

文章编号:1007-2780(2009)04-0592-05

基于 Bresenham 的仪用液晶屏图形生成算法应用研究

李银华, 韩郡业

(郑州轻工业学院 电气信息工程学院, 河南 郑州 450002, E-mail: zzfcc@126.com)

摘 要: 针对 ST7920 液晶控制器和点阵式液晶显示模块 KXM12864M(LCD Module, LCM) 的特点, 画出了 ST7920 与单片机的硬件接口电路。在 Bresenham 算法的基础上, 采用离散的整数增量来代替斜率增量的计算方法, 给出了适合 MCU 作图运算的优化算法推导过程, 通过 C 语言编程实现了画线、画圆等操作。工程实际使用结果表明, 该优化算法具有简单可行、移植性好的特点, 为在中小型仪器仪表显示设备上绘制其它图形提供了新的途径。

关 键 词: 液晶显示屏; Bresenham 算法; 差值比较; 作图
中图分类号: TP391 **文献标识码:** A

1 引 言

液晶显示器 (Liquid Crystal Display, LCD) 作为人机交换的重要组成部分, 因具有低功耗、体积小、显示信息量大、散热小、寿命长等诸多特点, 被广泛应用在各种仪器仪表领域^[1,2]。随着自动化水平的提高, 人们对仪器仪表的操作界面要求越来越高, 不仅需要 LCD 能够显示字符或汉字, 还要求能够显示各种图形, 使操作更加便捷。图形的绘制, 都是在画点、线的基础上来完成的, 在人机系统中需要有相应的生成算法才能完成图形的绘制。现在的图形生成算法有很多, 其中最著名的是 Bresenham 算法^[3,4], 现已有许多文献对该算法进行了超快速的改进。但是在仪器仪表设备中, 对于单片机控制的仪用中小尺寸 LCD, 受 MCU 处理速度和内存空间等众多因素的限制, 这些超快速、有效的改进算法难以得到实际应用。为此, 寻找一种适合仪用液晶屏图形生成的优化算法十分必要。

2 典型仪用 LCD 接口电路设计

LCM 与 MCU 的接口有直接访问和间接访问两种方式。直接访问, 是利用 MCU 数据总线和控制信号直接经过 I/O 访问形式控制液晶控制器; 间接访问, 是利用 MCU 通过并行接口间接实现对液晶控制器的控制。KXM12864M 是一

种点阵式液晶显示模块, 控制芯片为 ST7920, 内含中文字库, 可提供 8 位或 4 位的并行数据接口方式和串行数据接口方式^[5]。出于做图的考虑, 在此选择 8 位并行数据接口直接访问形式^[6], LCM 的数据端接 MCU 的 PB 口, PSB 口上拉, 其它控制线与 PD 口连接。LCM 与 MCU 都用 +5 V 供电。液晶显示模块原理如图 1 所示。

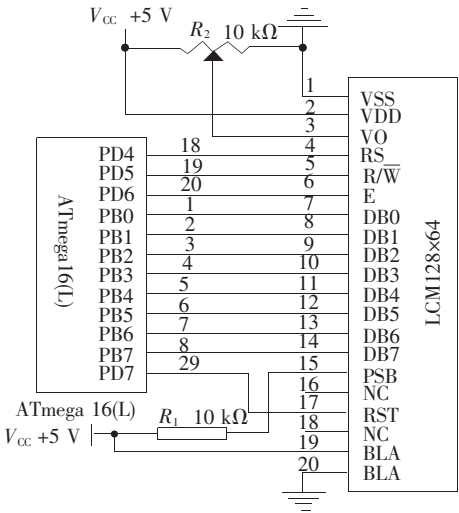


图 1 液晶显示模块原理图

Fig. 1 Diagram of LCD module principle

3 Bresenham 算法及优化

Bresenham 算法是在计算机图形学领域发展

较早的算法之一,虽然该算法有很多优点(如不进行小数运算),并且发展出了很多改进型算法^[7,8],但是并不适合在 MCU 上运算。本文根据 MCU 的特点,在 Bresenham 算法的基础上,给出了该算法的优化推导公式,通过编程^[9,10]实现了在仪用 LCM 上画线、画圆的操作。其基本原理为:在已知两点中画出最逼近直线和已知圆心、半径作圆时,都是使用整数的加、减法和移位决定哪些点是被选中的点,即采用离散的整数增量来代替斜率增量的计算。

