改进的六角网格系统上矩形窗口裁剪算法*

付文超，张传林

(暨南大学 信息科学技术学院，广东 广州 510632)

摘 要：基于类直角坐标系的六角网格系统的特点，提出了一个用矩形窗口裁剪圆的算法，该算法中矩形窗口的左右边界可以用常数表达式表示，相比于传统60°角六角网格系统，算法复杂度大大降低。算法中对圆的裁剪操作分圆与矩形窗口包含、相离、相交三种情况，在包含和相离的情况中，用简单的比较运算代替复杂的求交运算，大大降低了算法的计算量。

关键词：六角网格系统；类直角坐标系；裁剪算法；算法模拟实现

中图分类号：TP391.41 **文献标识码**：A **文章编号**：1674-7720(2010)22-0044-05

An improved clipping algorithm in rectangle window based on hexagonal grids system

FU Wen Chao, ZHANG Chuan Lin

(College of Information Science and Technology, Jinan University, Guangzhou 510632, China)

Abstract: According to the characteristics of the Hexagonal Grids System which based on the Similar Rectangular Coordinate System, this paper presents an improved clipping algorithm for clipping circles in rectangle window. Compared with the Traditional Hexagonal Grids System, the given clipping algorithm of the new one is much less complex, because of the left and the right boundary of the rectangle window represented by a constant expression. The clipping circle operation can be sort into three categories: including, disjointing and intersecting. As we use the simple comparison operation to replace the complex intersection operation in the first two cases, the calculating amounts can be greatly reduced.

Key words: hexagonal grids system; similar rectangular coordinate system; clipping algorithm; algorithm implementation

裁剪是计算机图形学中的一个重要内容，它能对图形做出正确的判断，去除不可见部分，选取可见信息提供给显示系统进行显示。目前研究的裁剪算法很多，矩形窗口的图形裁剪算法更是使用最广泛的一类算法，但这些算法都是基于方形网格系统提出和实现的。而早在20世纪60年代初，数学家们就已经提出了平面上点的最佳分布是按六角网格分布的[1]，这种分布就是用正六边形来覆盖整个平面，每个像素对应着一个正六边形，将正六边形的中心点作为网格点。相比于方形网格系统，六角网格系统具有更好的图形图像显示特性[2-4]，因此基于六角网格显示系统的图形生成和处理算法的研究已成为了发展的必然。

本文给出了一种基于类直角坐标系的六角网格系统，并将其在方形网格显示设备上进行模拟显示，以作为六角网格系统上相关算法实现的模拟平台。

总体来说，六角网格上的裁剪操作要比方形网格上麻烦，参考文献[5]和参考文献[6]中分别给出了一种基于六角网格的矩形窗口的线段和圆裁剪算法，参考文献[7]给出了一种基于六角网格的圆形窗口的裁剪算法。这几种算法都是基于传统60°角六角网格系统提出的，算法处理比较复杂，导致六角网格系统优于方形网格系统的特性不再明显。本文基于类直角坐标系的六角网格系统提出了一种改进的矩形窗口图形裁剪算法，算法的复杂程度远远低于基于传统60°角六角网格系统的相关算法。

1 两种六角网格系统的比较

传统60°角六角网格系统与方形网格系统的关系：传统60°角六角网格系统上的任何一个点的坐标(x',y')，都可以通过变换：

* 基金项目：国家自然科学基金天元基金项目(10926141)，广东省科技计划项目(2009B01080030)

