基于直线特性的直线生成集成算法

程 锦 陆国栋 谭建荣

(浙江大学 CAD&CG 国家重点实验室,杭州 310027)

摘 要 在分析和比较了现有的直线生成算法后,以 Bresenham 算法为基础,充分利用直线的对称性、方向性和连续性,设计了一个基于直线特性的直线生成集成算法,以进一步提高直线生成速度.实验表明,该集成算法与 Bresenham 算法相比,直线绘制速度提高了 50%以上.特别是在工程图样中,由于绝大部分直线为水平线、垂直线、±45°方向直线,因此本算法将更为有效.

关键词 Bresenham 算法 对称性 方向性 连续性 集成算法 工程图样

中图法分类号:TP391.4 **文献标识码**:A **文章编号**:1006-8961(2001)04-0392-04

A Property-Based Integrated Line-Generating Algorithm

CHENG Jin, LU Guo-dong, TAN Jian-rong

(*State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou* 310027)

Abstract Present line-generating algorithms, which include Bresenham algorithm, symmetrical algorithm and the algorithm based on the chain code theory of the line, are analyzed and compared in this paper. A property-based on the chain code theory of the line, are analyzed and compared in this paper. A property-based integrated line-generating algorithm, which is based on the Bresenham algorithm and makes full use of the symmetry, direction and continuity of the line, is presented to increase the speed of line generating. The results of our experiments have proved that the line-generating speed of the integrated algorithm is increased by more than 50 percent compared to that of Bresenham algorithm. The new algorithm is especially efficient for engineering drawings, in which the inclinations of most lines are 0°, 180°, ±90°, ±45 °and ±135°. In can be thought that the integrated line-generating algorithm has make great progress in the way to reach the minimum value of the operation time in generating a line. Existing CAD&CG algorithms must take into account the particularity and entireness of the object while the integrity and universality of the processing object is pursued. To handle the related algorithms in some special way according to the characteristics of engineering drawings is an effective method to improve present CAD&CG algorithms.

Keywords Bresenham algorithm, Symmetry, Direction, Continuity, Integrated algorithm, Engineering drawings

0 引 言

大家知道,直线生成算法是计算机图形学及 CAD 最基本的算法之一,因为一般来说,每一个实际的图形都要包含成百上千个直线段,而圆弧和复杂曲线也往往可由直线段逼近来表示.另外,由于对图形的许多操作,如图形裁剪、图形变换等,也都是以直线的生成为基础的,因此,直线的生成速度极大地影响着图形系统的效率.

目前,代表性的直线生成算法有 DDA 法、正负法与 Bresenham 算法[1~3]等,其中 Bresenham 算法被公认为是最有效的直线绘制算法,其实际应用也最为广泛.它的基本思想是:每走一步,一个坐标改变 1 象素值,而另一个坐标则根据决策参数 e 的符号来决定是否改变.这种算法的优点是:通过引入决策参数 e,即可除了在对变量进行初始化时要进行除法操作外,以后每走一步只需进行一次整数加减法运算和一次判断操作.

文献[4]和[5]对 Bresenham 算法进行了改进,先后提出了一个对称的快速直线生成算法[4]和一个基

基金项目:863/CIMS 主题资助项目(863-511-9842-006)
收稿日期:2000-03-02 改回日期:2000-06-01

于直线链码理论的快速直线生成算法[5]，这两种算法每循环一次即有可能绘制出两个点，从而提高了直线生成速度.

本文通过对直线进行整体特性的分析，将直线的对称性、方向性和连续性三者有机地结合起来，尽可能地避免了直线生成中的判断操作，并且在一次循环中有可能生成 4 个点，从而进一步提高了直线生成速度.

1　直线生成集成算法

1.1　现有算法分析

大家知道，Bresenham 算法每循环一次生成一个点，而文献[2]的对称直线生成算法则利用直线段关于中点的对称性，使得每次循环可生成相对于中点的两个对称点，这一特性适用于任意方向的直线，这也是对 Bresenham 算法较有意义的改进. 文献[3]根据直线的连续性，并基于链码理论来进行直线绘制，虽然其一次循环有可能同时生成两个点，但该算法每循环一次，其生成两个点的概率却需取决于具体直线的斜率. 例如，对于图 1 中的位于 0°到 45°范围的直线，若直线的斜率接近于 1/2，则几乎每次循环均可生成两个点；若直线的斜率接近于 0，则一次循环生成两个点的概率几乎为零. 显然，出现直线斜率接近 1/2 的机会远比出现 8 个特殊方向的机会小，因而实际上，用这种方法来绘制 8 个特殊方向的直线时，由于只出现一个基码，而没有另一个单独出现的基码，因此生成直线所需的循环次数与 Bresenham 算法完全相同.

