

## ming1006的专栏

目录视图

摘要视图

RSS 订阅

## 个人资料



ming1006

访问： 77571次

积分： 855

等级： BLOG &gt; 3

排名： 千里之外

原创： 11篇 转载： 5篇

译文： 0篇 评论： 45条

## 文章搜索

## 文章分类

nios ii (6)  
液晶屏 (7)  
Micro SD卡 (3)  
FAT16 (1)  
MSP430 (3)  
画图算法 (1)  
液晶屏 stm32 (1)  
51单片机 (1)  
C51 (2)

## 文章存档

2013年09月 (1)  
2013年04月 (1)  
2012年10月 (1)  
2012年09月 (1)  
2012年07月 (1)

展开

## 阅读排行

Micro SD卡 (TF卡) sp (25839)  
我的5110画图库--Nokia! (8635)  
nios ii之Micro SD卡 (TF (6631)  
nios ii 之5110液晶屏 (6\* (6307)

【专家问答】韦玮：Python基础编程实战专题 【知识库】Swift资源大集合 【公告】博客新皮肤上线啦 CSDN福利第二期

## 直线光栅化算法--Bresenham算法

2012-09-22 12:28

2566人阅读

评论(0) 收藏 举报

分类： 液晶屏 (6) 画图算法

好久没写文章了，最近在用LCD5110写各种画图函数，写完之后发现浮点数和乘法的使用偏多，感觉会对画图速度有影响，于是百度了一下。刚才发现了一篇文章，是介绍直线光栅化算法中常用到的Bresenham算法，用这个算法就可以在画直线时不使用浮点数，乘法的使用也大多被加减法代替了，大大提高了作图速度。在此先把这篇文章转载过来，他对算法原理讲得很清楚简洁易懂，希望对大家有帮助。

以下是原文：

Bresenham算法是计算机图形学典型的直线光栅化算法。

## • 从另一个角度看直线光栅化显示算法的原理

由直线的斜率确定选择在x方向或y方向上每次递增（减）1个单位，另一变量的递增（减）量为0或1，它取决于实际直线与最近光栅网格点的距离，这个距离的最大误差为0.5。

## • 1) Bresenham的基本原理

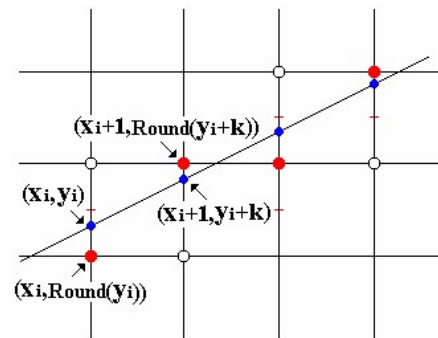
假定直线斜率k在0~1之间。此时，只需考虑x方向每次递增1个单位，决定y方向每次递增0或1。设

直线当前点为(x<sub>i</sub>, y<sub>i</sub>)直线当前光栅点为(x<sub>i</sub>, y<sub>i</sub>)

则

下一个直线的点应为(x<sub>i</sub>+1, y<sub>i</sub>+k)

下一个直线的光栅点

或为右光栅点(x<sub>i</sub>+1, y<sub>i</sub>) (y方向递增量0)或为右上光栅点(x<sub>i</sub>+1, y<sub>i</sub>+1) (y方向递增量1)记直线与它垂直方向最近的下光栅点的误差为d，有：d=(y<sub>i</sub>+k)-y<sub>i</sub>，且

0≤d≤1

当d<0.5：下一个像素应取右光栅点(x<sub>i</sub>+1, y<sub>i</sub>)当d≥0.5：下一个像素应取右上光栅点(x<sub>i</sub>+1, y<sub>i</sub>+1)

5110液晶屏初始化和data (4533)  
SD卡中FAT16文件分析 (4174)  
msp430 学习经验总结 (3636)  
error: msp430:Could not (2804)  
nios ii 之 LCD 1602 (2679)  
直线光栅化算法--Bresen (2564)

