Scan Conversion in Computer Graphics

INTRODUCTION

- **“Scan Conversion is te represention of regular objects in the form of discrete pixels,”** Every graphics system must transform the primitives into a collection of pixels. The scan conversion is also called **“Rasterization.”** Each pixel in the graphical system has two states either **on** or **off**.
- **“The Scan Converting rate is a technique that is used in video processing. In this technique, we can change the horizontal and vertical frequency for many different purposes.”** We use a scan converter to perform scan conversions. We can define a line by its starting and ending points. On the other hand, we can define a circle by its radius, circle equation, and center point.

Methods of Scan Conversion

- We can perform scan conversion by using two methods.
- **Analog Method**
- **Digital Method**
- **Analog Method:** It is the best method for analog videos. The Analog approach is also known as **"Non-retentive, memory-less, or real-time method."**
- **Digital Method:** Digital method is also known as a **"retentive or buffered method."** In the digital method, there is a concept of $\mathbf{n_1}$and $\mathbf{n_2}$ speed. We can save (Store) the picture in line or frame buffer with n_1 speed. The image can be read with n_2 speed.

Advantages of Scan Conversion

- The scan conversion technique is used for various display units, such as videoProjectors, TV, HDTV, Video capture card, LCD monitor, etc.
- The Scan Conversion technique has wide applications.
- We can efficiently perform scan conversion using high speed integrated circuits.

Disadvantages of Scan Conversion

- We can only apply scan conversion with LSI and VLSI integrated circuit.
- In the digital scan conversion, the analog video signal has changed into digital data.

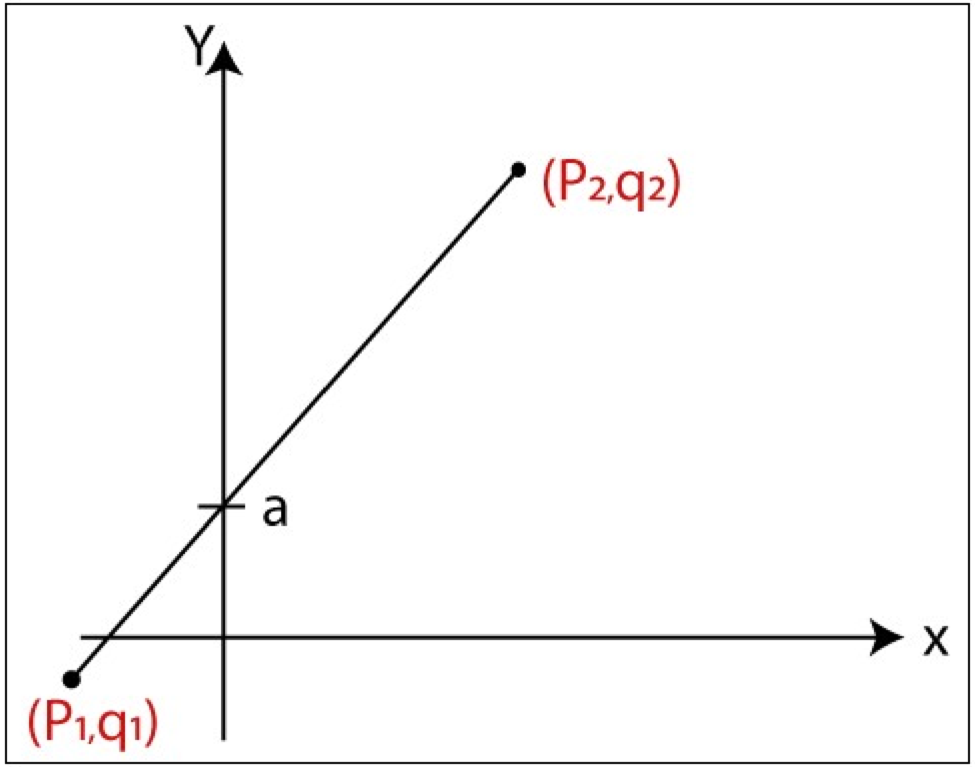
Line Drawing Algorithm in Computer Graphics

