

# 图形“裁剪与覆盖”的计算机辅助设计

姚涵珍 楚大庆 李彦启 张 键

(天津轻工业学院机械系 CAD 中心, 天津 300222)

**摘 要** 图形的裁剪与覆盖是计算机图形学中较为热点研究的问题, 作者对此进行了深入的研究与实践, 提出了较为理想的圆形窗口“裁剪与覆盖”的计算机辅助设计新算法。该算法利用计算机最简单的右移一位来取代原算法中大量的乘除运算, 从而加快了图形裁剪与覆盖的速度, 提高了计算机进行图形处理的能力。

**关键词** 裁剪与覆盖, 圆形窗口, “真、假”区

## 1 引 言

图形的裁剪与覆盖是计算机图学中的基本问题之一。国内外已发表了很多有关研究裁剪问题的文章, 提出了多种算法, 并且以不同的方式实现了这些算法。Cohen-Sutherland 编码算法、矢量裁剪算法、及平面多边形裁剪算法<sup>[1]</sup>; 梁友栋-Barsky 算法、Sutherland-Hodgson 多边形裁剪算法及图形求交集多边形裁剪算法<sup>[2]</sup>, 这些算法的裁剪窗口的形状多数为矩形窗口, 而我国工程图局部放大多用圆形窗口; 原算法需先分别对矩形窗口延伸后的 9 个区域进行判别, 然后再分别对矩形窗口的 4 边作大量的乘、除求交运算, 严重影响了计算机图形处理的速度, 而裁剪的核心是速度问题, 该文所研究的圆形窗口裁剪与覆盖算法是资料上未曾见过的新算法, 用“真、假区”判别及计算机简单右移一位来取代与矩形窗口 4 边大量的乘、除求交运算, 从而提高了计算机进行图形处理的速度, 也较适合我国绘制局部放大图的习惯。

## 2 算法研究

裁剪是把处于窗口外的图形剪掉, 即输出窗口内的线段; 覆盖是保留窗口外的图形, 将窗口内的图形相当于用一张白纸覆盖掉, 即输出窗口外的线段。

虽然它们表现形式是相反的, 但共同的目标必须取得线段与窗口的交点坐标, 所以裁剪与覆盖的计算机辅助设计核心是研究线段与窗口的求交算法问题。

### 2.1 “真、假区”算法

当始点在窗口外或窗口上, 过始点作窗口的切线, 切线将平面分成“真”“假”两个区域。圆形窗口一侧为“真”区, 异侧为“假区”。“真区”内的线段存在与窗口继续判别求交的可能; “假区”中的线段不存在与窗口求交的问题(图 1, 图 2)。下面以裁剪为例论述该算法与窗口的求交原理。

如果始点在圆形窗口内, 则进一步判别终点所在的区, 若终点也在窗口内, 则输出线段; 若终点在圆形窗口外, 则调用中点再分子程序, 求出交点后输出始点与所求交点之间的线段。

下面以  $A(X_1, Y_1) B(X_2, Y_2)$  线段为例来说明该算法的执行步骤。已知条件: 窗口圆心坐标  $CW(X_{cw}, Y_{cw})$  及窗口半径  $R_1$ 。

#### 2.1.1 快速预处理

(1) 为提高算法运算效率, 应迅速而准确地排除圆形窗口外切正方形之外的线段, 即线段满足下述 4 个条件之一:

$$\begin{aligned} \max(X_1, X_2) &< (X_{cw} - R_1); \\ \min(X_1, X_2) &> (X_{cw} + R_1); \end{aligned}$$

$$\max(Y_1, Y_2) < (Y_{cw} - R_1);$$

$$\min(Y_1, Y_2) > (Y_{cw} + R_1)$$

则线段不可能处于窗口内,过程结束,无线段输出。

(2)对有可能落在窗口内的线段的起点  $A(X_1, Y_1)$ 、终点  $B(X_2, Y_2)$  分别求出其到窗口圆心的距离  $S_1, S_2$  及线段  $AB$  的长度  $S_{12}$ 。

