**DEERWALK INSTITUTE OF TECHNOLOGY**

**Tribhuvan University**

**Faculties of Computer Science**

****

**Bachelors of Science in Computer Science and Information Technology (BSc. CSIT)**

**Course: Computer Graphics (CSC214)**

**Year/Semester: II/III**

**A Lab report on:**

**Implementation of DDA and Graphics Functions**

Submitted by:

Name: Parth Poudyal

Roll: 1317

Submitted to:

Binod Sitaula

Department of Computer Science

**Theory:**

[1]DDA is a scan conversion method to digitalize the pixel data of a straight line of slope ; from initial points on frame buffer. This method involves of calculating y interval or the x interval for a straight line, which is

The slope of a line from initial points is given by:

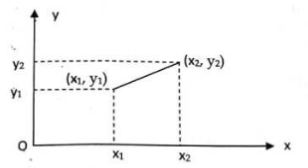


Figure : Line Path between two endpoints (x1,y1) and (x2,y2)

Similarly, for [[1]](#footnote-1):

When the i.e, the line progresses by unit in x direction in every iteration,

When the line progresses with slope lesser than equal to 1, provides the value of y in the next iteration.

Similarly, for different values of slope there are different cases of DDA, which provides us the coordinates for next points for the straight line to be scanned in the output buffer.

For values of slope:

For positive values of Slope:

From left to right:

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

From right to left:

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

For negative values of Slope:

Left to right

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

From right to left:

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

**Algorithm: DDA**

* STEP 1: Read
* STEP 2: Calculate and
* STEP 3: if i.e.
  + Steps =
* STEP 4: Otherwise:
  + Steps =

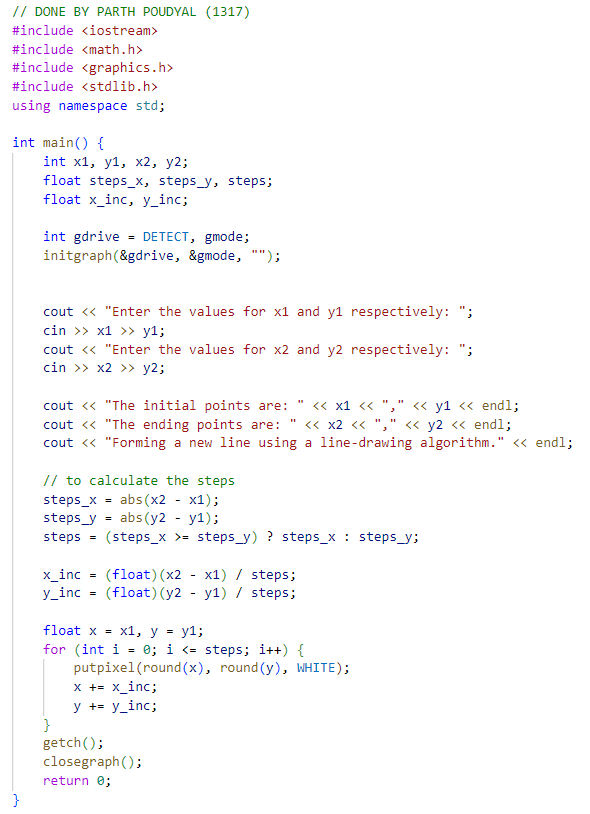
End

* STEP 5: Calculate
  + =
* STEP 6: For I = 1 to steps ; increment by 1
  + Plot //putpixel (x1,y1,RED)
  + x1 = x1 + xinc
  + y1 = y1 + yinc

End

* STEP 7: END

**Implementation of DDA in C program**

****

OUTPUT:

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

**Conclusion**

The DDA algorithm was successfully implemented to print a straight line in the output buffer. During this implementation it was observed that there are some limitations following this algorithm. Some of them were

* Constraints such as calculation floating point numbers while calculating a coordinate for a pixel (which is a quantified quantity) might bring out a staircase effect in the line when a line of long length printed using DDA
* The desired endpoint for the straight line might not be met due to inaccuracy brought by the floating-point calculations.

While the limitations cease to exist, DDA still proves to be simplest algorithm to print out a straight line in the graphics buffer.

# Bibliography

|  |  |
| --- | --- |
| [1] | B. S. S. I. C. Arjun Singh Saud, Computer Graphics, Kathmandu, Nepal: KEC, 2022. |

1. Here, represents the concurrent coordinate (pixel), while represents the next pixel in iterative algorithm to form a straight line. [↑](#footnote-ref-1)