$$\begin{cases} x=x'+\frac{1}{2}y' \\ y=\frac{\sqrt{3}}{2}y' \end{cases}$$

转换成方形网格坐标(x,y)。

在方形网格显示设备上模拟显示传统60°角六角网格系统，如图1所示。

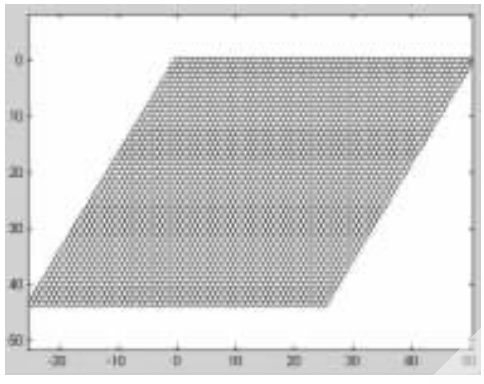


图1 传统60°角六角网格系统

基于类直角坐标系的六角网格系统与方形网格系统的关系：基于类直角坐标系的六角网格系统上的任何一个点的坐标(x',y')，都可以通过变换：

$$\begin{cases} x=x'+\frac{1}{2}r_0 \\ y=\frac{\sqrt{3}}{2}y' \end{cases}$$

转换成方形网格坐标(x,y)，其中$r_0=\mathrm{mod}(y',2)$。

在方形网格显示设备上模拟显示基于类直角坐标系的六角网格系统，如图2所示。

图2 基于类直角坐标系的六角网格系统

在传统60°角六角网格系统上设矩形窗口裁剪算法要比在方形网格系统上麻烦，这是因为在传统60°角六角网格系统上窗口的垂直边界需要用一条斜线来表示，而不是用常数表达式来表示。

在传统60°角六角网格系统上，设矩形窗口左下角点的坐标为(x_l,y_b)，右上角点的坐标为(x_r,y_t)，则矩形窗口的上边界和下边界表达式为：$y=y_t$和$y=y_b$。进一步推导出矩形窗口左上角点的坐标为$(x_l-\frac{y_t-y_b}{2},y_t)$，右下角点的坐标为$(x_r+\frac{y_t-y_b}{2},y_b)$，从而得到矩形窗口的左边界和右边界的表达式，分别为：$y+2(x-x_l)-y_b=0$和$y+2(x-x_r)-y_t=0$。这样，矩形窗口左右边界的表达式就相当复杂，大大增加了算法的计算量。

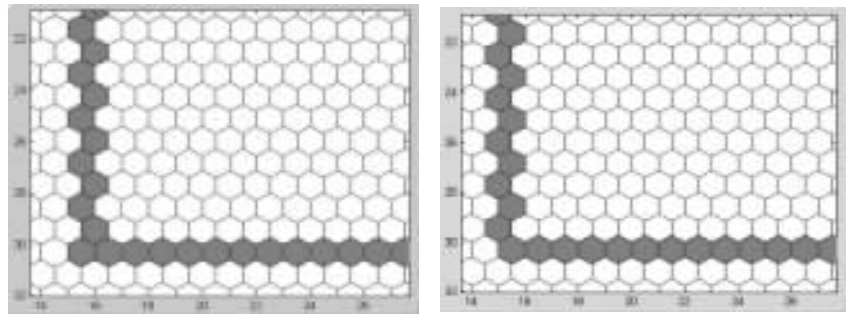
在基于类直角坐标系的六角网格系统上，设矩形窗口左下角点的坐标为(x_l,y_b)，右上角点的坐标为(x_r,y_t)，则矩形窗口的上边界和下边界表达式为$y=y_t$和$y=y_b$。至于矩形窗口的左边界和右边界，可基于类直角坐标系的六角网格的排布特点（如图3所示）来确定。

图3 基于类直角坐标系系的六角网格的排布方式

根据左下角的点(x_l,y_b)，可以知道矩形窗口左边界的方程为$x=x_l$，是一个常数表达式。但是，如果纵坐标y_b为偶数，则左边界$x=x_l$直线上六角网格的排布就如图4(a)所示；如果纵坐标y_b为奇数，则左边界$x=x_l$直线上六角网格的排布就如图4(b)所示。

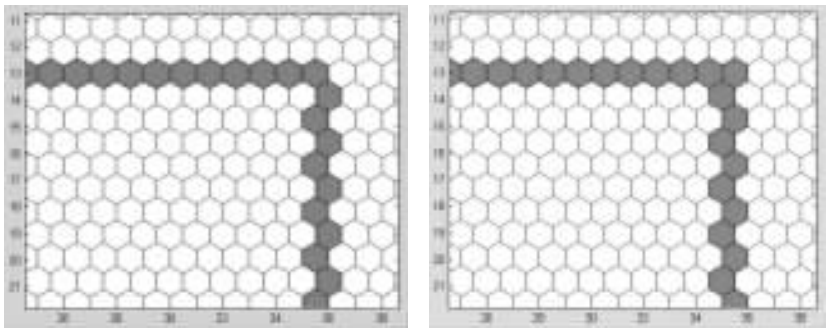
同样地，根据右上角的点(x_r,y_t)，矩形窗口的右边界的方程为$x=x_r$，也是一个常数表达式。如果纵坐标y_t为偶数，则右边界$x=x_r$直线上六角网格的排布就如图5(a)所示；如果纵坐标y_t为奇数，则右边界$x=x_r$直线上六角网格的排布就如图5(b)所示。