3.1 Bresenham 算法的基本原理

Bresenham 算法的基本原理为:在光栅显示器中构造一组水平和垂直的虚拟网格线,从起点出发,利用整数的运算,按照绘图顺序依次计算与当前垂直线的交点,与该交点最近的像素点为被选中的点^[3]。

在 Bresenham 算法优点的基础上,优化后的算法运用比较先后 2 个像素点的差值,通过推导出决策因子来确定选中的点,更优地保持了曲线的线性,同时对控制器的处理速度要求也不高,能够胜任各种仪器仪表显示设备的作图要求。

3.2 画线优化算法的推导

假设直线位于第一象限内,斜率在 $[0, 1]$ 之间,其它象限的直线可通过对称性得到。设直线方程:

$$F(X, Y) = aX + bY + c = 0 \quad (X_1 < X < X_2, Y_1 < Y < Y_2) \quad (1)$$

令 $dx = X_2 - X_1, dy = Y_2 - Y_1$, 则有斜率:

$$-\frac{a}{b} = \frac{dy}{dx} \quad (2)$$

从第一个点开始,有 $F(X_1, Y_1) = aX_1 + bY_1 + c = 0$ 。

下面求线段 $aX + bY + c = 0$ 与 $X = X_1 + 1$ 的交点:

由 $a(X_1 + 1) + bY + c = 0$, 可求出交点坐标 $Y = \frac{-c - a(X_1 + 1)}{b}$ 。所以交点与 M 的 Y 坐标差值:

$$\begin{aligned} \text{Sub1} &= \frac{-c - a(X_1 + 1)}{b} - (Y_1 + 0.5) = \\ &= -\frac{a}{b} - 0.5 \end{aligned} \quad (3)$$

即 Sub1 的初始值为 $-\frac{a}{b} - 0.5$ 。

则可得条件:当 $\text{Sub1} = -\frac{a}{b} - 0.5 > 0$ 时,下

个点为 U,即当前点的右上相邻点;反之,下个点为 B,即当前点的右相邻点。M、U、B 3 点的位置关系如图 2 所示。将式(2)代入式(3),则:

$$\text{Sub1} = \frac{dy}{dx} - 0.5 \quad (4)$$

因为是循环中判断 Sub,所以需要求出循环下的 Sub 表达式,通过比较下下个像素点,即经过第 2 次比较,推导得出决策因子。

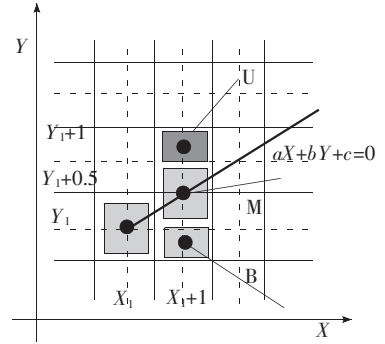


图 2 画线取点示意图

Fig. 2 Schematic diagram of drawing line and choosing point

下面求 $X = X_1 + 2$ 时的 Sub, 即 Sub2:

(a) 如果下下个点是下个点的右上相邻点, 则有:

$$\begin{aligned} \text{Sub2} &= \frac{-c - a(X_1 + 2)}{b} - (Y_1 + 1.5) = \\ &= -\frac{2a}{b} - 1.5 \end{aligned} \quad (5)$$

故 Sub 差值为:

$$D_{\text{sub}} = \text{Sub2} - \text{Sub1} =$$

$$-\frac{2a}{b} - 1.5 - \left(-\frac{a}{b} - 0.5\right) = -\frac{a}{b} - 1 \quad (6)$$

