一种基于几何变换的高效的线裁剪新算法*

汪灏泓 吴锐迅 蔡士杰

(南京大学计算机软件新技术国家重点实验室 南京 210093)

摘要 线裁剪是计算机图形学的重要基础问题之一.在对现有的两种优秀算法作了分析之后提出一种利用简单几何变换,将裁剪问题简化为对两种基本情况的处理,并先后对被裁剪线段的首末端点作变换处理的新算法,有效地克服了上述两种方法中存在的调用函数多、基本情况处理复杂等弱点.理论分析和实例测试均表明,该算法优于当前国际最快的几种裁剪方法.

关键词 几何变换,线裁剪,算法.

中图法分类号 TP391

线裁剪是计算机图形学基础理论的重要内容之一,经过许多年的不断发展,人们已提出了多种可行的解决方案.[1~10]文献[1,2]先后提出并发展了分区编码法,通过确定线段端点的编码以判别线段与窗口的关系,然后作相应的处理;文献[3,4]以参数形式描述被裁剪线段,并通过参数返回来判别并计算出裁剪结果;文献[5~7]分别对平行截断法作了尝试,充分利用矩形裁剪窗口边界线平行于坐标轴的特点,分别沿两组平行边界线对线段进行裁剪.

T. M. Nicholl 等人在 1987 年首次提出了利用几何变换的思想进行线裁剪[8],介绍了一个独立于机器的统一的算法评估标准,即累计理论上的运算量,还提出了一个具有理论最小运算量的 NLN(Nicholl-Lee-Nicholl)算法.该文在二维图形学裁剪理论研究上作出了重要的贡献.1992 年,AS(Andeev-Sofianska)算法[9]对几何变换方法作了进一步发展.

尽管 NLN 算法与 AS 算法在理论上均具有较少的运算量,但在实际运行中,由于它们对基本情况处理复杂,导致子程序多,从而造成大量的参数传递和较长的程序,在相当程度上影响整个系统的效率.本文提出了一种新的几何变换方法,有效地避免了上述算法的弱点.它不仅在理论运算量上优于 NLN 算法和其他算法,通过实例验证,该方法的运行效率也超过目前国际上的最新算法.

1 几何变换模型

所谓几何变换模型,是指该类算法利用裁剪窗口的对称性.通过几何变换,将被裁剪线段与窗口的位置关系转化为几种基本关系之一,然后作相应的处理.几何变换是该模型最为重要的特征,在 NLN 算法与 AS 算法中,主要采用了以下 6 种变换方式:

- 逆时针旋转 90°(Rotate90),将点(a,b)变换至$(-b,a)$处.
- 逆时针旋转 180°(Rotate180):将点(a,b)变换至$(-a,-b)$处.
- 逆时针旋转 270°(Rotate270):将点(a,b)变换至$(b,-a)$处.
- X 轴对称变换(ReflectionX):将点(a,b)变换至$(a,-b)$处.
- Y 轴对称变换(ReflectionY):将点(a,b)变换至$(-a,b)$处.
- 关于直线$x=-y$的对称变换(ReflectionXY):将点(a,b)变换至$(-b,-a)$处.

如图 1 所示,裁剪窗口将平面划分为 9 个区域,依次编号为 0~8,并将它们定义为 3 类:窗口、边域和角域.在几何变换模型中,通常先要确定一个边域和角域为基本区域,这里不妨假设为区域 6 和 7(NLN 中为区域 0 和 3).使用基本变换可将任意区域(窗口除外)中的点映射至基本区域内(如图 2 所示).这样,原裁剪要求被简化为首端点在基本区域中时的处理问题.完整的算法由 3 步组成:(1) 判别首端点所在的区域,并将其变换至基本区域(另一端点也作同样

---

* 作者汪灏泓,1973 年生,博士生,主要研究领域为计算机图形学,可视化.吴锐迅,1974 年生,理学学士,主要研究领域为计算机图形学.蔡士杰,1944 年生,教授,博导,主要研究领域为计算机图形学,CAD,用户接口,可视化.

本文通讯联系人,蔡士杰,南京 210093,南京大学计算机软件新技术国家重点实验室

本文 1997-06-12 收到原稿,1997-09-22 收到修改稿

变换);(2) 求出裁剪后的新端点;(3) 将新端点作逆变换映射至原来的位置.