综上所述，可以在模拟实现的基于类直角坐标系的六角网格系统平台上生成一个矩形窗口，如图6所示。



(a) y_b为偶数时 (b) y_b为奇数时

图4 左边界直线上六角网格的排布



(a) y_b 为偶数时　　(b) y_b 为奇数时

图 5 右边界直线上六角网格的排布

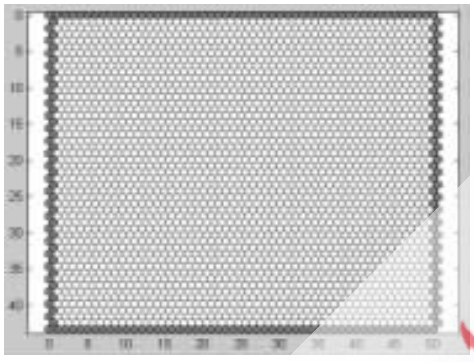


图 6 六角网格系统上模拟实现的矩形窗口

2 基于类直角坐标系的六角网格上的矩形窗口裁剪算法

前面已经在六角网格系统模拟平台生成了矩形窗口，下面讨论基于类直角坐标系的六角网格上的矩形窗口裁剪算法，并给出一个在矩形窗口中裁剪圆的操作实例。

圆在基于类直角坐标系的六角网格系统中的方程为：

$$\left(x+\frac{1}{2}\times\mathrm{mod}(y,2)-x_0-\frac{1}{2}\times\mathrm{mod}(y_0,2)\right)^2+\left(\frac{\sqrt{3}}{2}y-\frac{\sqrt{3}}{2}y_0\right)^2=R^2$$

其中，(x_0,y_0) 为圆心坐标，(x,y) 和 (x_0,y_0) 均为六角网格系统中的坐标，R 为圆的半径。

圆与矩形窗口的位置关系，有包含、相离、相交三种。对于包含和相离的情况，裁剪操作很容易处理。对于相交的情况，必须求出交点坐标，并且把交点分为入点和出点两类：入点即圆在该点处进入矩形窗口，出点即圆在该点处离开矩形窗口。由于圆是封闭曲线，因此如果圆在某一个点处进入矩形窗口，那么在它再次进入矩形窗口之前，中间必须有一个第三点使得圆先离开矩形窗口，即：在矩形窗口的边界上，不会有相邻的两个入点；同理，在矩形窗口的边界上，也不会有相邻的两个出点。

2.1 算法描述

在基于类直角坐标系的六角网格系统上使用矩形窗口裁剪圆曲线的流程是：先判断圆与矩形窗口是否包含或者相离，如果是，则裁剪算法结束；否则，求出圆与矩形窗口边界的交点，并进行裁剪操作。

由圆在基于类直角坐标系的六角网格系统中的方程可以计算出：

$$x=x_0\pm\sqrt{R^2-\left(\frac{\sqrt{3}}{2}y-\frac{\sqrt{3}}{2}y_0\right)^2}-\frac{1}{2}\times\mathrm{mod}(y,2)+\frac{1}{2}\times\mathrm{mod}(y_0,2)$$

$$y=y_0\pm\sqrt{\frac{4}{3}\times\left(R^2-\left(x+\frac{1}{2}\times\mathrm{mod}(y,2)-x_0-\frac{1}{2}\times\mathrm{mod}(y_0,2)\right)^2\right)}$$

其中，x 坐标的最大值为 x_0+R，最小值为 x_0-R；y 坐标的最大值为 $y_0+\frac{2\sqrt{3}}{3}\times R$，最小值为 $y_0-\frac{2\sqrt{3}}{3}\times R$。

2.1.1 整个圆在矩形窗口内部(包含关系)

