

基于循环减法加速的多像素 Bresenham 直线绘制算法^①

李 向

(郑州航空工业管理学院 计算机科学与技术系, 郑州 450015)

摘 要: 直线生成算法是图形的基本元素, 也是计算机图形学基础理论的重要内容。Bresenham 算法是直线生成算法中最经典的算法, 但是它一次计算只能生成一个像素点, 效率低下。为了提高直线生成效率, 充分利用待生成线段的起点和终点坐标及线段自身的对称性, 对 Bresenham 算法进行改进, 一次计算生成一个像素行。讨论硬件实现该算法的加速方法。

关键词: 直线; Bresenham; 计算机图形学; 多像素; 算法

Accelerated Multi-Pixel Bresenham Line Drawing Algorithm Based on Circular Subtraction

LI Xiang

(Zhengzhou Institute of Aeronautical Industry Management, Zhengzhou 450015, China)

Abstract: Line drawing algorithm is very important in the computer graphics. It is the most famous algorithm that Bresenham Algorithm uses to draw a straightness line in a smooth fashion. The advantage of classic Bresenham Algorithm is that the operations are all integer numeric without division and decimal fraction. Meanwhile, the disadvantage of the algorithm is that it can just generate one pixel in a computation time and its low in efficiency. In this paper, an accelerated algorithm was proposed, which give full attention to the initiative relation between generation model of line and its linear slope, and it does generate more than one pixel of a line through row of raster display device. The core principle of the improved algorithm is utilizing the coordinate position of both ends of line and the symmetry characteristic for segments. After discussing theory and structure of the improved algorithm, the algorithm implementation and simulation are given at last. And finally, the hardware acceleration in the use of circular subtraction technology based on shift register was briefly described.

Keywords: line; bresenham; computer graphics; multi-pixel; algorithm

直线是图形中最基本的元素之一, 直线生成算法是其他各类图形算法的基础, 直线绘制算法的好坏对图形应用系统的效率和质量有直接而又重要的关系。

1 引言

早在图形学诞生伊始, 人们对直线生成算法进行了广泛的研究, 最著名的直线生成算法是 20 世纪 60 年代中期出现的 Bresenham 算法^[1]。Bresenham 算法的优点在于所有运算都是整数运算, 不需要进行小数和取整运算, 绘制一个点的运算量是 1 次整数加法运算

和 1 次符号判断。其缺点在于效率不高, 一次计算只能生成一个像素点。

为了提高 Bresenham 算法的绘制速度, 许多研究, 如参考文献中的[2-7]都对该算法进行了改进, 研究工作主要集中在一次生成位于直线上的尽可能多个像素点, 参考文献[2-4]提出二步法, 一次判断可以生成两个像素点, 该算法对斜率接近于 1/2 的直线效果较好, 对误差的一次判断可以生成两个像素点, 但对趋于相对平坦的直线效果不够显著。文献[5-7]利用直线起点和终点的坐标计算直线斜率, 进而计算各像素行的像

① 基金项目:河南省教育厅教育科学研究“十一五”规划([2009]-JKGHAG-0730);郑州航院教育科学研究课题(2010-58-38)

收稿时间:2010-07-29;收到修改稿时间:2010-09-05

素点数目，一次计算可以计算一行的像素点，但是需要进行除法和取整运算，引入了浮点运算，增加了运算量。文献[8-10]中提出的多步法都是以增加代码复杂度和存储容量来提高速度，一次判断可以生成多个像素点，文献[11]则根据直线的对称性，同时从直线两端生成两个像素点。

本文在深入研究 Bresenham 算法和其他改进算法的基础上，充分理解经典 Bresenham 算法精髓，结合其他改进算法中合理的因素，摒弃各种算法的缺陷，提出了一种改进算法。该改进算法继承了 Bresenham 算法只需要使用整数加法和乘法的优点，通过一次判断，生成一个像素行，算法实现简单、运算量少，算法效率有了明显的提高。

2 改进的不含除法的Bresenham算法

本文只对斜率在[0, 1]之间的直线进行讨论，斜率在[1, ∞]之间的直线可通过互换 x 轴和 y 轴的坐标得到。

传统的 Bresenham 算法的原理如图 1 所示。由于像素坐标的整数性，数学点 (x_i, y_i) 与所取显示设备像素点 (x_i, y_i) 间会引起误差 (ϵ_i) ，当 x_i 列上已用像素坐标 (x_i, y_i) 表示直线上的点 (x_i, y_i) ，下一行直线上点 B (x_{i+1}, y_{i+1}) ，是取像素点 C (x_{i+1}, y_i) ，还是 D (x_{i+1}, y_{i+1}) 遵循这样的规则：设 A 为 CD 边的中点：若 B 点在 A 点上方，选择 D 点；否则，选 C 点。