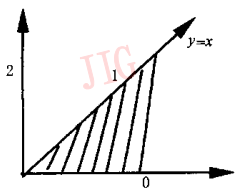

图 1

1.2　集成算法描述

现有的各种直线生成算法，很少考虑直线的方向性，即对水平方向、垂直方向、45°方向、－45°方向这 8 种特殊方向的直线均采用与一般位置直线相同的绘制方法. Bresenham 算法及文献[2]、[3]所描述的两种改进算法，在绘制 8 个特殊方向的直线时，每循环一次，都要进行一次更新决策参数 e 所需的整数加减法运算和判断 e 符号所需的判断操作.

实际上，绘制 8 个方向的特殊直线时，只需判断一次直线方向，即可循环多次，直至生成整条直线，而不必根据决策参数 e 的符号来判断下一次循环的走向，从而可以省去每次循环中更新 e 所需的整数加减运算和判断 e 符号所需的判断操作，并能使得生成这 8 种特殊方向直线的速度大大提高.

以直线位于图 1 所示区域的情况为例进行说明，其他区域的直线可以由区域对称性获得. 在初始化变量后，先判断所要绘制的直线方向是否为特殊方向，并作相应的处理. 若直线方向为特殊方向，则利用特殊直线的方向性，一次性生成整条直线，即，若 $dy = 0$(直线为 0 方向)，则进行 dx 次循环，每循环一次使 x 增 1，y 不变；若 $dx = dy$(直线为 1 方向)，则进行 dx 次循环，每循环一次使 x 和 y 都增 1，即可生成整条直线(dx、dy 分别表示直线两端点的水平和垂直差值). 若直线不属于特殊方向，则采用将文献[2]、[3]两种方法集成的算法进行绘制，这样，由于在绘制一般位置直线时，直线的生成是由两端向中间进行的，且直线一端通过单独出现的方向走到下一点时，下一个点的方向也是已知的(必定是另一个方向)，因此，集成算法一次循环最多可能生成 4 个点.

以下为集成算法具体描述：

```
算法开始：
x1 = xstart ;y1 = ystart ;x2 = xend ;y2 = yend ;flag = 0 ;
求 dx,dy 及每走一步的增量 sx,sy ;
if dy > dx then
   {交换 dx 与 dy,flag = 1 ;}
if dy = = 0 then // 绘制水平直线或垂直线
   {for i = 1 to dx
      {plot(x1 ,y1) ;根据 flag 值更新 x1 或 y1 ;}
  }
if dy = = dx then //绘制斜率为 ± 1 的直线
   {for i = 1 to dx
      {plot(x1 ,y1) ;更新 x1 和 y1 ;}
  }
dx2 = 2dx ;dy2 = 2dy ;e = dy2 - dx ;
dyx1 = dy2 - dx2 ;dyx2 = dyx1 + dy2 ;
if dx > dy2 then
 {for i = 1 to (dx - dy + 1)/2
    {根据 flag 更新 x1 ,x2 或 y1 ,y2 ;
     if e < 0 then e = e + dy2 ;
```

```
            else {根据 flag 更新 y1,y2 或 x1,x2;
                  plot(x1,y1);plot(x2,y2);
                  根据 flag 更新 x1,x2 或 y1,y2;
                  e = e + dyx2;
                 }
            plot(x1,y1);plot(x2,y2);
      }
}
else
      {for i = 1 to (dy + 1)/2
        {根据 flag 更新 x1,x2 或 y1,y2;
         if e < 0 then
                  {plot(x1,y1);plot(x2,y2);
                   根据 flag 更新 x1,x2 或 y1,y2;
                   e = e + dyx2;
                   }
                  else e = e + dyx1;
         根据 flag 更新 y1,y2 或 x1,x2;
         plot(x1,y1);plot(x2,y2);
        }
      }
if 中点没有生成 then
   {在直线一端再前进一步,生成中点;}
算法结束
```

1.3 运算量的比较

现以直线方向为方向 0 或方向 1 的两种特殊情况为例来进行讨论,其他 6 种特殊方向的直线可由区域对称关系得到相同的结果. 表 1 和表 2 分别列出了采用 Bresenham 算法、对称算法、基于链码理论的算法及集成算法绘制 0 方向和 1 方向直线时所需的运算量.