如果 x 坐标的最大值、最小值和 y 坐标的最大值、最小值满足：

$$\begin{cases}(x_0-R\geqslant x_l)\&(x_0+R\leqslant x_r)\\ \text{and}\\ \left(y_0-\frac{2\sqrt{3}}{3}\times R\geqslant y_b\right)\&\left(y_0+\frac{2\sqrt{3}}{3}\times R\leqslant y_t\right)\end{cases}$$

则整个圆在矩形窗口内部，如图 7 所示，这种情况下不需要裁剪，整个圆在矩形窗口中都得到显示。

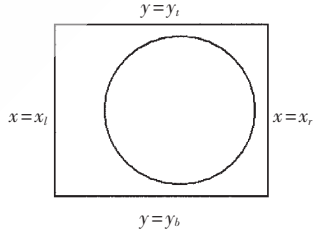


图 7 矩形窗口包含整个圆

2.1.2 整个圆在矩形窗口外部(相离关系)

如果 x 坐标的最大值、最小值和 y 坐标的最大值、最小值满足：

$$\begin{cases}(x_0+R<x_l)\text{or}(x_0-R>x_r)\\ \text{or}\\ \left(y_0+\frac{2\sqrt{3}}{3}\times R<y_b\right)\text{or}\left(y_0-\frac{2\sqrt{3}}{3}\times R>y_t\right)\\ \text{or}\\ \left(x_0+\frac{\sqrt{2}}{2}\times R<x_l\,\&\,y_0-\frac{\sqrt{6}}{3}\times R<y_b\right)\text{or}\left(x_0-\frac{\sqrt{2}}{2}\times R>x_r\,\&\,y_0+\frac{\sqrt{6}}{3}\times R<y_b\right)\\ \text{or}\\ \left(x_0-\frac{\sqrt{2}}{2}\times R>x_r\,\&\,y_0-\frac{\sqrt{6}}{3}\times R>y_t\right)\text{or}\left(x_0+\frac{\sqrt{2}}{2}\times R<x_l\,\&\,y_0-\frac{\sqrt{6}}{3}\times R>y_t\right)\end{cases}$$

则整个圆在矩形窗口外部，如图 8 所示，这种情况下不需要裁剪，整个圆在矩形窗口中都没有显示。

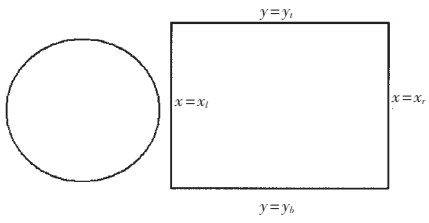


图 8 矩形窗口与圆相离

2.1.3 圆部分在矩形窗口内部，部分在矩形窗口外部（相交关系）

圆与矩形窗口相交是裁剪算法中最复杂的部分，这种情况下需要求出圆与矩形窗口的交点，并将窗口外的部分裁剪掉，显示窗口内的部分。

将矩形边界方程 $x=x_l$、$x=x_r$、$y=y_b$ 和 $y=y_t$ 与圆的方程联立，求出所有交点坐标，写入交点集合 I。如果某个边界与圆相切，只有一个交点，则该点不写入交点集合。

(1)矩形窗口与圆有 8 个交点，如图 9 所示，记最下方交点中靠左的交点为 I_1，然后按照逆时针顺序依次记其他交点为 $I_2,I_3,\cdots,I_8$。

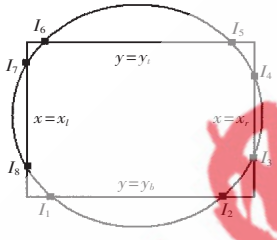


图 9 矩形窗口与圆有 8 个交点的情形

判断 I_1 是出点还是入点，方法是：如果 I_1 在矩形窗口的下边界上，则将 $I_1=(x,y_b)$ 的横坐标增加 1 并代入圆的方程，可求出相应的纵坐标 y' 有两个，取圆弧上 I_1 沿逆时针方向的下一个点 $I_1'=(x+1,y')$，如果 $y'>y_b$，则 I_1 为入点，否则 I_1 为出点；如果 I_1 在矩形窗口的右边界上，则将 $I_1=(x_r,y)$ 的纵坐标增加 1 并代入圆的方程，求出圆弧上 I_1 沿逆时针方向的下一个点 $I_1'=(x',y+1)$，如果 $x'<x_r$，则 I_1 为入点，否则 I_1 为出点；如果 I_1 在矩形窗口的上边界上，则将 $I_1=(x,y_t)$ 的横坐标减小 1 并代入圆的方程，求出圆弧上 I_1 沿逆时针方向的下一个点 $I_1'=(x-1,y')$，如果 $y'<y_t$，则 I_1 为入点，否则 I_1 为出点；如果 I_1 在矩形窗口的左边界上，则将 $I_1=(x_l,y)$ 的纵坐标减小 1 并代入圆的方程，求出圆弧上 I_1 沿逆时针方向的下一个点 $I_1'=(x',y-1)$，如果 $x'>x_l$，则 I_1 为入点，否则 I_1 为出点。

