

CSO351: Computer Graphics

Lab Assignment 1.(a): Line Generation using DDA

Objective:

Write a program in C/C++ for implementation of line generation algorithm using DDA.

Algorithm:

- **Step 1:** Get the input of two end points : (X0, Y0) and (X1, Y1).
- **Step 2:** Calculate the difference between two end points.
$$dx = X1 - X0$$
$$dy = Y1 - Y0$$
- **Step 3:** Based on the calculated difference in step 2, identify the number of steps to put pixel. If $dx > dy$, then we need more steps in x coordinate; otherwise in y coordinate.

```
if (absolute(dx) > absolute(dy))
    Steps = absolute(dx);
else
    Steps = absolute(dy);
```

- **Step 4:** Calculate the increment in x and y coordinates.
$$Xincrement = dx / (\text{float}) \text{ steps};$$
$$Yincrement = dy / (\text{float}) \text{ steps};$$
- **Step 5:** Put the pixel by successfully incrementing x and y coordinates accordingly and complete the drawing of the line.

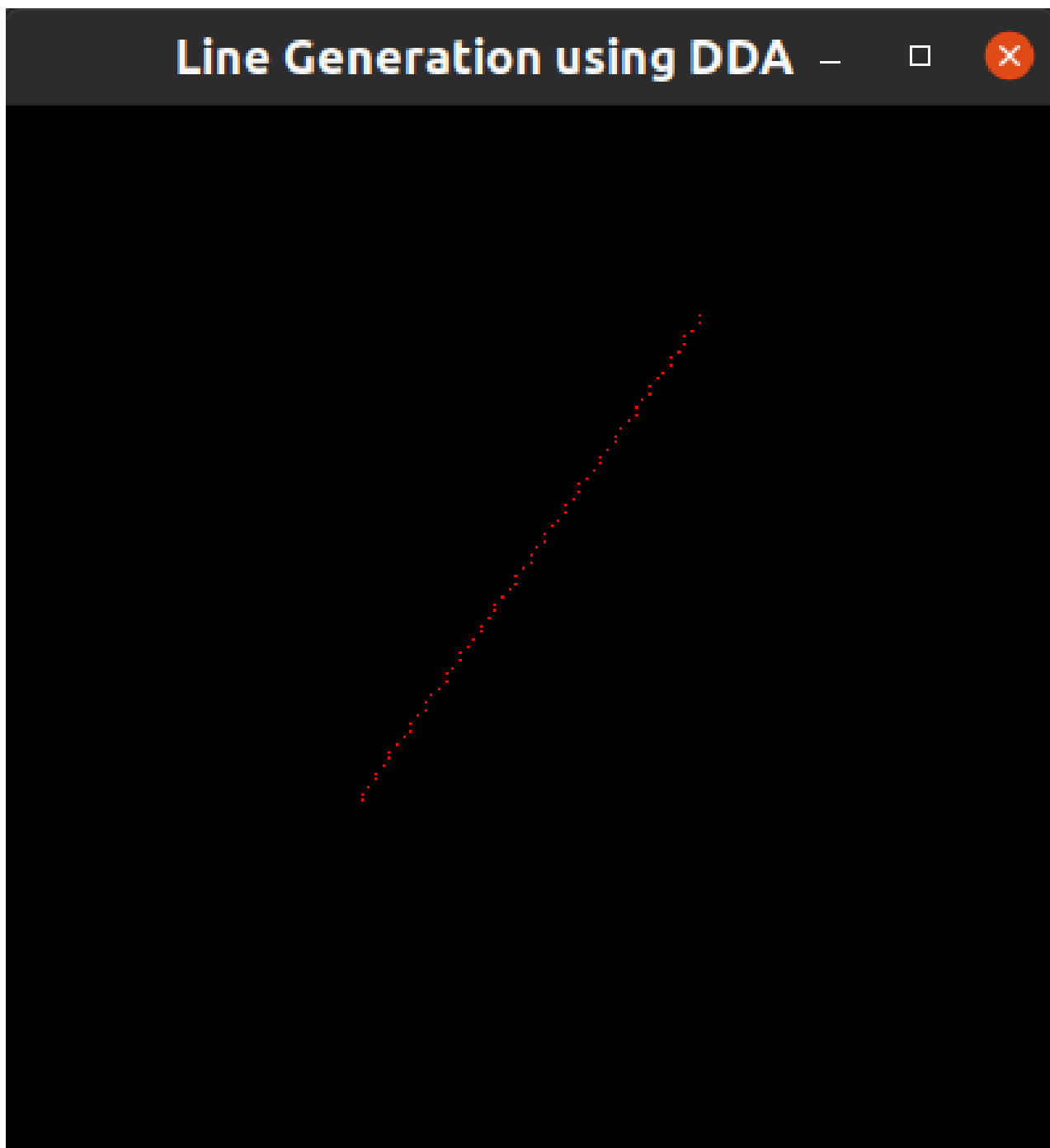
```
for(int v=0; v < Steps; v++)
{
    x = x + Xincrement;
    y = y + Yincrement;
    putpixel(Round(x), Round(y));
}
```

Result:

Input:

```
swaraj@shiv-raj-75:~/Documents/Assignments/Sem5/CG$ ./1.i  
Enter x1 and y1 : 1 1  
Enter x2 and y2 : 50 70
```

Output:



Conclusion:

- In DDA, slope is a crucial factor in line generation. The line generation through DDA is only for first quadrant.
- It uses an enormous number of floating point multiplication and divisions, so it is expensive.
- It round off the coordinates to integer that is nearest to the line.
- It is a simple algorithm to implement.

Appendix: Code

```
#include<GL/glut.h>
#include<stdlib.h>
#include<stdio.h>

float x1, x2, y1, y2;

void display(void)
{
    float dy, dx, step, x, y, k, Xin, Yin;
    dx = x2 - x1;
    dy = y2 - y1;
    if (abs(dx) > abs(dy))
        step = abs(dx);
    else
        step = abs(dy);
    Xin = dx / step;
    Yin = dy / step;
    x = x1;
    y = y1;
    glBegin(GL_POINTS);
    glVertex2i(x, y);
    glEnd();
    for (k = 1 ; k <= step; k++)
    {
        x = x + Xin;
        y = y + Yin;
        glBegin(GL_POINTS);
        glVertex2i(x, y);glEnd();
    }
    glFlush();
}

void init(void)
{
    glClearColor(0.7, 0.7, 0.7, 0.7);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(-50, 100, -50, 100);
}
```

```
int main(int argc, char** argv)
{
    printf("Enter x1 and y1 : ");
    scanf("%f %f", &x1, &y1);
    printf("Enter x2 and y2 : ");
    scanf("%f %f", &x2, &y2);
    glutInit(&argc, argv);
    glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize (500, 500);
    glutInitWindowPosition (10, 10);
    glutCreateWindow ("Line Generation using DDA");
    init();
    glutDisplayFunc(display);
    glutMainLoop();
    return 0;
}
```