#### 评论排行

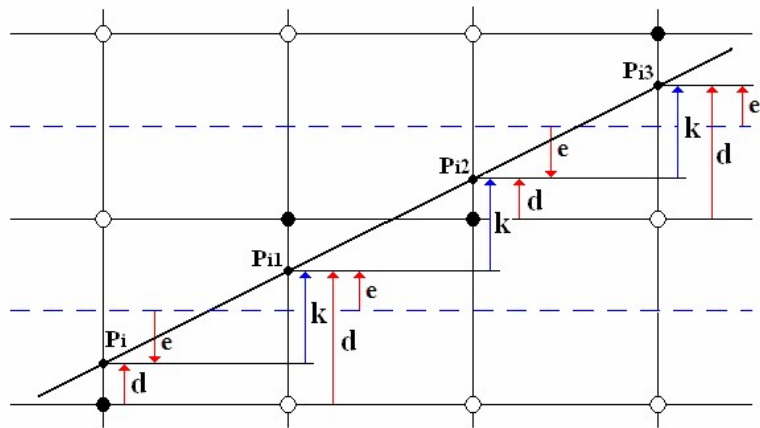
nios ii之Micro SD卡 (TF (12)  
Micro SD卡 (TF卡) sp (12)  
nios ii 之 LCD 1602 (5)  
SD卡中FAT16文件分析 (5)  
nios ii 之5110液晶屏 (6\* (3)  
5110液晶屏初始化和data (3)  
我的5110画图库--Nokia! (2)  
nios ii 之 使用自带LCD 1 (2)  
C51, faster, faster, we (1)  
msp430 学习经验总结 (0)

#### 推荐文章

\*Android官方开发文档Training系列课程中文版: 网络操作之XML解析  
\*Delta - 轻量级JavaWeb框架使用文档  
\*Nginx正反向代理、负载均衡等功能实现配置  
\*浅析ZeroMQ工作原理及其特点  
\*android源码解析 (十九) -->Dialog加载绘制流程  
\*Spring Boot 实践折腾记 (三): 三板斧, Spring Boot下使用Mybatis

#### 最新评论

nios ii 之5110液晶屏 (6\*8、8\*16 longcai1988: LZ, 有个问题请教下, 我采用你的程序, 其它都正常, 图片整屏显示没有, 这个你试过没, 我看你程序里面也没...  
nios ii之Micro SD卡 (TF卡) spi qq\_25168835: 楼主我目前再用AVR系列M128编写MicroSD程序, 楼主能发一份程序给小弟学习一下吗, 7094...  
SD卡中FAT16文件分析 lijie33402: @u012480047:你好你测试成功的吗 是用的是什么sd卡  
SD卡中FAT16文件分析 lijie33402: 博主 那测试的话sd卡是不是要买一些专门的型号啊  
我的5110画图库--Nokia 5110液晶 u012727766: 大神, 膜拜, 学习中  
nios ii 之5110液晶屏 (6\*8、8\*16 u013541878: 好东西  
Micro SD卡 (TF卡) spi 模式实 yzlijinkui: 您好, 我用AVR单片机编了个程序, 1G卡能正常复位、初始化, 但读写一个扇区时, 返回值都正常, 但读出的...  
Micro SD卡 (TF卡) spi 模式实 qq\_20193017: 您好, 可以发一份SPI驱动SD卡的代码给我嘛? 505500894@qq.com, 太感谢您了!  
5110液晶屏初始化和datasheet详 u014183377: 非常好的文章!!!  
Micro SD卡 (TF卡) spi 模式实



如果直线的(起)端点在整数点上, 误差项d的初值:  $d_0=0$ ,  
x坐标每增加1, d的值相应递增直线的斜率值k, 即:  $d=d+k$ 。

一旦 $d \geq 1$ , 就把它减去1, 保证d的相对性, 且在0-1之间。

令 $e=d-0.5$ , 关于d的判别式和初值可简化成:

e的初值 $e_0=-0.5$ , 增量亦为k;

$e < 0$ 时, 取当前像素 $(x_i, y_i)$ 的右方像素 $(x_i+1, y_i)$ ;

$e > 0$ 时, 取当前像素 $(x_i, y_i)$ 的右上方像素 $(x_i+1, y_i+1)$ ;