0 角域	1 边域	2 角域
3 边域	4 窗口	5 边域
6 角域	7 边域	8 角域

图1　窗口对平面的划分

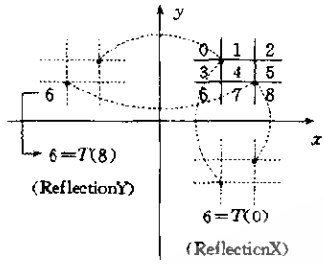

图2　基本几何变换例

在 NLN 算法中,对简化后的基本情况的处理是,将首端点与裁剪窗口的四角点连线,从而将平面划分为若干个区域,根据线段的斜率与边线斜率的大小比较结果,可确定出线段所跨越的区域和边界,然后求出线段与它们的交点,即为新的有效端点.显而易见,该算法中的多次斜率计算必然包含一些额外的除法开销,尽管 NLN 将部分除法转化为乘法运算,这仍使它在许多情况下具有多余的乘(除)法,特别是当裁剪线段二端点跨越较多的区域和边界时.AS 算法具有理论上优于 NLN 算法的效率.它简单地根据末端点所在的位置和线段所经过的边界进行分类,共有 3 大类、18 种情况.AS 算法对每种情况制定出相应的处理方案,这必然导致其包含大量的子程序,从而严重影响了实际的运算效率.本文在 NLN 与 AS 基础上提出了一种新算法,简称 ELC(efficient line clipping)算法,它对基本情况作了进一步化简,其目标在于降低程序的复杂度,减少系统的子程序调用,从而提高效率.

2　算法描述

ELC 算法先后对线段的两端点分别作变换处理,以求降低基本情况的复杂度,这样,每次只需了解线段的首端点和走向,而无需关心末端点的具体位置.

2.1　基本情况

如图 3 所示,ELC 仅需考虑两类基本情况(a 和 b)、7 种可能性.基本情况的处理仅需求出由首端点出发进入裁剪窗口的第 1 个有效交点,或判断出无交点.假设裁剪窗口的两角点坐标为($XLeft$,$YTop$),($XRight$,$YBottom$),裁剪线段为 P_1P_2,则 ELC 算法中对基本情况的处理步骤描述如下.

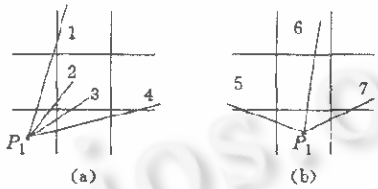

图3　两种基本情况

基本情况 a:

```
Procedure Basic_Case_a;
begin
  求出 P1P2与下边界的交点 Q;
  if (Q.X<XLeft) then
    begin                              /* Case1 or Case2 */
      求出 P1P2与左边界的交点 M;
      if (M.y>YTop) then               /* Case1 */
        该线段不可见
      else                             /* Case2 */
        P1=M;
    end
  else
      if (Q.x>XRight) then             /* Case4 */
        该线段不可见
```

```
        else
          P1=Q;                          /* Case3 */
    end;

    基本情况b:
    Procedure Basic_Case_b;
    begin
      求出线段与下边界的交点Q;
      if (Q.x<XLeft)or(Q.x>XRight) then   /* Case5 or Case7 */
        该线段不可见
      else                               /* Case6 */
        P1=Q;
    end;
```

2.2 ELC算法步骤

(a) 判别P_1所在区域,若P_1在窗口内,则转(e),否则,将其变换至基本区域(设变换为T);

(b) 对基本情况进行处理(如2.1节所述);

(c) 若线段不可见,则转(f);

(d) 将裁剪线段(此时P_1可能已被更新)作变换T^{-1};

(e) 若P_1在窗口内,则转(f),否则,将P_1与P_2互换,然后执行步骤(a)~(d);

(f) 算法结束.

3 算法分析与比较

3.1 评估标准

算法的比较必须建立在一个统一的评估标准之上.事实上,裁剪的速度总是与所使用的机型密不可分.在不同的机器上,各种基本运算的操作速度有很大差异(表1列出了不同机型运算操作差异的数据).因此,简单地用产生随机输入数据的方法来测试比较不同算法的速度,所得出的结论并不十分可靠,即使结果是真实的,也无法说明为什么一种算法比另一种快.[10]

表1 不同机型的基本运算所花费时间的比较[6] (CPU时钟)

操作	HP9000/340	Sun3/50	SunSPRACSLC
浮点:除法	1.00	1.00	1.00
乘法	0.86	0.73	0.51
加法	0.76	0.52	0.46
减法	0.76	0.56	0.46
比较	0.56	0.09	0.35

3.2 运算量比较

根据文献[8]的统计方法,我们将ELC算法与NLN算法[8]、SPY[2]算法、WLP(王-梁-彭)算法[7]、AS算法[9]作了比较.由图4~6可以看出:ELC算法在几乎所有情况下都比NLN算法、SPY算法、WLP算法的运算量少,但其比较运算略多于AS算法,这主要是由在处理斜率问题时对水平、垂直情况的判别所造成的,而这实际上也使水平、垂直的特例情况的处理运算得以减少.