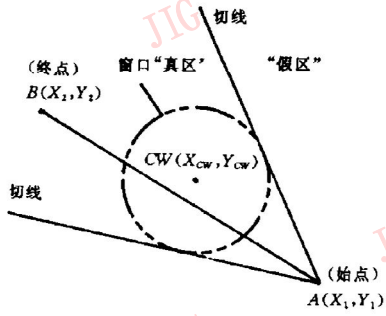


图 1 窗口与“真、假”区  
Fig. 1 Window and “true、false”

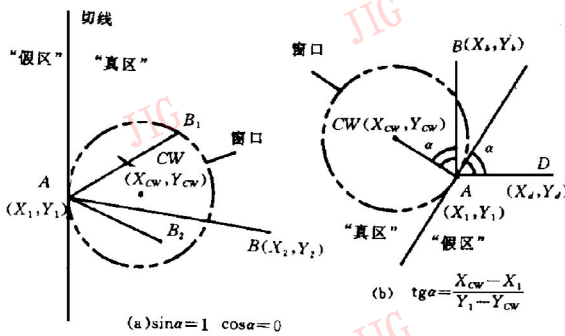


图 2 始点 A 在窗口上  
Fig. 2 Start point on the windows

口的切线,并计算出切线与水平方向夹角的正弦值  $S$  及余弦值  $C$  及其切线的斜率  $AK = (X_{cw} - X_1) / (Y_1 - Y_{cw})$

现以图 2a 及图 2b 中的  $B$  点来说明算法过程:由于终点  $B$  落在“真区”,则线段  $AB$  部分位于窗口内,利用中点再分原理进行线段与圆形窗口求交运算,即:求出中点→与窗口半径比较→舍弃一半→求中点→比较……直至中点与窗口重合,最后将窗口内可见线段起、终点坐标作“窗口—视图区变换”后,输出线段,过程结束。反之,若终点落在“假区”,(图 2b)中  $D$  点,无线段输出,过程结束。

2.1.3 若线段  $AB$  的起点满足  $S_1 < R_1$ , 则线段  $AB$  的起点一定在窗口内。而且  $AB$  线段必有部分落在窗口内。

2.1.2 若线段  $AB$  满足  $S_1 = R_1$ , 则始点  $A$  在窗口上。

(1)若  $S_2 \leq R_1$ , 则终点  $B$  在窗口上或窗口内,输出线段,见图 2 中的  $B_1, B_2$  点。

(2)若  $S_2 > R_1$ , 则终点  $B$  在窗口外,过起点  $A$  作窗

(1)若终点  $B(X_2, Y_2)$  与圆心距离  $S_2 \leq R_1$  则输出线段  $AB$  坐标值,供“窗口—视图区变换”用。

(2)若终点  $B(X_2, Y_2)$  与圆心距离  $S_2 > R_1$ , 则利用中点再分原理进行线段与圆形窗口求交运算(调用预设子程序),然后输出线段,过程结束。

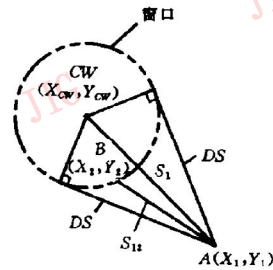


图 3 终点 B 在窗口上  
Fig. 3 Endpoint on the windows

2.1.4 若起点  $A(X_1, Y_1)$  与圆心距离  $S_1 > R_1$ , 则线段起点必在窗口外。

(1)若  $S_2 = R_1$ , 则终点  $B$  在窗口上,(图 3)。

(a)若  $S_{12} \leq DS$ , 则终点必在虚线上,而且无线段输出,过程结束。

(b)若  $S_{12} > DS$ , 则使用中点再分原理,求出线段与窗口交点,输出线段,过程结束。

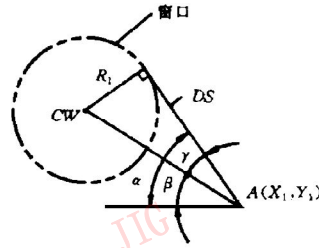


图 4 切线方向角  $\alpha = \beta + \gamma$   
Fig. 4 Tangent direction angle  $\alpha = \beta + \gamma$