$e=0$ 时, 可任取上、下光栅点显示。

Bresenham算法的构思巧妙: 它引入动态误差e, 当x方向每次递增1个单位, 可根据e的符号决定y方向每次递增0或1。

$e < 0$ , y方向不递增

$e > 0$ , y方向递增1

x方向每次递增1个单位,  $e = e + k$

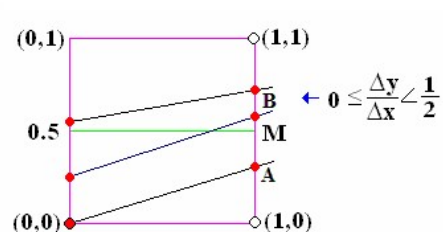
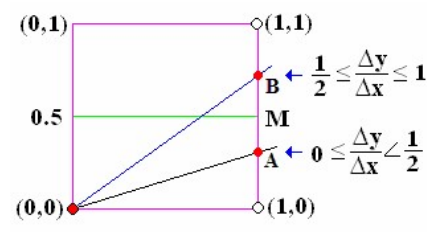
因为e是相对量, 所以当 $e > 0$ 时, 表明e的计值将进入下一个参考点(上升一个光栅点), 此时须:  $e = e - 1$

#### 2) Bresenham算法的实施——Rogers 版

通过(0,0)的所求直线的斜率大于0.5, 它与 $x=1$ 直线的交点离 $y=1$ 直线较近, 离 $y=0$ 直线较远, 因此取光栅点(1,1)比(1,0)更逼近直线:

如果斜率小于0.5, 则反之:

当斜率等于0.5, 没有确定的选择标准, 但本算法选择(1,1)



(程序)

```
//Bresenham's line rasterization algorithm for the teal octal.
```

```
//The line end points are (xs,ys) and (xe,ye) assumed not equal.
```

```
// Round is the integer function.
```

```
// x,y, Δx, Δy are the integer, Error is the real.
```

```
//Initialize variables
```

```
x=xs
```

```
y=ys
```

```
Δx = xe - xs
```

```
Δy = ye - ys
```

```
//initialize e to compensate for a nonzero intercept
```

```
Error = Δy/Δx-0.5
```

```
//begin the main loop
```

```
for i=1 to Δx
```

```
WritePixel (x, y, value)
```

```
if (Error ≥ 0) then
```

ming1006: @snowxiaoer:您好!  
我的下篇文章就有全部代码, 可以自己复制, 或者在我的资源里面下都可以

```
y=y+1
Error = Error -1
end if
x=x+1
Error = Error +Δy/Δx
next i
finish
```

#### • 3) 整数Bresenham算法

上述Bresenham算法在计算直线斜率和误差项时要用到浮点运算和除法, 采用整数算术运算和避免除法可以加快算法的速度。

由于上述Bresenham算法中只用到误差项(初值 $Error = \Delta y / \Delta x - 0.5$ )的符号因此只需作如下的简单变换:

$$NError = 2 * Error * \Delta x$$

即可得到整数算法, 这使本算法便于硬件(固件)实现。

(程序)

```
//Bresenham's integer line rasterization algorithm for the first octal.
//The line end points are (xs,ys) and (xe,ye) assumed not equal. All variables are assumed integer.
//initialize variables
x=xs
y=ys
Δx = xe -xs
Δy = ye -ys
//initialize e to compensate for a nonzero intercept
NError =2*Δy-Δx      //Error =Δy/Δx-0.5
//begin the main loop
for i=1 to Δx
    WritePixel (x, y)
    if (NError >=0) then
        y=y+1
        NError = NError -2*Δx //Error = Error -1
    end if
    x=x+1
    NError = NError +2*Δy      //Error = Error +Δy/Δx
next i
finish
```

#### • 4) 一般Bresenham算法

要使第一个八卦的Bresenham算法适用于一般直线, 只需对以下2点作出改造:

当直线的斜率 $|k| > 1$ 时, 改成y的增量总是1, 再用Bresenham误差判别式确定x变量是否需要增加1; x或y的增量可能是“+1”或“-1”, 视直线所在的象限决定。

(程序)

