

# Simulation of DDA (Digital Differential Analyzer) Line Generation Algorithm

Jolly Trivedi

Indira Gandhi National Open University, India

**Abstract** - Line Generation in Computer Graphics is accomplished by calculating the intermediate point coordinates along the line path between two given end points. Screen pixels are referred with integer values. Pixels which are intensified are those which lie very close to the line path if not exactly on the line path which is in the case of perfectly horizontal, vertical or 45 degrees lines only. Standard algorithms are available to determine which pixels provide the best approximation to the desired line. I will discuss one such algorithm - DDA (Digital Differential Analyzer) Algorithm in this paper.

**Keywords** - *Line Generation, DDA, Algorithm.*

## 1. Introduction

DDA Algorithm is a Line Generation Algorithm. Line is generated by calculating intermediate points coordinates along line path between two given end points. Only those pixels are intensified which lie very close to line path because they are the ones which best approximate the line.

## 2. Algorithm

The DDA Algorithm starts by calculating the slope. This slope can be expressed in DDA as

$$m = (y_2 - y_1) / (x_2 - x_1)$$

Considering a line with positive slope, if the slope is less than or equal to 1, we sample at unit x intervals ( $dx=1$ ) and compute successive y values as

$$\begin{aligned}x\{k+1\} &= x\{k\} + 1 \\y\{k+1\} &= y\{k\} + m\end{aligned}$$

Subscript k takes integer values starting from 0, for the 1st point and increases by 1 until endpoint is reached. y value is rounded off to nearest integer to correspond to a screen pixel.

For lines with slope greater than 1, we reverse the role of x and y i.e. we sample at  $dy=1$  and calculate consecutive x

values as

$$\begin{aligned}x\{k+1\} &= x\{k\} + 1/m \\y\{k+1\} &= y\{k\} + 1\end{aligned}$$

## 3. C++ Code for Simulation of DDA

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>

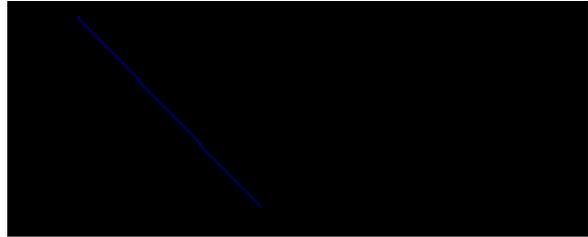
int main(void)
{
    int x,x1,y1,x2,y2,dx,dy,m;
    int gd=DETECT;
    int gm;
    initgraph(&gd,&gm,"C://TC//BGI");
    printf("Enter the values of A(x1,y1)");
    scanf("%d %d", &x1,&y1);
    printf("Enter the values of A(x2,y2)");
    scanf("%d %d", &x2,&y2);
    dx= abs(x1-x2);
    dy= abs(y1-y2);
    m= dy/dx;
    x=x1;
    y=y1;
    if(m<0)
    {
        while(x<=x2)
        {
            putpixel(x,y,WHITE);
            x=x+1;
            y=y+m;
        }
    }
    if(m>0)
    {
        while(y<=y2)
        {
            putpixel(x,y,WHITE);
            x=x+1/m;
            y=y+1;
        }
    }
}
```

```
}  
getch();  
closegraph();  
return 0;  
}
```

**Output Generated**



Line Generated is



**5. Conclusion**

After execution of simulation codes of DDA Algorithm, I come to the conclusion that for DDA algorithm, slope is the crucial factor in line generation. The line generation through DDA discussed is only for the 1st Quadrant.

**References**

[1] Alan Watt: 3D Computer Graphics

**4. FlowChart**

