

# z 缓冲区消隐算法的改进

生 滨<sup>1</sup> 李 东<sup>1</sup> 徐学敢<sup>2</sup>

<sup>1</sup>(哈尔滨工业大学计算机系, 哈尔滨 150001)

<sup>2</sup>(广东省汕尾港监局船检科, 广东 516600)

E-mail: sheng\_bin@263.net

**摘 要** z 缓冲区算法是一种比较简单的消隐算法,但算法需要占用相当大的系统资源,且计算量很大。扫描线算法是对 z 缓冲区算法的改进,该算法大大减少了对系统资源的占用,同时提高了算法的效率。

**关键词** z 缓冲区算法 扫描线算法 消隐

**文章编号** 1002-8331-(2001)23-0114-03 **文献标识码** A **中图分类号** TP301.6

## An Improvement to Z-Buffer Hidden Surface Removal Algorithm

Sheng Bin Li Dong Xu Xuegan

(Computer Science Department, Harbin Institute of Technology, Harbin 150001)

(Shan Wei bureau of harbor superintendence, Guang Dong province, Guangdong 516600)

**Abstract:** Although z-buffer method is a simple hidden surface removal algorithm, it needs considerable systemic resources and induces a lot of computing work. As an improvement to z-buffer method, scan-line method decreases the demand for systemic resources and improves the efficiency at the same time.

**Keywords:** z-buffer method, scan-line method, hidden surface removal

### 1 引言

z 缓冲区消隐算法是最简单的消除隐藏面算法之一。z 缓冲区是一组存储单元,每个单元对应于屏幕上的一个像素,同时也对应于帧缓冲区的一个单元,它们之间是一一对应的。z 缓冲区中每个单元的值是屏幕上相应像素点所反应的对象上点的 z 坐标值,所以每个存储单元的位数取决于图形在 z 方向上的变化范围(一般取 2,3 个字节)。z 缓冲区中每个单元的初值取 z 的最大值(假设采用左手坐标系),帧缓冲区每个单元的初值则取对应背景颜色的值。图形消隐和生成的过程就是向 z 缓冲区和帧缓冲区相应单元填值的过程:对于要显示对象上的一点 A,比较 A 点的 z 坐标值与 z 缓冲区中相应单元的值,若前者小于后者,则将 z 缓冲区中相应单元的值改为 A 点的 z 坐标值,同时帧缓冲区中相应单元的值也要改为 A 点的颜色值;若前者大于后者,则说明屏幕上相应像素已经显示了对象上某个点的颜色,该点比 A 点更接近观察者,于是 z 缓冲区和帧缓冲区相应单元的值均不改变。对于要显示对象每个面上的每一点都做上述处理后,便可得到消除了隐藏面的图。

z 缓冲区消隐算法的优点是简单、可靠,不需要对显示对象的面预先进行排序,但缺点也很突出,即需要很大的缓冲区(对于一个 1024x1024 分辨率的系统,需要一个容量超过 100 万个单元的缓冲区,且每个单元需要有足够的位数来表示深度值),同时在显示对象表面的每一个点处都要计算深度值,因而工作量很大。

### 2 扫描线 z 缓冲区算法

要克服 z 缓冲区算法占用内存过大的缺点,可把整个屏幕

分成若干个区域,一区一区来显示,这样 z 缓冲区的单元数只需取屏幕上一个区域内的像素个数。当把这个区域取成屏幕上的五行时,就得到了扫描线算法,这时 z 缓冲区的单元数就是屏幕五行上的像素数。从最上面的一条扫描线开始,向下对每一条扫描线作处理:在 z 缓冲区每个单元中存放 z 坐标的最大值,把帧缓冲区每个单元置成背景颜色,对每个多边形检查它在 oxy 平面上的投影和当前的扫描线是否相交,若不相交,则不考虑该多边形;如果相交,对每对交点(扫描线和多边形边界的交点一定是成对出现)之间的像素计算多边形所在平面对应点的深度(即 z 值),并和 z 缓冲区中相应单元存放的深度值做比较,若前者小于后者,则 z 缓冲区中相应单元内容要被前者代替,帧缓冲区相应单元的内容也要换成相应的颜色值。当对所有的多边形都做上述处理后,帧缓冲区中这一行的值就反应了消隐后屏幕五行上的图形,对帧缓冲区每一行的单元都做上述处理后也就得到了整个消隐后的图形。