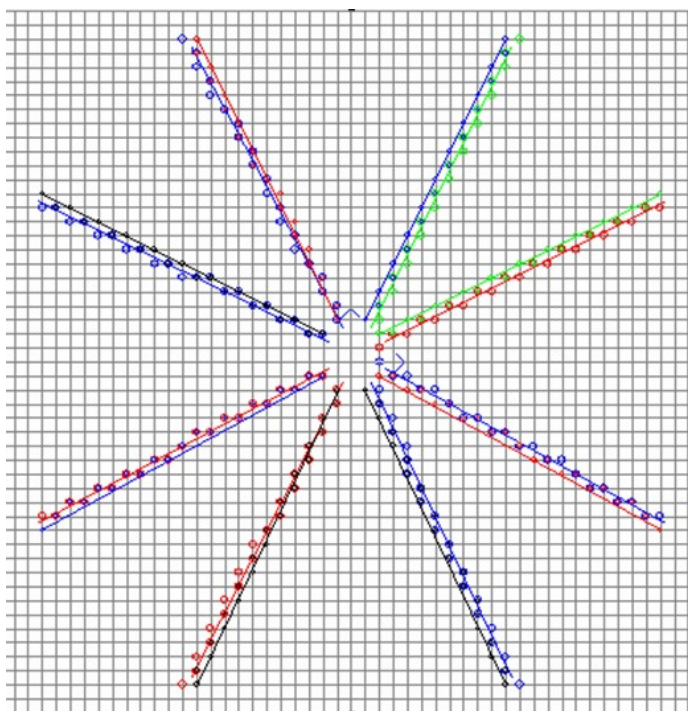
```
//Bresenham's integer line rasterization algorithm for all quadrants
//The line end points are (xs,ys) and (xe,ye) assumed not equal. All variables are assumed integer.
//initialize variables
x=xs
y=ys
Δx = abs(xe -xs)      //Δx = xe -xs
Δy = abs(ye -ys)      //Δy = ye -ys
sx = isign(xe -xs)
sy = isign(ye -ys)
//Swap Δx and Δy depending on the slope of the line.
if Δy > Δx then
    Swap(Δx,Δy)
    Flag=1
else
```

```

Flag=0
end if
//initialize the error term to compensate for a nonzero intercept
NError =2*Δy-Δx
//begin the main loop
for i=1 to Δx
  WritePixel(x, y , value)
  if (Nerror>=0) then
    if (Flag) then  //Δy>Δx, Y=Y+1
      x=x+sx
    else
      y=y+sy
    end if      // End of Flag
    NError = NError -2*Δx
  end if      // End of Nerror
  if (Flag) then  //Δy>Δx, X=X+1
    y=y+sy
  else
    x=x+sx
  end if
  NError = NError +2*Δy
next i
finish

```

- 例子



顶 踩  
0 0

上一篇 [error: msp430:Could not access device - security fuse is blown的解决办法](#)

下一篇 [我的5110画图库--Nokia 5110液晶屏实现画点、画线、画矩形、画圆和画图](#)

液晶屏 (6)

• n... • 5...

2012-2012-02-06 02-04

阅读 阅读

6307 4534

• 任... • n...

2012-2012-

参考知识库



算法与数据结构知识库

3549 关注 | 4535 收录

猜你在找

- iOS8开发技术 (Swift版)：iOS基础知识
- 光栅直线生成算法
- 三维游戏引擎开发-图形理论基础
- 用Bresenham算法画直线二
- 数据结构和算法
- 直线扫描——Bresenham算法
- Linux运维高薪入门及进阶全新经典视频-老男孩Linux\$ OpenGL改进Bresenham算法画直线
- 零基础学HTML 5实战开发 (第一季)
- C语言实现的DDA和Bresenham直线算法

乐视电视盒子, 开放购买

\$79.99 + 免运费！ 硬件免费+1年乐视超级影视会员

查看评论

暂无评论

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

\* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题

Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件 Docker OpenStack

VPN Spark ERP IE10 Eclipse CRM JavaScript 数据库 Ubuntu NFC WAP jQuery

BI HTML5 Spring Apache .NET API HTML SDK IIS Fedora XML LBS Unity

Splashtop UML components Windows Mobile Rails QEMU KDE Cassandra CloudStack

FTC coremail OPhone CouchBase 云计算 iOS6 Rackspace Web App SpringSide Maemo

Compuware 大数据 aptech Perl Tornado Ruby Hibernate ThinkPHP HBase Pure Solr

Angular Cloud Foundry Redis Scala Django Bootstrap