(2)若  $S_2 < R_1$ , 则终点必在窗口内,类同(3)中(b)。

(3)若  $S_2 > R_1$ , 则线段终点  $B_1$  处在窗口外。(a)  $\alpha = \beta + \gamma$ , (图 4)。

其中,  $\alpha$ :切线与水平方向夹角  
 $\beta$ :  $S_1$  与水平方向夹角  
 $\gamma$ :切线与  $S_1$  夹角

先求出: 
$$\begin{cases} \sin\alpha = \sin(\beta + \gamma) \\ \cos\alpha = \cos(\beta + \gamma) \end{cases}$$

其中: 
$$\begin{aligned} \sin\gamma &= R_1 / S_1 \\ \cos\gamma &= DS / S_1 \\ \sin\beta &= (Y_{cw} - Y_1) / S_1 \\ \cos\beta &= (X_{cw} - X_1) / S_1 \end{aligned}$$

再调用判断线段  $AB$  的终点  $B$  的“真、假区”子程序,只有终点  $B$  在“真区”,才有进一步判断的必要,同下面(b)进一步判断;否则  $AB$  处在窗口外,无线段输出,过程结束。

(b)  $\alpha = \beta - \gamma$ , (图 5)。

先求出: 
$$\sin\alpha = \sin(\beta - \gamma)$$

$$\cos\alpha = \cos(\beta - \gamma)$$

其中:  $\sin\gamma, \cos\gamma, \sin\beta, \cos\beta$  同(a),再下面进一步判

断:即终点 B 在图 6 中的“真区”内,线段 AB 才有可能部分落在窗口内,否则过程结束,无线段输出。

下面是终点 B 在阴影区域内的进一步判断(图 6):

(1) 若  $S_{12} \leq DS$  时,线段全部在窗口外,无线段输出,过程结束。

(2) 若  $S_{12} > DS$ ,则线段 AB 与窗口处有两个交点。过窗口圆心  $CW(X_{cw}, Y_{cw})$  作线段 AB 的垂线(必在窗口内),求出线段的垂足坐标  $(XX, YY)$  及其斜

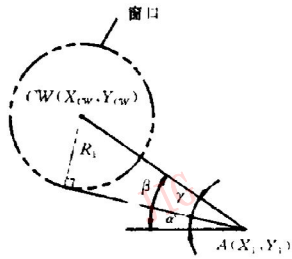


图 5 切线方向角  $\alpha = \beta - \gamma$   
Fig. 5 Tangent direction angle  $\alpha = \beta - \gamma$

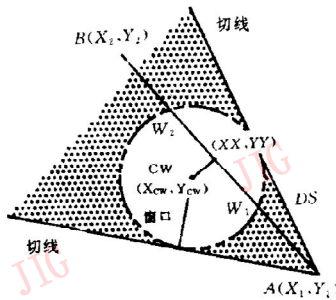


图 6 终点在“真区”  
Fig. 6 Endpoint on the “true field”

率 BK。先求出线段 AB 的斜率  $AK = (Y_2 - Y_1) / (X_2 - X_1)$  再求出  $BK = -1/AK$ ,最后求出垂足坐标:

$$XX = (AK * X_1 - BK * X_{cw} - Y_1 + Y_{cw}) / (AK - BK)$$

$$YY = AK * (XX - X_1) + Y_1$$

然后用中点再分原理,分别求出垂足与起点 A 及垂足与终点 B 组成的两线段与窗口的交点。经下一步算法处理后输出可见线段,过程结束。2.1.5 窗口—视图区变换

对前面 4 步中输出的可见线段起、终点(如:  $W_1$  及  $W_2$ )进行“窗口—视图区变换”(略)(图 7)。

如:将窗口内任意可见点  $W(X_{w1}, Y_{w1})$  变换成视图区中的映象  $S(X_{s1}, Y_{s1})$ 。

求出图形比例:  $RATE = R_2 / R_1$

$$XS = RATE * (X_{w1} - X_{cw}) + X_{cs}$$

$$YS = RATE * (Y_{w1} - Y_{cw}) + Y_{cs} \quad (1)$$

确定了窗口中心坐标  $(X_{cw}, Y_{cw})$  及其半径  $R_1$ ,视图区中心坐标  $(X_{cs}, Y_{cs})$  及其半径  $R_2$ ,那么前面所输出的

