

Chapter 21: Line Algorithm

Line drawing is accomplished by calculating intermediate positions along the line path between two specified endpoint positions. An output device is then directed to fill in these positions between the endpoints.

Section 21.1: Bresenham Line Drawing Algorithm

Background Theory: Bresenham's Line Drawing Algorithm is an efficient and accurate raster line generating algorithm developed by Bresenham. It involves only integer calculation so it is accurate and fast. It can also be extended to display circles and other curves.

In Bresenham line drawing algorithm:

For Slope $|m| < 1$:

Either value of x is increased

OR both x and y is increased using decision parameter.

For Slope $|m| > 1$:

Either value of y is increased

OR both x and y is increased using decision parameter.

Algorithm for slope $|m| < 1$:

1. Input two end points (x_1, y_1) and (x_2, y_2) of the line.

2. Plot the first point (x_1, y_1) .

3. Calculate

$$\text{Delx} = |x_2 - x_1|$$

$$\text{Dely} = |y_2 - y_1|$$

4. Obtain the initial decision parameter as

$$P = 2 * \text{dely} - \text{delx}$$

5. For $i = 0$ to delx in step of 1

If $p < 0$ then

$$X1 = x1 + 1$$

Pot(x_1, y_1)

$$P = p + 2\text{dely}$$

Else

$$X1 = x1 + 1$$

$$Y1 = y1 + 1$$

Plot(x_1, y_1)

$$P = p + 2\text{dely} - 2 * \text{delx}$$

End if

End for

6. END

Source Code:

```

/* A C program to implement Bresenham line drawing algorithm for |m|<1 */
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<math.h>

int main()
{
    int gdriver=DETECT,gmode;
    int x1,y1,x2,y2,dx,dy,p,i;
    initgraph(&gdriver,&gmode,"c:\\TC\\BGI");

    printf("Enter the initial points: ");
    scanf("%d",&x1);
    scanf("%d",&y1);
    printf("Enter the end points: ");
    scanf("%d",&x2);
    scanf("%d",&y2);

    putpixel(x1,y1,RED);

    dx=fabs(x2-x1);
    dy=fabs(y2-y1);
    p=(2*dy)-dx;
    for(i=0;i<dx;i++){
        if(p<0)
        {
            x1=x1+1;
            putpixel(x1,y1,RED);
            p=p+(2*dy);
        }
        else
        {
            x1=x1+1;
            y1=y1+1;
            putpixel(x1,y1,RED);
            p=p+(2*dy)-(2*dx);
        }
    }
    getch();
    closegraph();
    return 0;
}

```

Algorithm for slope $|m|>1$:

1. Input two end points (x_1,y_1) and (x_2,y_2) of the line.
2. Plot the first point (x_1,y_1) .
3. Calculate
 $dx = |x_2 - x_1|$
 $dy = |y_2 - y_1|$
4. Obtain the initial decision parameter as
 $P = 2 * dx - dy$
5. For $i = 0$ to dy in step of 1

If $p < 0$ then
 $y_1 = y_1 + 1$
 Plot(x_1,y_1)

$P = p + 2\text{delx}$

Else

$X1 = x1 + 1$

$Y1 = y1 + 1$

$\text{Plot}(x1, y1)$

$P = p + 2\text{delx} - 2 * \text{dely}$

End if

End for

6. END

Source Code:

```
/* A C program to implement Bresenham line drawing algorithm for |m|>1 */
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<math.h>
int main()
{
    int gdriver=DETECT,gmode;
    int x1,y1,x2,y2,delx,dely,p,i;
    initgraph(&gdriver,&gmode,"c:\\TC\\BGI");
    printf("Enter the intial points: ");
    scanf("%d",&x1);
    scanf("%d",&y1);
    printf("Enter the end points: ");
    scanf("%d",&x2);
    scanf("%d",&y2);
    putpixel(x1,y1,RED);
    delx=fabs(x2-x1);
    dely=fabs(y2-y1);
    p=(2*delx)-dely;
    for(i=0;i<delx;i++){
        if(p<0)
        {
            y1=y1+1;
            putpixel(x1,y1,RED);
            p=p+(2*delx);
        }
        else
        {
            x1=x1+1;
            y1=y1+1;
            putpixel(x1,y1,RED);
            p=p+(2*delx)-(2*dely);
        }
    }
    getch();
    closegraph();
    return 0;
}
```