易知,根据线段两端点的位置,共有81种可能性,如表2所示.这样,图4~6中各区域边上的分数即为该结果发生的概率.图中结果亦为几种可能情况取平均的结果.可以将各区域的运算量乘以概率进行累计,即为较合理的算法结果.只需将多种机器的基本运算数据代入,就可得到理论上的平均运算时间.本文采用该方法对表1所列的3种机器,将NLN算法与ELC算法作了比较,ELC算法均优于NLN算法.

表2　裁剪线段端点位置的81种可能分布

P_1 \ P_2	窗口	边域	角域
窗口	1	4	4
边域	4	4(r)+4(1)+8(s)	8(1)+8(s)
角域	4	8(1)+8(s)	4(r)+4(s)+8(1)

其中(r)表示两点位于同一区域,(1)表示两点位于同一水平或垂直方向上的两区域,(s)表示两点位于斜向的两区域

NLN	AS	SPY	WLP	ELC	NLN	AS	SPY	WLP	ELC	
8	8	15	15.88	8	7.5	7.5	14	7.5	7.5	
1	1.5	5.5	1.5	1.5	1	1	5	1	1	
4	3.5	3.5	3.5	3.5	3	3	3	3	3	
2	1	1.5	1	1	1	1	1	1	1	
1	1.5	1	1.5	1.5	1	1	1	1	1	
4/81					4/81					
					NLN	AS	SPY	WLP	ELC	
			compare		8	8	14	8	8	
			+		0	0	0	0	0	
			−		0	0	0	0	0	
			*		0	0	0	0	0	
			/		0	0	0	0	0	
					1/81		P_1			

图4　P_1在窗口中时的运算量分析

NLN	AS	SPY	WLP	ELC	NLN	AS	SPY	WLP	ELC	NLN	AS	SPY	WLP	ELC
3.5	3.5	14	3.88	3.5	7.5	6.5	15	10.63	6.25	7.5	6.5	15	10.63	6.25
0	0	4	0	0	0	1	5	1	1	0	1	5	1	1
0	0	0	0	0	4	3	3	3	3	4	3	3	3	3
0	0	0	0	0	2	1	1	1	1	2	1	1	1	1
0	0	0	0	0	0	1	1	1	1	0	1	1	1	1
(8/81)					(8/81)					(8/81)				
					NLN	AS	SPY	WLP	ELC	NLN	AS	SPY	WLP	ELC
			compare		8.5	8.5	15	15.88	8.75	9.5	8.5	16	17.88	10.125
			+		2	2	6	2	2	2	2.5	6.5	2.25	2.5
			−		4	4	4	4	4	5	4.5	4.5	4	4.5
			*		2	1	2	1	1	3	2	1.5	1.5	1.75
			/		2	2	1	2	2	2	1.5	2	2	1.75
NLN	AS	SPY	WLP	ELC	NLN	AS	SPY	WLP	ELC	NLN	AS	SPY	WLP	ELC
3.5	3.5	14	3.88	3.5	8	8.5	14	13.38	9.25	8	8.5	14	16.38	9.75
0	0	4	0	0	1	1	5	1	1	2	2	6	1.5	2
0	0	0	0	0	3	3	3	3	3	4	4	4	3.5	4
0	0	0	0	0	1	1	1	0.5	1	2	2	2	1	2
0	0	0	0	0	1	1	1	1.5	1	2	1	1	2	1
(4/81)		P_1			(4/81)					(4/81)				

图5　P_1在边域中时的运算量分析

NLN	AS	SPY	WLP	ELC
2.5	2.5	14	3.88	2.5
0	0	4	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
(4/81)		P_1		

NLN	AS	SPY	WLP	ELC
3.75	3.75	14	3.88	3.75
0	0	4	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
(8/81)				

NLN	AS	SPY	WLP	ELC
3.75	3.75	14	3.88	3.75
0	0	4	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
(6/81)				

compare	NLN	AS	SPY	WLP	ELC
	8	9.5	15	14.38	10
+	1	1.5	5.5	1.5	1.5
−	4	3.5	3.5	3.5	3.5
*	2	1	1.5	1	1
/	1	1.5	1	1.5	1.5
	(4/81)				