窗口内任意可见点  $W_1(X_{w1}, Y_{w1})$  可通过式(1)在视图区确定该点映象  $S(X_{s1}, Y_{s1})$  的具体位置,最终在输出设备的视图区内绘制窗口内的可见图线,过程全部结束。

### 2.2 参变量判别法

直线与圆求交方程经推导、简化得:  $at^2 + bt + c = 0$  的简单形式,然后用类似  $b^2 - 4ac < 0, = 0, > 0$  来判断直线与圆不相交、相切、相交 3 种可能,排除不相交与相切,对相交的情况再用直线与圆在交点处的参数  $t$  值来判断其需不需继续求交,最终(需求交的情况下)将  $t$  值代入直线方程,

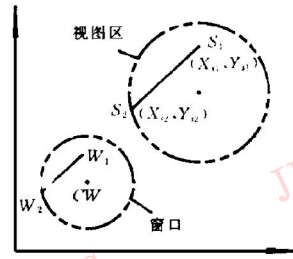


图 7 窗口—视图区变换

Fig. 7 Window-view revolution

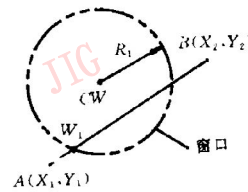


图 8 参变量与 AB 的关系

Fig. 8 Relation of parameter and AB

求得线段与圆的交点坐标。

#### 2.2.1 快速预处理

同 2.1.1 节中的(1),略。

#### 2.2.2 程序设计算法

(1) 直线段、直线、圆形窗口的定义:

$$\text{线段} \begin{cases} X = X_1 + (X_2 - X_1) * t \\ Y = Y_1 + (Y_2 - Y_1) * t \end{cases} \quad 0 \leq t \leq 1 \quad (2)$$

$$\text{直线} \begin{cases} X = X_1 + (X_2 - X_1) * t \\ Y = Y_1 + (Y_2 - Y_1) * t \end{cases} \quad -\infty \leq t \leq +\infty \quad (3)$$

$$\text{圆: } (X - X_{cw})^2 + (Y - Y_{cw})^2 = R_{12}^2 \quad (4)$$

(2) 参数  $t$  及判别变量  $WD$  的求解及推导

$$\text{令: } \begin{cases} dx = X_2 - X_1 \\ dy = Y_2 - Y_1 \end{cases} \quad \text{代入(3)}$$

$$\text{得: } \begin{cases} X = X_1 + dx * t \\ Y = Y_1 + dy * t \end{cases} \quad -\infty \leq t \leq +\infty \quad \text{代入(4)}$$

$$\text{得: } (X_1 + dx * t - a)^2 + (Y_1 + dy * t - b)^2 = R_1^2 \\ (dx^2 + dy^2) * t^2 + 2[(X_1 - X_{cw}) * dx + (Y_1 - Y_{cw} * dy)] * t + (X_1 - X_{cw})^2 - (Y_1 - Y_{cw})^2 - R_1^2 = 0 \quad (5)$$

$$\text{令: } \begin{cases} WA = dx^2 + dy^2 \\ WB = (X_1 - X_{cw}) * dx + (Y_1 - Y_{cw}) * dy \\ WC = (X_1 - X_{cw})^2 + (Y_1 - Y_{cw})^2 - R_1^2 \end{cases}$$

$$\text{即: } \begin{cases} WA = (X_2 - X_1) * (X_2 - X_1) + \\ \quad (Y_2 - Y_1) * (Y_2 - Y_1) \\ WB = (X_1 - X_{cw}) * (X_2 - X_1) + \\ \quad (Y_1 - Y_{cw}) * (Y_2 - Y_1) \\ WC = (X_1 - X_{cw}) * (X_1 - X_{cw}) + \\ \quad (Y_1 - Y_{cw}) * (Y_1 - Y_{cw}) - R_1 * R_1 \end{cases} \quad (6)$$

