以实现系统创建的物体如图 10 所示。

3 工作展望

本系统还远没有达到一个可用的完备的系统。今

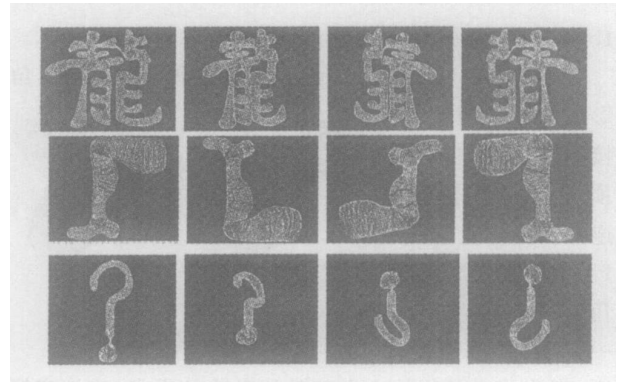


图 10 一些用“黑熊”生成的形体

后还有很多值得继续研究的地方:

对多物体的支持。这样就要研究多个物体之间的空间关系,才能有机地将它们组合起来。组合之中还可以加入布尔运算等操作。

目前的膨胀算法主要是针对圆形膨胀的物体。要画出棱角分明的物体,就要在创建模型之初选择以什么样的方式来建造模型。这种选择方式要以定义手势的方式来解决。

增加对已有模型的修改操作。

参考文献

- [1] T Igarashi, S Matsuoka, H Tanaka Teddy: A Sketching Interface for 3D Freeform Design. SIGGRAPH, 1999
- [2] Lipson H. Computer Aided 3D Sketching for conceptual design[D]. Israel, Haifa: The Israel Institute of Technology, 1998: 37-47.
- [3] Hiroaki NISHINO, Hideyuki TAKAGI, Sato SAGA, et al A Virtual Modeling System for Intuitive 3D Shape Conceptualization, 2002
- [4] Thomas F Stahovich, Randall Davis, Howard Shrobe Turning Sketches into Working Geometry. ASME Design Theory and Methodology 1995, 603-611.
- [5] 潘文鹤,董金祥,陈德人. 计算机图形学——原理、方法及应用(修订版). 北京: 高等教育出版社, 2003.
- [6] Naya F, Conesa J, Contero M, et al Smart Sketch System for 3D Reconstruction Based Modeling. Lecture Notes in Computer Science (ISSN: 0302-9743). 2003, 2 733: 58-68.
- [7] Zeleznik R, Herndon, K, Hughes J. SKETCH: An Interface for Sketching 3D Scenes, Proc of ACM SIGGRAPH '96, 1996: 163-170.
- [8] H Lipson, M Shpitalni. A new interface for conceptual design based on object reconstruction from a single freehand sketch. Annals of the CIRP, 1995(441): 133-136.