NLN	AS	SPY	WLP	ELC
8	9.5	16	17.88	10.125
2	2.5	6.5	1	2.5
5	4.5	4.5	4	4.5
3	2	2.5	2	1.5
2	1.5	1	1	2
NLN	AS	SPY	WLP	ELC
9	9.5	15	18.88	10.125
2	2.5	6.5	2	2.5
5	4.5	4.5	5	4.5
3	2	2.5	2	1.5
2	1.5	1	2	2
(8/81)				
NLN	AS	SPY	WLP	ELC
8	8	15	14.88	8
0	2	5	1	1.5
5	4	3	4	3.5
3	1	1	2	1
0	2	1	1	1.5

NLN	AS	SPY	WLP	ELC
10	9.5	17.5	18.13	11
2	3	6.5	1.75	3
6	5	5	4.5	5
4	2	3	1.75	2
2	2	1	1.75	2
(4/81)				

图6　P_1在角域中时的运算量分析

3.3　算例比较

运算量分析是较合理的评估标准，但我们还应考虑一些实际运算时存在的其他因素，如子程序调用的次数、程序量大小等等.

为了进一步证明ELC算法的高效性，我们还将它与AS，NLN，WLP等算法在执行时间上作了比较. 我们用不同尺寸的裁剪窗口做了4组实验，每次用裁剪窗口对50根随机线段重复裁剪20 000次. 表3给出了在486DX/33上选取的4组随机数据的执行时间的比较. 表4中，Case1表示两端点均落在窗口外的同侧；Case2表示两端点均落在窗口内；Case3表示一端点落在窗口内，另一端点落在窗口外；Case4表示两端点均落在窗口外，且有可见部分；Case5表示两端点均落在窗口外，且无可见部分.

表3　不同算法在PC486上的执行时间　(s)

Group	WLP	NLN	AS	ELC
1	13	15	9	10
2	15	19	13	10
3	14	19	15	14
4	16	22	16	13

表4　线段数目与窗口的位置关系

Group	Case1	Case2	Case3	Case4	Case5
1	35	0	0	11	4
2	10	13	8	12	7
3	3	11	25	9	2
4	2	5	10	17	16

4 结 论

ELC算法通过首末端点二次几何变换及合理选择基本情况,大大降低了裁剪处理复杂性,其子程序调用数比NLN,AS算法减少了近50%.经过运算量及实例速度比较,证明该算法均优于当前国际上的一些流行算法.同时,该方法还较易于推广至三维线裁剪的处理.

参考文献

1 Foley J D, Dam A V. Fundamentals of Interactive Computer Graphics. Reading, MA: Addison-Wesley, 1983. 144～152

2 Sobkow M S, Poapisil P, Yang Y H. A fast two-dimensional line clipping algorithm via line encoding. Computers & Graphics, 1987,11(4):459～467

3 Cyrus M, Beck J. Generalized two- and three-dimensional clipping. Computers & Graphics, 1978,3(1):23～28

4 Liang Y D, Barsky B A. A new concept and method for line clipping. ACM Transactions on Graphics, 1984,3(1):1～22

5 Duvanenko V J, Gyurcsik R S, Robbins W E. Simple and efficient 2D and 3D span clipping algorithms. Computers & Graphics, 1993,17(1):39～54

6 Day J D. An algorithm for clipping lines in object and image space. Computers & Graphics, 1992,16(4):21～26

7 王骏,梁友栋,彭群生.具有最少算术运算量的二维线裁剪算法.计算机学报,1991,7(7):495～504
(Wang Jun, Liang You-dong, Peng Qun-sheng. A 2-D line clipping algorithm with the least arithmetic operations. Chinese Journal of Computers, 1991,7(7):495～504)

8 Nicholl T M, Lee D T, Nicholl R A. An efficient new algorithm for 2-D line clipping: its development and analysis. Computers & Graphics, 1987,21(4):253～262

9 Andeev R, Sofianska E. New algorithm for two-dimensional line clipping. Computers & Graphics, 1991,15(4):519～526

10 Sharma N C, Manohar S. Line clipping revisited: two efficient algorithm based on simple geometric observations. Computers & Graphics, 1992,16(1):51～54

A New Efficient Line Clipping Algorithm Based on Geometric Transformation

WANG Hao-hong WU Rui-xun CAI Shi-jie

(*State Key Laboratory for Novel Software Technology Nanjing University Nanjing* 210093)

Abstract Line clipping is one of the fundamental problems in computer graphics. In this paper, two well known algorithms are analyzed and a new algorithm, which simplifies the line-clipping into the processes in two basic cases by doing simple transformation on two end-points of a line in turn, is introduced. The algorithm effectively overrides the shortcomings existed in the above two algorithms such as too many subroutine calls and complex basic cases. Both theoretic analysis and example testing show that the new algorithm is better than well known algorithms.

Key words Geometric transformation, line clipping, algorithm.