PREMIER UNIVERSITY CHITTAGONG

Department of Computer Science & Engineering

Course Code : CSE-451

Course Title : Computer Graphics and Image Processing Laboratory.

Report Name: Implementation of DDA line drawing algorithm .

Date of Submission : 08/01/2023

SUBMITTED BY

MD: Mohiuddin Faysal
ID:1903610201765
Department : CSE
Semester: 7th
Section: C2

SUBMITTED TO

MD. NEAMUL HOQUE
Lecturer
Department of CSE

Abstract:
The Digital Differential Analyzer (DDA) line drawing algorithm is a widely used method for generating a line segment between two given points on a computer screen. The algorithm works by calculating the slope of the line and incrementing either the x or y coordinate by the slope at each step. This process continues until the endpoint is reached, and pixels are plotted at each step to form the line. The DDA algorithm is simple and efficient, making it a popular choice for applications such as computer graphics and image processing. However, it does have some limitations, such as limited accuracy and susceptibility to rounding errors. Despite these limitations, the DDA algorithm remains a fundamental building block in computer graphics and continues to be studied and refined.

.

Introduction:
The Digital Differential Analyzer (DDA) line drawing algorithm is a fundamental technique used in computer graphics and image processing. It is a simple and efficient method for generating a straight line between two points on a computer screen. The algorithm works by incrementally calculating the slope of the line and plotting pixels at each step to form the line. This approach is widely used because it can be implemented quickly and easily, making it a popular choice for real-time graphics applications. However, the DDA algorithm does have some limitations, such as limited accuracy and susceptibility to rounding errors. Despite these limitations, the DDA algorithm remains an essential building block in computer graphics, and it continues to be studied and refined by researchers and practitioners. In this article, we will discuss the DDA line drawing algorithm in detail, including its principles, advantages, limitations, and applications.

.

Software & hardware required:
1. Google Colab
2. Microsoft word
3. Snipping Tool

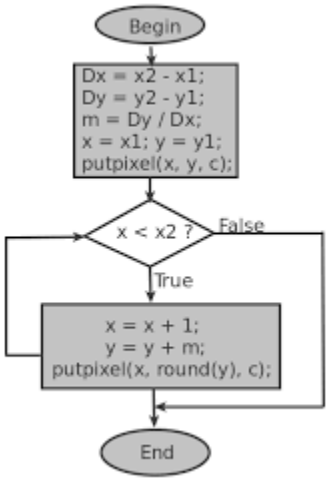
Algorithms:
1. Start
2. Get the values of the starting and ending co-ordinates i.e. (x1,y1) and (x2,y2)
3. Find the value of slope m.m=dy/dx=((y2-y1)/(x2-x1))
4. If [m] <=1 then Δx=1, Δy=mΔx xk+1=xk+1,yk+1=yk+m
5. If [m] >=1 then Δy=1, Δx=Δy/m xk+1=xk+1/m,yk+1=yk+m
6. Stop.

Pseudocode:
1. Get the input of two input ends (x1,y1) and (x2,y2)
2. Calculate m, m=(y2-y1)/(x2-x1)
3. Depending upon the absolute value of m Check the conditions:

```
If(m<1){
y=y1
for(int x=x1;x<=x2;x++;y+=m) setPixel(x,round(y));
}
```

else if(m>1){
x=x1;
for(int y=y1;y<=y2;y++;x+=1/m) setPixel(round(x),y);
}
4. Exit.

Flow Chart:



Code:

```
x1 = int(input("Enter the value of x1 : "))
y1 = int(input("Enter the value of y1 : "))
x2 = int(input("Enter the value of x2 : "))
y2 = int(input("Enter the value of y2 : "))
dx = int(x2-x1)
dy = int(y2-y1)
steps = abs(dx) if abs(dx) > abs(dy) else abs(dy)
values = {
"x":[],
"y":[]
}
values["x"].append(x1)
values["y"].append(y1)
print((x1),(y1))
for i in range(int(steps)):
cx, cy = values["x"][i],values["y"][i]
m = (y2-cy)/(x2-cx)
if m<1:
ux = cx+1
uy = cy+m
print(ux,round(uy))
elif m>1:
ux= cx+(1/m)
uy=cy+1
print(round(ux),uy)
else:
```

```
ux = ux+1
uy=uy+1
print(ux,uy)
values["x"].append(ux)
values["y"].append(uy)
```

Output:

```
Enter the value of x1 : 2
Enter the value of y1 : 3
Enter the value of x2 : 12
Enter the value of y2 : 8
2 3
3 4
4 4
5 4
6 5
7 6
8 6
9 6
10 7
11 8
12 8
```

Limitations:

DDA algorithm is simple and easier to calculate since each step has only two additions. One of the disadvantages of this algorithm is the involvement of round-off functionality. Round-off operation consumes a lot of time and accumulations of rounding off values cause accumulation error.

Discussion: In this lab class we learned about Bresenham line drawing algorithm. We learned about the implementation of bresenham line drawing algorithm.

References:

No reference used.