将式(6)代入(5)得:

$$WA * t^2 + 2WB * t + WC = 0$$

$$t = (-WB \pm \sqrt{WB^2 - WA * WC}) / WA$$

$$\text{令 } WD = WB^2 - WA * WC \quad (7)$$

(3) 程序设计途径

(a) 按式(6)计算出  $WA, WB, WC$

(b) 由  $WD$  判别直线  $AB$  与窗口的相对位置

按(7)求出  $WD$

若  $WD < 0$ : 直线与窗口不相交 } 无线段输出, 过程

若  $WD = 0$ : 直线与窗口相切 } 结束。

若  $WD > 0$  直线与窗口相交, 须作进一步的判断, (图 9)。

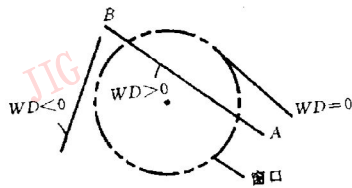


图 9 WD 判别形式  
Fig. 9 WD judgement

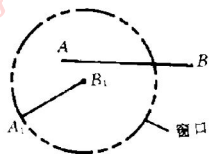


图 10 始点 A 在窗口上或窗口内  
Fig. 10 Start point A on or in window

(c) 计算

$$T_1 = (-WB - \sqrt{WD}) / WA$$

$$T_2 = (-WB + \sqrt{WD}) / WA \quad (T_1 < T_2)$$

由参数  $t$  值来判别直线  $AB$  与窗口交点在  $AB$  线段上的相对位置:

(I) 若  $0 \leq T_1 \leq 1$ , 并且  $0 \leq T_2 \leq 1$ , 则直线与窗口两交点在线段  $AB$  之间, 求交有效,

$$\text{按 } \begin{cases} X = X_1 + (X_2 - X_1) * T_1 \\ Y = Y_1 + (Y_2 - Y_1) * T_1 \end{cases} \quad \text{及}$$

$$\begin{cases} X = X_1 + (X_2 - X_1) * T_2 \\ Y = Y_1 + (Y_2 - Y_1) * T_2 \end{cases}$$

得窗口内线段两端点坐标值, 输出交点, 过程结束。

(II) 若  $WC \leq 0$  及  $0 \leq T_1 \leq 1$ , 求交有效, 输出交点, 如图 10 中的  $AB$  线段。

若  $WC \leq 0$  及  $T_2 \geq 1$ , 不必求交, 用  $B(X_2, Y_2)$  为输出可见点, 过程结束, 见图 10 中的  $A_1B_1$  线段。

(III) 若  $WC > 0$  则  $A$  点在窗口外, (图 11)。

若  $0 < T_1 < 1$ , 则求交有效,

按  $\begin{cases} X = X_1 + (X_2 - X_1) * T_1 \\ Y = Y_1 + (Y_2 - Y_1) * T_1 \end{cases}$  求得交点坐标, 并输出交点, 再转向  $T_2$  的判断, 同 (II)。如图 11 中的  $AB$  线段。

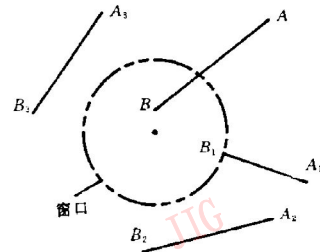


图 11 A 在窗口外  
Fig. 11 A out of window

若  $T_1 \geq 1$  或者  $T_1 < 0$  则不必进一步求交, 无图线输出, 过程结束, 如图 11 中的  $A_3B_3$  及  $A_1B_1, A_2B_2$  在窗口内无可见部分。

最后将可见线段进行类同“真、假区”算法中的“窗口—视图区变换”, 在输出设备的视图区内输出。

### 3 算法的应用实例及算法的质量分析

该算法可用各种高级语言设计成通用子程序, 供各种 CAD 调用。图 12 是在大量的计算机辅助设计中的一个应用实例, 即调用 FORTRAN 77 设计的“裁剪及覆盖”通用子程序, 在 PLOT 底层软件支撑下实现的。图 12a 是调用圆形窗口“参变量判别法”设计的“裁剪”图例; 图 12b 是调用圆形窗口“真、假”算法子程序设计的“覆盖”图例; 图 12c 是原图; 经过大量的实例考证, 证明此算法实用可靠、简单易行, 裁剪速度快, 既吸取了原有算法中的精华部分又具备了自己独创的算法风格。程序简明易读, 通用性好, 调用简单, 具有较大的实用价值和较高的推广价值。