如果 I_1 为出点，则圆弧 I_2I_3、I_4I_5、I_6I_7、I_8I_1 位于窗口内部，将其余部分裁剪掉；如果 I_1 为入点，则圆弧 I_1I_2、I_3I_4、I_5I_6、I_7I_8 位于窗口内部，将其余部分裁剪掉。

(2)矩形窗口与圆的交点少于 8 个，如图 10 所示，仍然记最下方靠左的交点为 I_1，如果不存在这样的 I_1，则按照逆时针顺序，将遇到的第一个交点记为 I_1，其余交点顺次标记。然后按照(1)中的步骤进行裁剪操作。

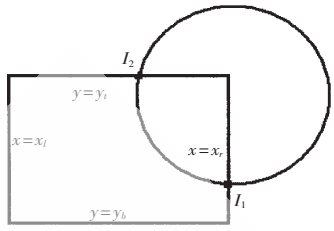


图 10 矩形窗口与圆交点少于 8 个的情形

2.2 算法模拟实现

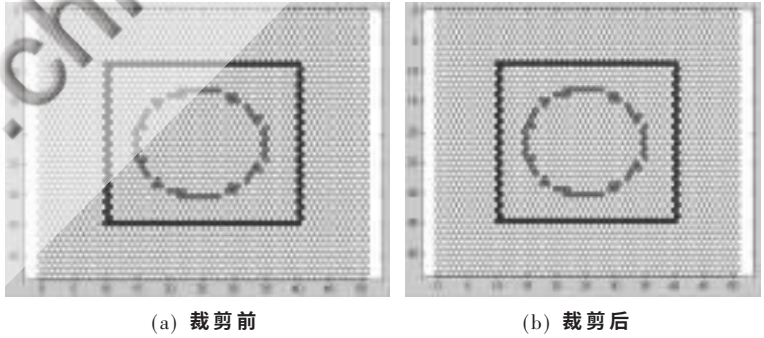
对本文提出的算法进行模拟，结果如图 11～图 14 所示。

2.3 算法分析

由于六角网格系统的特殊性，一些在方形网格系统中比较简单的公式在六角网格系统下就变得比较复杂，本文给出的基于类直角坐标系的六角网格系统的相关算法对该问题做了简化，具有以下优点：

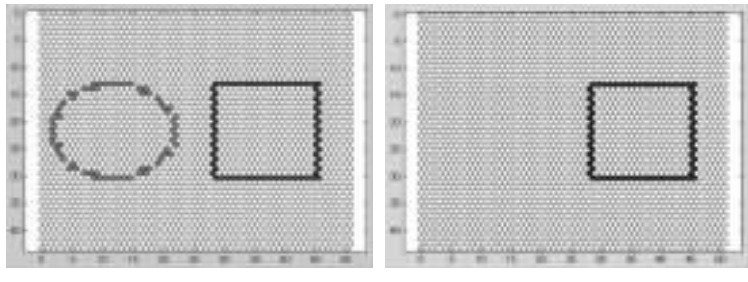
(1)矩形窗口的左右边界也是一条简单的垂直线，可以用常数表达式来表示，这样就可以大大降低算法的计算量；

(2)算法处理速度快。本算法分三种情况来讨论圆与矩形窗口的关系，前期先判断圆与矩形窗口是否为包含或者相离的关系，通过简单的比较运算来替代复杂的求交运算，减少了计算量；