代入式(2),得:

$$D_{\text{sub}} = \frac{dy}{dx} - 1 \quad (7)$$

(b) 如果下下个点是下个点的右相邻点, 则有:

$$\begin{aligned} \text{Sub2} &= \frac{-c - a(X_1 + 2)}{b} - (Y_1 + 0.5) = \\ &= -\frac{2a}{b} - 0.5 \end{aligned} \quad (8)$$

Sub 差值为:

$$D_{\text{sub}} = \text{Sub2} - \text{Sub1} =$$

$$-\frac{2a}{b} - 0.5 - \left(-\frac{a}{b} - 0.5\right) = -\frac{a}{b} \quad (9)$$

代入式(2),得:

$$D_{\text{sub}} = \frac{dy}{dx} \quad (10)$$

需要判断的是差值的正负号,同时应尽量避免除法和浮点数的计算,故把公式进行优化。总结为:取 $Sub1 = 2dy - dx$, 当 $Sub1 > 0$ 时, $D_{sub} = 2dy - 2dx$, 选择当前点的右上相邻点;当 $Sub1 < 0$ 时, $D_{sub} = 2dy$, 选择当前点的右相邻点。

从以上的描述中就可以画出任意两点间的直线,其 C 语言程序如下:

```
void GUI_line (uchar x1, uchar y1, uchar x2,
uchar y2)
{
    int dx,dy;           //定义 X,Y 方向上的增量
    int sub;
    int temp;            //定义数据交换时的变量
    if(x1>x2)             //X 坐标的值,起点大于
                          //终点,交换数据
    {
        temp=x2;
        x2=x1;
        x1=temp;
    }
    if(y1>y2)            //Y 坐标的值,起点大于
                          //终点,交换数据
    {
        temp=y2;
        y2=y1;
        y1=temp;
    }
    dx=x2-x1;           //X 方向上的增量
    dy=y2-y1;           //Y 方向上的增量
    /* 使用 Bresenham 法进行画直线 */
    if(dx>dy)            //以 X 方向上的增量为基准
    {
        sub=2 * dy-dx;   //判断下个点的位
        while(x1!= x2)   //判断是否到终点
        {
            GUI_Point(x1,y1,1); //画起点
            x1++;           //X 轴方向上增加
            if(sub>0)       //判断下下个点的位置
            {
                y1++;      //该点为右上相邻点
                sub+=2 * dy-2 * dx;
            }
            else
                sub+=2 * dy;   //该点为右相邻点
            GUI_Point(x1,y1,1); //调用画点函数
        }
    }
    else                  //以 Y 方向上的增量为基准
    {
        sub=2 * dx-dy;     //判断下个点的位
        while(y1!= y2)     //判断是否到终点
        {
            GUI_Point(x1,y1,1); //画起点
            y0++;           //Y 轴方向上增加
            if(sub>0)       //判断下下个点的位置
            {
                x1++;      //该点为右上相邻点
                sub+=2 * dy-2 * dx;
            }
            else
                sub+=2 * dy;   //该点为右相邻点
            GUI_Point(x1,y1,1); //调用画点函数
        }
    }
}
```

```
sub+=2 * dy;   //该点为右相邻点
}
GUI_Point(x1,y1,1); //调用画点函数
}
else          //以 Y 方向上的增量为基准
{
    sub=2 * dx-dy;     //判断下个点的位
    while(y1!= y2)     //判断是否到终点
    {
        GUI_Point(x1,y1,1); //画起点
        y0++;           //Y 轴方向上增加
        if(sub>0)       //判断下下个点的位置
        {
            x1++;      //该点为右上相邻点
            sub+=2 * dy-2 * dx;
        }
        else
            sub+=2 * dy;   //该点为右相邻点
        GUI_Point(x1,y1,1); //调用画点函数
    }
}
```