表 1 几种不同算法运算量比较
(直线为 0 方向,$dx > 0, dy = 0$)

算法名称	Bresenham 算法	对称算法	基于链码理论的算法	集成算法
更新 e 所用加减运算	dx	$(dx+1)/2$;$dx+1$ 为偶数 $(dx+3)/2$;$dx+1$ 为奇数	dx	0
对 e 符号判断操作	dx	$(dx+1)/2$;$dx+1$ 为偶数 $(dx+3)/2$;$dx+1$ 为奇数	dx	0
x 坐标运算总次数	dx	$dx+1$	dx	dx
y 坐标运算总次数	0	0	0	0

表 2 几种不同算法运算量比较
(直线为 1 方向,$dx = dy > 0$)

算法名称	Bresenham 算法	对称算法	基于链码理论的算法	集成算法
更新 e 所用加减运算	dx	$(dx+1)/2$;$dx+1$ 为偶数 $(dx+3)/2$;$dx+1$ 为奇数	dx	0
对 e 符号判断操作	dx	$(dx+1)/2$;$dx+1$ 为偶数 $(dx+3)/2$;$dx+1$ 为奇数	dx	0
x 坐标运算总次数	dx	$dx+1$	dx	dx
y 坐标运算总次数	dy	$dy+1$	dy	dy

由表 1 和表 2 可以看出,用集成算法绘制 8 种特殊方向的直线,可以使更新 e 所需的加减运算次数和判断符号 e 所需的判断操作次数为零. 至于使 x 坐标和 y 坐标变化所需的加减法运算,则对每种算法都是相同的.

另外,由于一般位置直线必然会出现两个相邻的基元方向,且其中必定有一个方向是单独出现的,因此,集成算法在绘制一般位置直线过程中,一定会出现一次循环生成 4 个点的情况(一般情况下每一次循环可生成两个点). 可见,用集成算法绘制一般位置直线可获得比对称算法更快的速度.

综上所述可见,集成算法不仅能提高绘制任意方向(尤其是特殊方向)直线的速度,而且,整幅图形中特殊方向直线所占比例越大、所需绘制的直线越长、生成直线所需的循环次数越多,其节省的运算量就越大,也越能体现算法的优越性. 由于工程图样的图线绝大部分为特殊方向直线,据统计,其比例超过 90%,因此,将集成算法应用在工程图样中必将极大地加快直线绘制的速度.

2 实验结果分析与比较

表 3 给出了分别用 Bresenham 算法、对称算法、基于链码理论算法及集成算法绘制5 000根随机直线所用的时间(其中,水平线、垂直线、45°方向、-45°方向直线及一般位置直线各1 000根,其中特殊方向直

表 3 几种不同算法绘制时间比较

算法名称	Bresenham 算法	对称算法	基于链码理论算法	集成算法
画线所用时间(ms)	490	330	440	244
与 Bresenham 算法相比效率提高(%)		32.65	10.20	50.20

(CPU:PENTIUM-S 133 MHz)

线占 80%).由于各种算法绘制相同直线时所生成的点数相同,所以表 3 中记录的时间未包括画点时间,以便能更清楚地比较各种算法画线相对速度.

由表 3 可以看出,与 Bresenham 算法、对称算法、基于链码理论算法相比,集成算法明显提高了直线的生成速度.

3　结　论

由于现有的图形学及 CAD 算法在追求处理对象的完备性与普适性的同时,必须充分考虑处理对象的特殊性与整体性,因此本文即在完备普适的 Bresenham 算法基础上,融直线的对称性、方向性与连续性为一体,提出了直线生成的集成算法.实验结果也说明,集成算法离直线生成计算量最小的极限更近了,而且直线生成集成算法作为 CAD 和 CG 领域的基本算法若能实施商品化的固化,将有助于 CAD 系统图形生成速度的进一步提高.

鉴于工程图样中的直线绝大部分为水平线、垂直线和斜率为 ± 1 的直线,因此本文提出的集成算法将极大地加速工程图样中直线段的绘制.由此可见,针对工程图样的特点进行相关算法的特殊处理,是面向工程图样对现有图形学与 CAD 算法进行改进的有效途径.

参 考 文 献

1　Bresenham J E. Algorithms for computer control of a digital plotter. IBM systems Journal,1965,4(1):2530.

2　唐荣锡,汪嘉业,彭群生等编著. 计算机图形学教程. 北京:科学出版社,1990.

3　金廷赞著. 计算机图形学. 杭州:浙江大学出版社,1988.

4　刘勇奎. 一个对称的快速直线生成算法. 微计算机应用,1993,14(2):4251.

5　刘勇奎. 一个基于直线链码理论的快速直线绘制算法,微计算机应用,1994,15(6):2931.

程　锦　1978 年生,博士研究生.主要研究方向为计算机辅助设计与计算机图形学.

陆国栋　1963 年生,博士,教授,浙江大学工程及计算机图学研究所副所长.主要研究领域为智能 CAD、三维重建、工程图样计算机理解等.

谭建荣　1954 年生,教授,博士生导师,浙江大学机械工程与自动化系主任,中国图象图形学报编委.主要研究领域为产品信息建模、计算机辅助设计、计算机图形学等.