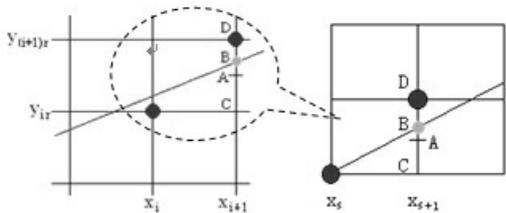


图 1 Bresenham 算法的原理

为了便于描述改进的算法，引入如下定义：

定义 1. 直线在一行中的像素点的个数定义为步长。

传统 Bresenham 算法及两步法等改进算法把注意力都集中在算法本身，忽略了直线的特性。如知道了

直线的起点坐标就可以知道直线的斜率，进一步可得到直线在每一行中的步长的范围(该直线除起始和终止两像素行外)。例如：起点坐标为(0, 0)，终点坐标为(20, 4)的线段，其斜率为 $k=0.2$ 。可知该线段的步长 $Q=1/k=5$ ，无须对偏差项进行计算和判断，根据每行步长为 4 可直接画出该线段。

对于更为一般的情况，例如起点坐标为(0, 0)，终点坐标为(100, 18)的线段，其斜率为 $k=0.18$ ，步长 $Q=1/k=5.56$ (保留 2 位小数)，则该直线除了起始和终止两行的步长外，其他各行的步长个数为 5 或 6。

定理 1. 用 Bresenham 算法生成的直线除了在第一行和最后一行外其他各行的步长 Q 满足：

$$E \leq Q \leq E + 1$$

设直线的起点坐标为 (x_a, y_a) ，终点坐标为 (x_b, y_b) ， $dx = x_b - x_a$ ， $dy = y_b - y_a$ 。在 Bresenham 算法中，直线像素点的判定变量为 $d = F(x_p + 1, y_p + 1/2)$ ，式中 (x_p, y_p) 是像素点的坐标， $F(x, y)$ 为直线方程。用各个像素点的判定量来确定下一个像素点的位置。若 $d < 0$ ，在本像素行的下一个像素点上生成一个像素点，若 $d > 0$ 则在下一个像素行生成一个像素点。

根据定理 1，我们只需确定在本像素行第一个像素点之后，第 n 个 ($n=E$) 像素点对应的判定变量的符号就可以知道该行的步长。这时该行的判定向量即可定义为 $d_n = d_1 + 2ndy - 2dx$ ，其中 d_1 为该行第一个像素对应的判定向量。若 $d_n > 0$ 则该行步长为 E，此时 d_n 为下一行第一个像素点的判定向量。反之步长为 E+1，下一像素行第一个像素点的判定为 $d_n + 2dy$ 。这样算法的核心问题集中在求上。

改进后的算法计算一次判定向量即可画出一个像素行的全部像素点，可大大提高 Bresenham 算法的效率。

如何计算 E 成为直线绘制的关键问题，若直接计算，就需要进行除法运算，这将破坏 Bresenham 算法只需要整数加法和乘法，不需要进行小数和取整运算的优点。本文所用改进的 Bresenham 绘制直线算法用循环减法来计算 E，从而避免了除法运算，且不需要进行小数和取整运算。

对于斜率小于 1 的情况, 该算法的描述如下:

```

A ← xb - xa, B ← yb - ya, C ← A - B, E ← 1
x = xa, y = ya
while x ≤ xb do
    if C ≥ 0
        C = C - B, E = E + 1
    end if
end while
    
```

对算法的证明过程如下:

因为 $E \cdot dy - dx \leq 0, (E+1) \cdot dy - dx > 0$
 所以 $E \cdot dy \leq dx, (E+1) \cdot dy > dx$
 即 $E \leq \frac{dx}{dy} < E+1$, 等于斜率 k 的倒数取整。

3 算法实现与分析

该算法在 32 位计算机上得到验证, 其执行效率优于传统的 Bresenham 算法。程序在 Intel Pentium III 866Mhz, 256MB 内存的计算机上用 VC++6.0 编译运行, 设直线起点在 (0, 0)。

表 1 是新算法与 Bresenham 算法分别编程实现后的效率比较, 表中所列时间为坐标计算时间, 不包括将像素显示再屏幕上需要的时间。

表 1 新算法与 Bresenham 算法效率比较

终点坐标	运行时间 (μs)	
	Bresenham 算法	改进的算法
(1000, 1)	15	16
(1000, 10)	15	5
(1000, 100)	15	5
(1000, 1000)	15	15

经过试验分析, 该算法计算量的节省与待生成线段的斜率有直接关系, 斜率值越小, 节省越明显, 当斜率接近 1 时, 节省一半。

4 基于循环减法的硬件加速

考虑到在非嵌入式领域, 因为 CPU 运算速度快、存储器容量大, 图形绘制中通常都是不需要考虑运算效率直接使用的。设计不含除法运算的 Bresenham 算法, 是因为在嵌入式领域, 首先需要考虑的是算法中

的除法操作是否是必须的, 下面给出硬件实现图形绘制的分析。

从上面的分析可知, 用减法取代除法运算是加速直线绘制算法的必然之选。由于除法的本质其实就是在做减法运算和移位运算, 那么改进算法的核心就是如何加速循环减法, 同时兼顾硬件的复杂度。不管采用哪种算法, 只要是以循环减法为基本原理实现运算, 都能在一定程度上达到缩短计算时间的目的。