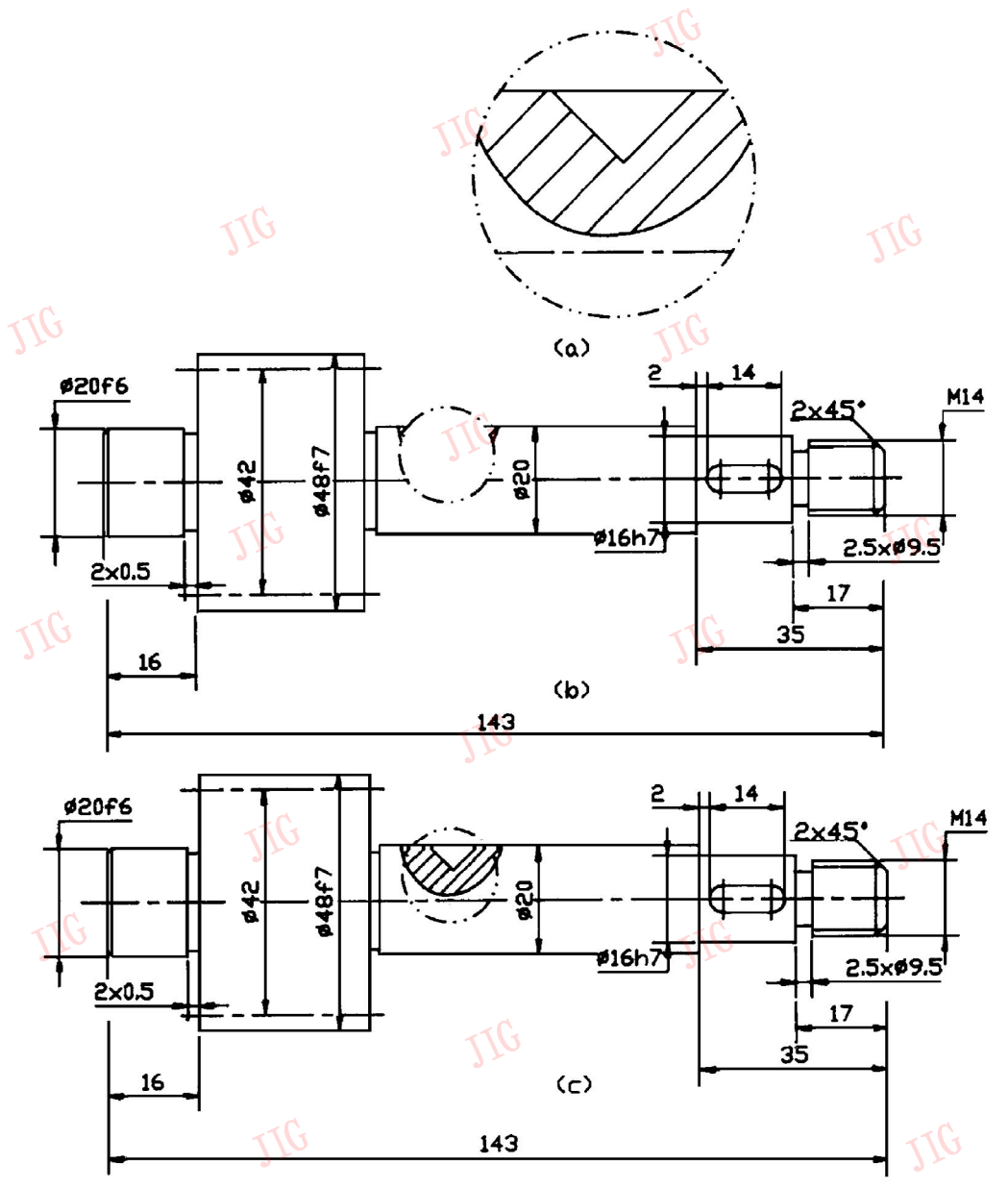


图 12 裁剪及覆盖的应用

Fig. 12 Clipping and covering of application

表 1 各种算法比较

Table 1 Algorithm Compare

窗口	分析对象 (图 13)	可靠性	绘制时间 (s)
① 矩形窗口	a		68
	b	可靠	49
	c	可靠	28
② 圆形窗口	d		68
	e	可靠	45
	f	可靠	25
③ 圆形窗口	d		68
	e	可靠	46
	f	可靠	26