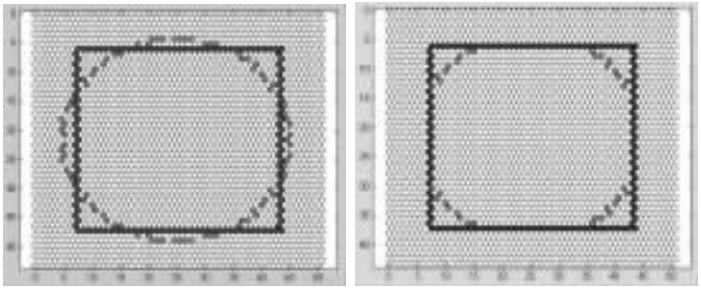
(a) 裁剪前 (b) 裁剪后

图 11 整个圆在矩形窗口内部的情形



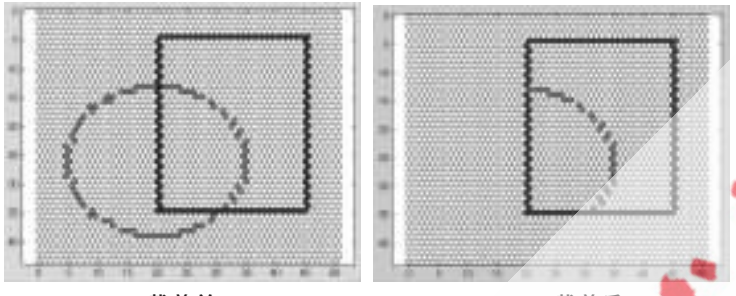
(a) 裁剪前 (b) 裁剪后

图 12 整个圆在矩形窗口外部的情形



(a) 裁剪前 (b) 裁剪后

图 13 圆与矩形窗口有 8 个交点的情形



(a) 裁剪前 (b) 裁剪后

图 14 圆与矩形窗口有少于 8 个交点的情形

(3)简化了入点和出点的判断。本算法利用封闭图形的入点和出点在矩形窗口边界上交互出现，只需判断一个起始点为入点或出点，然后按照逆时针顺序，根据入点和出点交互分布的原则依次判断其他点是入点或出点。

在基于类直角坐标系的六角网格系统上，矩形窗口的边界可以用常数表达式来表示，这大大简化了六角网格系统上图形裁剪算法的研究。由于这类六角网格系统上矩形窗口边界表达式与方形网格系统上矩形窗口表达式形式一致，可以将方形网格系统上的其他图形图像处理算法直接或间接地移植到该类六角网格系统上。此外，本文给出的算法具有一定通用性，只需将图形方程进行变化，即可适用于其他图形图像的矩形窗口裁剪。

参考文献

[1] ROGERS C A. Packing covering[M]. Cambridge: Cambridge University Press, 1964.

[2] TIRUNELVELI G, GORDON R, PISTORIUS S. Comparison of square-pixel and hexagonal-pixel resolution in image processing[J]. Proceedings of the 2002 IEEE Canadian Conference On Electrical & Computer Engineering.

[3] CONDAT L, Dimitri Van De Ville, BLU T. Hexagonal versus orthogonal lattices: a new comparison using approx-imation theory[J]. ICIP: IEEE International Conference on Image Processing, 2005, 3: 11-14.

[4] 刘勇奎，邹善举. 六角网格上的图像算法及几何量定义[J]. 计算机工程与设计，2000，21(1)：61-64.

[5] 赵慧杰，晏俊德，刘勇奎，等. 六角网格单色显示器及图形算法研究[J]. 沈阳工业大学学报，1997，19(5)：66-71.

[6] 韩丽. 基于六角网格的矩形窗口的圆裁剪算法[J]. 锦州师范学院学报(自然科学版)，2003，24(4)：64-66.

[7] 孙长嵩，李丽洁，宋阳. 一个基于六角网格的圆形窗口的裁剪算法[J]. 交通科技与经济，2006，33(1)：78-80.

[8] 付文超，罗小华，张文争，等. 改进的六角网格下椭圆绘制算法[J]. 暨南大学学报(自然科学与医学版). 已录用，待刊登.

(收稿日期：2010-08-26)

作者简介：

付文超，男，1986 年生，硕士研究生，主要研究方向：计算几何。

张传林，男，1964 年生，硕士生导师，博士，教授，主要研究方向：数值计算及其应用、计算几何、算法设计与复杂性分析、无线网络传感器的连通覆盖等。