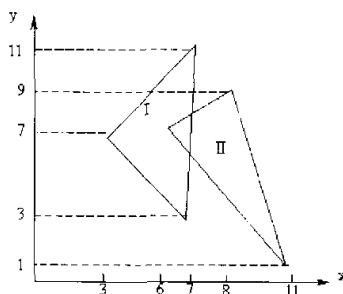


图 1 要消隐的对象

**基金项目:** 哈尔滨工业大学科研基金资助

**作者简介:** 生滨,男,1976 年出生,哈尔滨工业大学计算机系硕士研究生。李东,男,1967 年出生,哈尔滨工业大学计算机系副教授。研究方向:并行计算,计算机图形学等。

114 2001.23 计算机工程与应用

上述算法在处理每一条扫描线时,要检查各多边形是否和该扫描线相交,还要计算多边形所在平面上各点的 $z$ 值,这些都要花费很多时间。为了使这些工作能高效率地进行,可采取下列措施:

(1)建立一个多边形 $Y$ 桶和一个边 $Y$ 桶。图2和图3就是相应于图1的多边形 $Y$ 桶和边 $Y$ 桶。每个桶的深度和屏幕显示行数相同。根据多边形顶点 $Y$ 坐标的最大值来决定放入多边形 $Y$ 桶的相应行数,根据边两端点中较大的 $Y$ 坐标值来决定放入边 $Y$ 桶的相应行数。多边形 $Y$ 桶要记录多边形所在平面的方程 $ax+by+cz+d=0$ 的系数 $a, b, c$ 和 $d$ ,还要记录和多边形在 $oxy$ 平面上的投影相交的扫描线的条数 $\Delta y$ ,以及多边形的颜色值 $color$ 和编号 $IP$ 。边 $Y$ 桶中每一条边要记录下列信息:与该边在 $oxy$ 平面上的投影相交的扫描线条数 $\Delta y$ ,该投影与相邻的两条扫描线交点的 $x$ 坐标的差 $\Delta x$ ,该边所属多边形的编号 $IP$ 及边的上端点的 $x$ 坐标。

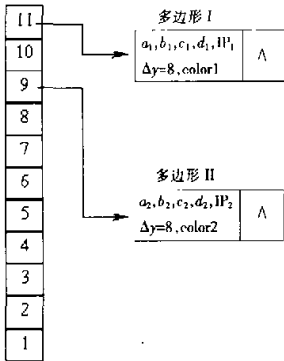


图2 多边形 $Y$ 桶

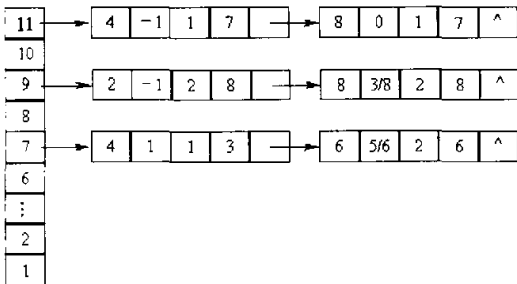


图3 边 $Y$ 桶

(2)建立一个多边形活化表和一个边活化表。多边形活化表中记录在 $oxy$ 平面上投影与当前考虑的扫描线相交的多边形。在图1中,当扫描线对应 $y=9$ 或 $10$ 时,多边形活化表只有一个多边形;当 $y=8$ 时,多边形活化表如图4所示(表中的 $\Delta y$ 值是经过修改的)。边活化表中记录多边形边界与扫描线相交的边对。在图1中 $y=6$ 的扫描线上边活化表中有两个边对:一个是与多边形I在 $oxy$ 平面上投影相交的两条边,另一个是与多边形II的投影相交的两条边。每一边对要记录下列信息:

$x_l$ :左交点的 $x$ 坐标值;

$\Delta x_l$ :左交点所在边和相邻两扫描线交点的 $x$ 坐标之差;

$\Delta y_l$ :以和左交点所在边相交的扫描线条数为初值,以后每处理一条扫描线减1;

$x_r$ :右交点的 $x$ 坐标值;