注: 矩形窗口为 CohenSatherlandth 算法

圆形窗口一为“真、假”区算法

圆形窗口二为“参变量”判别法

图 13b、c 为将数学公式:  $\rho = A * \sin 2\theta$  用

FORTRAN 77 语言设计的主程序调用原有的矩形窗口“裁剪与覆盖”的子程序在 SP-450 绘图机上绘制的裁剪及覆盖图, 图 13 为原图; 图 13e、f 为调用本算法圆形窗口“裁剪与覆盖”的子程序在同一绘图机上绘制的裁剪与覆盖图, 图 13d 为原图。从大量 CAD 实践应用及图 13a、表 1 可看出在可靠性方面新算法与原算法均可靠, 但在速度上圆形窗口的“裁剪与覆盖”比矩形窗口快, 显而易见, 图形越复杂, 就越显示本算法的优越性, 因而圆形窗口“裁剪与覆盖”的新算法很有推广应用的价值。

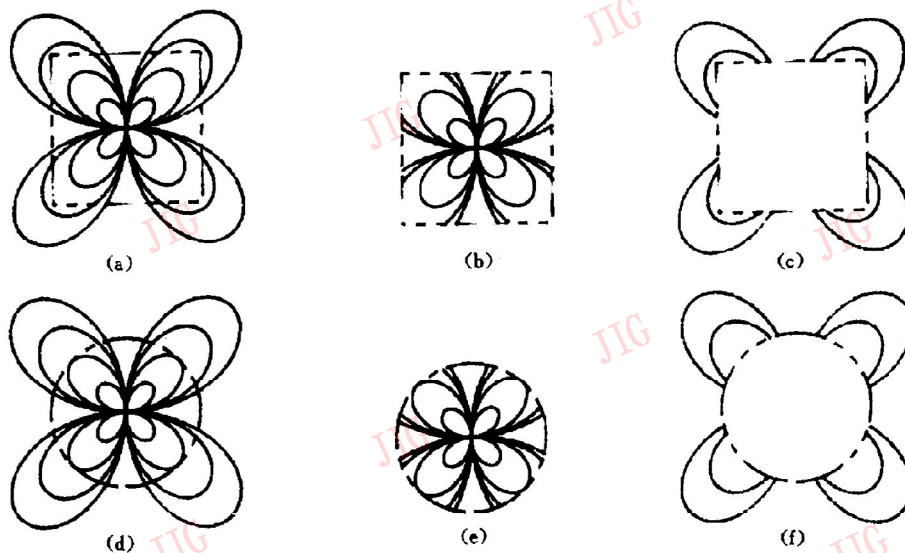


图 13 圆形窗口与矩形窗口算法比较

Fig. 13 Algorithm compare between circular and rectangular windows

## 参考文献

1 陆润民,杨惠英,施寅. 计算机绘图. 北京:清华大学出版社, 1988, 139~154.

2 何援军. 计算机图形学算法和实践. 湖南:科学技术出版社, 1990,145~152.

3 许隆文. 计算机绘图. 北京:机械工业出版社,1989, 412~434.



姚涵珍,副教授,天津轻工业学院机械系。主要研究方向为计算机图形学、计算机三维动画设计等。从事过天津市科委鉴定的塑料模具 CAD 科研多项,在全国刊物上发表过有关计算图象图形学的文章多篇。

## Computer Aide Designing of Graphics Clipping and Covering

Yao Hanzhen, Chu Daqing, Li Yanqi, Zhang Jian

(Tianjin Institute of Light Industry, Tianjin 300222)

**Abstract** Graphics clipping and covering is one of the key points in computer graphics field. After more study and practice, a new algorithm is give, which is for clipping and covering in circular window. Adopting shifting principle to do multiply and division operation, we have got a more speed in clipping and covering, raised the capability of computer in graphics processing.

**Keywords** Clipping and covering, Circular window, "True/false" field