用循环减法实现除法需要关注操作数的预处理。一般的移位操作, 初始状态时, 寄存器 HI 用于存放余数, LO 用于存放被除数; 运算过程中, 当被除数左移补充 HI 后, LO 用于存放新产生的商 f。改进后的移位操作忽略被除数前面无效的 0, 直接把 LO 中的最高有效位补充到 HI 中, LO 的低位用 0 补充^[12]。例如 3/2 (转化为 4 位的二进制运算 0011 ÷ 0010) 的运算过程如图 2 所示。

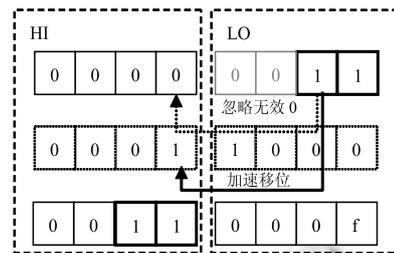


图 2 基于循环减法的移位操作加速过程

这样可以减少运算步骤, 相应地减少时钟周期。减少的运算步骤次数为, $n - m - k + 1 = n - (m + k) + 1$, n 表示除法位数, m 表示被除数前面无效 0 的个数, k 表示除数的有效位数。从公式中可以看出, m + k 越大, 即除数相对于被除数越大, 此时的除法运算也越快。

5 结论

本文所用改进的 Bresenham 绘制直线算法, 采用循环减法来计算斜率的倒数, 从而避免了除法运算, 一次运算得到多个像素, 并且不需要进行小数和取整运算。经验证, 其执行效率优于传统的 Bresenham 算法。此外, 在嵌入式软件编程中, 为了节省 CPU 运行时间, 应尽可能避免使用除法, 该算法在实际应用和 (下转第 230 页)

⑥ $C_0' \leftarrow \text{encrypt}(E_k, P_i);$

⑦ end for

⑧ $P \leftarrow \text{reform_data}(P);$ //对解密后的明文进行重组, 去掉添加的特殊字符

⑨ return P;

⑩ 算法结束

3.3 加解密的具体实现

根据 X-IDEA 算法进行的加密程序设计如图 2 所示, C:\t1.jpg 是加密源文件, 而 tt1.jpg 是加密后的源文件, 通过图 2 的加密后可以发现 tt1.jpg (图 3 所示) 已经被加密而无法查看其原内容, 而使用该加解密程序进行对应的解密程序可以还原 tt1.jpg 到 t1.jpg, 由此可以看出该算法的加解密过程是可以实现的, 并且在 Android 基础上进行了验证, 完成了手机通信中的图文加解密技术。



图 2 加解密应用程序



图 3 TT1.jpg 原图

4 结论

本文首先对手机即时通信系统进行了介绍, 然后着重介绍了该系统中图文的加密算法。先对 IDEA 进行了简单的介绍, 然后结合手机通信的特点, 对 IDEA 算法进行了改进, 提出了 X-IDEA 算法原理, 并设计了该算法的实现流程, 最后设计了加解密程序实现对图片的加解密, 达到了确保图片文件传输的安全, 随后将之引入到手机上运行也得到了预期目标。

参考文献

- 1 公磊,周聪.基于 Android 的移动终端应用程序开发与研究.计算机与现代化,2008,(8):85-89.
- 2 贾琴勇.即时通讯系统的研究与实现[硕士学位论文].武汉:武汉理工大学,2008.28-36.
- 3 潘凤,王华军,苗放,李刚.基于 XMPP 协议和 Openfire 的即时通信系统的开发.计算机时代,2008,(3):15-19.
- 4 张海燕.Java 多线程技术在手机互联网中的应用.农业网络信息,2008,(3):97-98.

(上接第 247 页)

推广中具有一定的意义。

参考文献

- 1 唐荣锡,汪嘉业.计算机图形学教程.北京:科学出版社,1990.60-65.
- 2 Bresenham JE. Algorithm for computer control of a digital plotter. IBM Systems Journal, 1965,4(1):25-30.
- 3 Rokne J, Rao Y. Double-Step incremental linear interpolation. ACM Transactions on Graphics, 1992,11(2):183-192.
- 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):42-43.
- 5 Foley JD. Introduction to Computer Graphics. Beijing: China Machine Press, 2004:48-56.
- 6 贾银亮,张焕春,经亚枝.Bresenham 直线生成算法的改进.中

国图象图形学报,2008,13(1):158-161.

- 7 郑宏珍,赵辉.改进的 Bresenham 直线生成算法.中国图象图形学报,1999,7(14):606-608.
- 8 苗兰芳,刘新国.自适应多步位移码直线绘制算法.软件学报,2002,(13)4:637-642.
- 9 刘晶,李俊,孙涵.改进的 Bresenham 直线生成算法.计算机应用与软件,2008,25(10):67-71.
- 10 孙岩,唐棣.并行的 Bresenham 直线生成算法.计算机工程与应用,2001,37(21):37-42.
- 11 Yao C, Rokne JG. Bi-Directional incremental linear interpolation. Computer & Graphics, 1996,20(2):295-305.
- 12 宣淑巍,李晓江,马成炎.一种基于循环减法原理除法器的加速方法.微电子学与计算机,2009,26(12):12-15.