$\Delta x_r$ :右交点所在边和相邻两扫描线交点的 $x$ 坐标之差;

$\Delta y_r$ :以和右交点所在边相交的扫描线条数为初值,以后每处理一条扫描线减1;

$z_l$ :左交点处多边形所在平面的深度值;

$\Delta z_l$ :沿扫描线向右走过一个像素时,多边形所在平面深度的增量,对方程为 $ax+by+cz+d=0$ 的平面来说 $\Delta z_l=-a/c(c \neq 0)$ ;

$\Delta z_r$ :沿 $y$ 方向向下移过一根扫描线时,多边形所在平面深度的增量,对方程为 $ax+by+cz+d=0$ 的平面来说 $\Delta z_r=b/c(c \neq 0)$ ;

$IP$ :所在多边形的编号。

图5是图1中 $y=6$ 处的边活化表,其中 $\Delta y_l, \Delta y_r, x_l, x_r$ 均经过修改,修改方法将在后面详细说明。

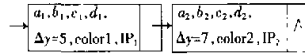


图4 多边形活化表

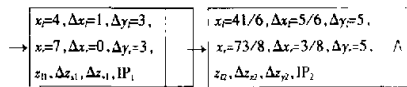


图5 边活化表

首先从最上面的一条扫描线开始处理(在处理之前,边和多边形的活化表均是空的)。对于图2和图3中的两个 $Y$ 桶,在处理每条扫描线时要做下列工作:

(1) $z$ 缓冲区的各单元放 $z$ 的最大值;

(2)把帧缓冲区相应行置成背景颜色;

(3)检查多边形的 $Y$ 桶,若有新的多边形涉及该扫描线,则把它放入多边形活化表中;

(4)若有新的多边形加入多边形活化表,则要把该多边形在 $oxy$ 平面上的投影和扫描线相交的边对加入边活化表中;

(5)如果某些边在该扫描线处结束了,但其所属的多边形仍在多边形活化表内,则可从边的 $Y$ 桶中找到该多边形在 $oxy$ 平面上的投影和扫描线相交的新的边或边对,修改或加到边的活化表中去(边对在活化表中的次序是不重要的)。

然后从边活化表中取出一个边对,令 $z_s=z_l$ ,对所有满足 $x_l \leq x \leq x_r$ 的坐标为 $(x, y)$ 的像素( $y$ 是当前考虑的扫描线高度),从左到右依次作下列处理:先计算 $z_s=z_l+\Delta z_l$ ,即对应像素处多边形所在平面的深度,若 $z_s$ 比 $z$ 缓冲区相应单元中存放的值小,则用 $z_s$ 代替 $z$ 缓冲区相应单元中的值,并把帧缓冲区中相应单元的内容改成这个多边形的颜色值。对边活化表中的所有边对都做上述处理后,帧缓冲区相应行中就保存了消隐后的结果。

下面说明对边活化表修改的方法(修改后的表就是下一条扫描线的边活化表)。对每一边对先计算 $\Delta y_r=\Delta y_r-1, \Delta y_l=\Delta y_l-1$ ,若 $\Delta y_l$ 或 $\Delta y_r$ 小于0,则相应的边就要从该边对中去掉,并从边活化表中找合适的边来代替。也有可能两条边同时结束于某一点,这时该边对就可以取消了。边和下一条扫描线交点的 $x$ 坐标值可由下式求得

$$x_r = x_l + \Delta x_l, x_s = x_l + \Delta x_r$$

多边形所在平面对应下一条扫描线在 $x=x_l$ 处的深度为

$$z_r = z_l + \Delta z_l \Delta x_l + \Delta z_r$$

对多边形活化表中每一个多边形的 $\Delta y$ 要做如下修改

$$\Delta y = \Delta y - 1$$

当  $\Delta y < 0$  时,该多边形就要从多边形活化表中删除。

完成这些处理后就可以开始对下一条扫描线做同样的消隐工作。当把每一条扫描线的消隐工作都完成后,整个图形的消隐工作也就完成了。

### 3 总结

扫描线算法是在  $z$  缓冲区算法的基础上,通过运用分治的思想,将消隐问题分散到每一条扫描线上去解决,减少了对系统资源的占用,使得问题变得简单化了。同时,由于在扫描线算法中引入了桶和活化表等数据结构,减小了计算量,算法的效率也随之大大提高。为了进一步说明问题,分别对两种算法进