3.3 画圆优化算法的推导

假设在第 1 象限的 $1/8$ 画圆弧,其它象限的圆弧通过对称性可得到。圆的半径为 R ,圆的方程为 $X^2 + Y^2 = R^2$,则画圆的起点坐标是 $(0, R)$ 。从起点 A 点开始向右下方逐点来寻找圆弧需要的点,下一个点可能是右相邻点 B,或是右下相邻点 U。通过判断待选点到圆心的平方与半径平方的差值来决定被选点,应尽量选离圆弧最近的点作为显示点,3 点的位置关系如图 3 所示。

设起点为 $A(X_i, Y_i)$,右相邻 B 点为 $(X_i + 1,$

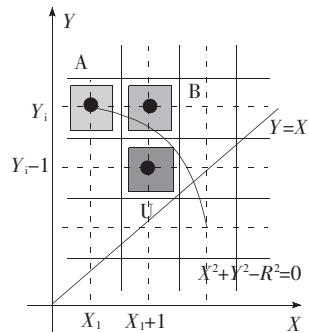


图 3 画圆取点示意图

Fig. 3 Schematic diagram of drawing circle and choosing point

Y_i),右下相邻 U 点为 (X_i+1,Y_i-1) 。如是 B 点,其差值为 $(X_i+1)^2+Y_i^2-R^2$;如是 U 点,其差值为 $R^2-[(X_i+1)^2+(Y_i-1)^2]$,一个在圆弧外,另一个在圆弧内。这两个点的距离:

$$\begin{aligned} \text{Sub1} &= (X_i+1)^2+Y_i^2+ \\ &[(X_i+1)^2+(Y_i-1)^2]-2R^2 = \\ &2X_i^2+2Y_i^2+4X_i-2Y_i+3-2R^2 \end{aligned} \quad (11)$$

当 $\text{Sub1} \geq 0$ 时,应该取 U,否则取 B。代入画圆的起点 $(0,R)$,则有:

$$\text{Sub1} = 3-2R \quad (12)$$

因为是循环中要判断 Sub1,所以需求出循环下的 Sub1 表达式,通过优化算法得出决策因子。下面求 $X=X_i+2$ 时的 Sub,即 Sub2:

(a)如果下下个点是下个点的右下相邻点,则下下个点的坐标是 $P(X_i+2,Y_i-1)$ 或 $Q(X_i+2,Y_i-2)$,计算得出 Sub2 为:

$$\begin{aligned} \text{Sub2} &= (X_i+2)^2+(Y_i-1)^2+ \\ &[(X_i+2)^2+(Y_i-2)^2]-2R^2 = \\ &2X_i^2+2Y_i^2+8X_i-6Y_i+13-2R^2 \end{aligned} \quad (13)$$

故 Sub 差值为:

$$D_{\text{sub}} = \text{Sub2}-\text{Sub1} = 10+4(X_i-Y_i) \quad (14)$$

(b)如果下下个点是下个点的右相邻点,则下下个点的坐标是 $P(X_i+2,Y_i)$ 或 $Q(X_i+2,Y_i-1)$,同样计算得出 Sub2 为:

$$\begin{aligned} \text{Sub2} &= (X_i+2)^2+Y_i^2+ \\ &[(X_i+2)^2+(Y_i-1)^2]-2R^2 = \\ &2X_i^2+2Y_i^2+8X_i-2Y_i+9-2R^2 \end{aligned} \quad (15)$$

Sub 差值为:

$$D_{\text{sub}} = \text{Sub2}-\text{Sub1} = 6+4X_i \quad (16)$$

通过以上的优化推导,可以总结为:

取 $\text{Sub1}=3-2R$ 。当 $\text{Sub} > 0$ 时, $D_{\text{sub}}=10+4(X_i-Y_i)$,选择右下相邻点;当 $\text{Sub} < 0$ 时, $D_{\text{usb}}=6+4X_i$,选择右相邻点。