行了测试,测试的结果如表 1 中所示。(收稿日期:2000 年 9 月)

表 1 两种算法计算时间的比较

算法 \ 多边形的数目	多边形的数目		
	100	2500	60000
$z$ 缓冲区算法	54	215	814
扫描线算法	11	64	307

### 参考文献

- 1.孙家广,杨长贵.计算机图形学(新版)[M].北京:清华大学出版社,1995
- 2.唐泽圣,周嘉玉.计算机图形学基础[M].北京:清华大学出版社,1995
- 3.Donald Hearn,M Pauline Baker.Computer Graphics,C version[M].2nd

(上接 63 页)

最后判断特大供应商客户:if trim(myrset("leixing"))="11" then 具体社区界面如图 7:



图 7

由特大供应商社区界面和一般供应商客户社区界面,可以看出特大供应商和一般供应商的区别就在于:特大供应商多了一个“议标区”,其他的功能都是一致的。

通过对权限的判断就实现了供应商客户不同权限的登录不同社区的功能。

(4)对分发用户的处理

else

Response.Redirect "../manage/sorry.htm"(重定向)

end if

(上接 113 页)

型以解决问题。首先在基于可执行 UML 的模型中找出问题的所在并改正;经过重新验证确定改正正确后,再修改系统的 UML 模型中对应的部分。

### 5 结束语

文章深入研究了基于 UML 的嵌入式系统模型的验证机制,提出了采用从系统的 UML 模型中抽象出可执行 UML 定义来进行仿真的验证方法。采用该方法能够在模型实现前尽早地发现系统的设计问题,促进 UML 在嵌入式系统中的应用。

此外,由于基于可执行 UML 的模型是形式化的,详细地描述了系统的动态行为属性,基于该模型考虑特定的硬件平台生

这样,供应商客户登录社区的功能就完成了,既对非法用户拒绝登录,也使不同权限的用户进入属于自己的社区,然后行使自己的权利。

### 5 结束语

文章是国家自然科学基金资助项目“基于 Internet 的招标投标新模式的研究”原型系统——网上招标采购管理系统研究的总结。供应商管理系统是网上招标采购管理系统的核心,它对供应商的产品目录、供应商的分类、供应商的评价等进行全面的管理,为买方和卖方提供了一个快速寻找机会、快速匹配业务和快速交易的基础。文章在网上招标采购管理系统分析研究的基础上,对供应商管理系统的功能进行结构上的分析,对主要模块的功能进行了设计,叙述了系统的开发环境,最后给出了一个具体功能的实现。文章的研究对其他类似系统的研发可起参考借鉴的作用。(收稿日期:2001 年 9 月)

### 参考文献

- 1.鲁耀斌等.Internet 环境下招标采购系统研究[J].计算机工程与应用,2001;37(23):57-60
- 2.鲁耀斌等.基于客户/服务器体系结构的大型建设项目招标投标管理系统的设计与实现[M].国防工业出版社,1997
- 3.二味工作室.网页设计超级 COOL—ASP[M].第一版,人民邮电出版社,2000.8
- 4.三味工作室.网页设计超级 COOL—Dreamweaver[M].第一版,人民邮电出版社,2000.8
- 5.清汉计算机工作室.ASP 开发实例[M].第一版,机械工业出版社,2000.11
- 6.Steve Banick,Michael Morrison.科欣翻译组译.Visual InterDev6 开发使用手册[M].第一版,机械工业出版社,1999.5

成完整的可以运行的程序代码也是完全可行的,这将是下一步研究的重点。(收稿日期:2001 年 4 月)

### 参考文献

- 1.邵维忠,梅宏.统一建模语言评述[J].计算机研究与发展,1999;4
- 2.王云,刘又斌,周伯生.UML 可视化建模系统的模型一致性检查机制[J].计算机研究与发展,2000.1
- 3.William E McUmbur,Betty H C Cheng.UML-Based Analysis of Embedded Systems Using a Mapping to VHDL[J].IEEE High Assurance Software Engineering,1999.11
- 4.Massimiliano Chiodo et al.Hardware-Software Codesign of Embedded Systems[J].IEEE Micro,1994.8