基于该优化算法推导,写出的 C 语言程序如下:

```
void Draw_circle(uchar x0,uchar y0,uchar r)
{ // x0, y0 是画圆位置的圆心,r 是半径
  int a,b;
```

```
  int sub1;
  a=0;
  b=r;
  sub1=3-2*r; //判断下个点位置的标志
  while(a<=b)
  { //运用对称性画出每个 1/8 分圆
    GUI_Point(x0-b,y0-a,1); //3
    GUI_Point(x0+b,y0-a,1); //0
    GUI_Point(x0-a,y0+b,1); //1
    GUI_Point(x0-b,y0-a,1); //7
    GUI_Point(x0-a,y0-b,1); //2
    GUI_Point(x0+b,y0+a,1); //4
    GUI_Point(x0+a,y0-b,1); //5
    GUI_Point(x0+a,y0+b,1); //6
    GUI_Point(x0-b,y0+a,1);
    a++;
  }
  /**使用 Bresenham 算法画圆**/
  if(sub1<0)
    sub1 +=4*a+6; //选择右相邻点
  else
  {
    sub1 +=10+4*(a-b); //选择右下相邻点
    b--;
  }
  GUI_Point(x0+a,y0+b,1);
}
```

4 结 论

以 ST7920 控制的 LCM 为例,详细推导了优化后的 Bresenham 算法。利用两次差值的比较实现了在仪用显示设备上的作图需要,既实现了设备显示的要求,又满足控制显示设备的 MCU 的处理能力,通过 C 语言编程实现了画线、画圆的操作。经在工程实际中使用,结果表明:该算法具有可行性,稳定性高,移植性强,且适用于在仪器仪表等中小尺寸显示模块上作图使用。

参 考 文 献:

[1] 李良成,张永顺,张璟. 基于 MSP430 单片机的液晶显示 [J]. 现代显示,2007,75(5):32-33.
[2] 王交春,王成花. 基于 DSP 的液晶显示系统设计 [J]. 现代显示,2008,91(8):44-47.

- [3] Foley James D. 计算机图形学导论 [M]. 北京:机械工业出版社,2004.
- [4] Slater Mel. 计算机图形学虚拟环境 [M]. 北京:机械工业出版社,2004.
- [5] 舒鑫,郭其一. ST7920 点阵式液晶显示模块的应用研究和编程 [J]. 液晶与显示,2007,22(2):192-196.
- [6] 淮俊霞,由学志,邱永前. 片剂硬度测试仪的液晶显示界面设计 [J]. 液晶与显示,2008,23(3):371-376.
- [7] 贾银亮,张焕春,经亚枝. Bresenham 直线生成算法的改进 [J]. 中国图象图形学报,2008,13(1):158-161.
- [8] 石广田,翟文正,周维琴. 基于 Bresenham 的高效直线生成集成算法 [J]. 组合机床与自动化加工技术,2006,(7): 22-25.
- [9] 沈文, Lee Eagle,詹卫前. AVR 单片机 C 语言开发入门指导 [M]. 北京:清华大学出版社,2003.
- [10] 王海涛,刘兆甲,张文明,等. 基于 DSP 的液晶图文显示技术 [J]. 液晶与显示,2008,23(1):73-76.

Graphics Generation Algorithm Application in Instrumental LCD Drawing Based on Bresenham

LI Yin-hua, HAN Jun-ye

(College of Electrical and Information Engineering, Zhengzhou University of Light Industry,
Zhengzhou 450002, China, E-mail: zzfcc@126.com)

Abstract

According to the characteristics of ST7920 LCD controller and dot-matrix LCD modules (KXM12864M), this paper draws the hardware interface circuit between ST7920 and single-chip. On the basis of Bresenham algorithm, the article adopts the calculation method which uses discrete integer increments instead of rate increments and proposes deduction process of optimization algorithm which can apply to MCU drawing operations. And then the painting line, round and other operations can be realized through the C programming language. The actual project results show that the optimization algorithm has simple and good-transplant characteristics. And it can offer a new approach for drawing other graphics on small and medium-sized instrumentation display equipments.

Key words: LCD; Bresenham algorithm; difference comparison; drawing

作者简介:李银华(1964—),男,湖北应城人,工学硕士,教授,主要研究方向:嵌入式系统的开发与智能仪表设计等。