Properties of a Line Drawing Algorithm

- There are the following properties of a good Line Drawing Algorithm.
- **An algorithm should be precise:** Each step of the algorithm must be adequately defined.
- **Finiteness:** An algorithm must contain finiteness. It means the algorithm stops after the execution of all steps.
- **Easy to understand:** An algorithm must help learners to understand the solution in a more natural way.
- **Correctness:** An algorithm must be in the correct manner.
- **Effectiveness:** The steps of an algorithm must be valid and efficient.
- **Uniqueness:** All steps of an algorithm should be clearly and uniquely defined, and the result should be based on the given input.
- **Input:** A good algorithm must accept at least one or more input.
- **Output:** An algorithm must generate at least one output.

Equation of the straight line

- We can define a straight line with the help of the following equation.
- **y= mx + a**
  - Where,
  - **(x, y**) = axis of the line.
  - **m** = Slope of the line.
    - **a =** Interception point


Y
(P2,q2)
a
x
(P1,q1)

Algorithms of Line Drawing

- There are following algorithms used for drawing a line:
- **DDA (Digital Differential Analyzer) Line Drawing Algorithm**
- **Bresenham's Line Drawing Algorithm**
- **Mid-Point Line Drawing Algorithm**
-

DDA (Digital Differential Analyzer) Line Drawing Algorithm

- The Digital Differential Analyzer helps us to interpolate the variables on an interval from one point to another point. We can use the digital Differential Analyzer algorithm to perform rasterization on polygons, lines, and triangles.
- Digital Differential Analyzer algorithm is also known as an **incremental method** of scan conversion. In this algorithm, we can perform the calculation in a step by step manner using the line equation.

As we know the general equation of the straight line is: $y = mx + c$

Here, **m** is the slope of (x_1, y_1) and (x_2, y_2). $m = (y_2 – y_1)/ (x_2 – x_1)$

Now, we consider one point (x_k, y_k) and (x_{k+1}, y_{k+1}) as the next point.

Then the slope $m = (y_{k+1} – y_k)/ (x_{k+1} – x_k)$

There can be following three cases to discuss:

Case 1: If **m < 1**

Then **x** coordinate tends to the Unit interval. $x_{k+1} = x_k + 1$

$y_{k+1} = y_k + m$

Case 2: If **m > 1**

Then **y** coordinate tends to the Unit interval. $y_{k+1} = y_k + 1$

$x_{k+1} = x_k + 1/m$

Case 3: If **m = 1**

Algorithm of Digital Differential Analyzer (DDA) Line Drawing

- **Step 1:** Start.
- **Step 2:** We consider Starting point as **(x_1, y_1),** and ending point **(x_2, y_2).**
- **Step 3:** Now, we have to calculate **?x** and **?y.** **?x = x_2-x_1**
- **?y = y_2-y_1**
- **m = ?y/?x**
- **Step 4:** Now, we calculate three cases.
- If **m < 1**
- Then **x** change in Unit Interval, **y** moves with deviation
- **(x_{k+1}, y_{k+1}) = (x_k+1, y_k+m)**
- If **m > 1**
- Then **x** moves with deviation, **y** change in Unit Interval
- **(x_{k+1}, y_{k+1}) = (x_k+1/m, y_k+1)**
- If **m = 1**
- Then **x** moves in Unit Interval, **y** moves in Unit Interval
- **(x_{k+1}, y_{k+1}) = (x_k+1, y_k+1)**
- **Step 5:** We will repeat step 4 until we find the ending point of the line.
- **Step 6:** Stop.

Example: A line has a starting point (1,7) and ending point (11,17). Apply the Digital Differential Analyzer algorithm to plot a line.

Solution: We have two coordinates,

Starting Point = **(x_1, y_1)** = **(1,7)**

Ending Point = **(x_2, y_2)** = **(11,17)**

Step 1: First, we calculate **?x**, **?y** and **m.**

?x = $x_2 – x_1$ = 11-1 = 10

?y = $y_2 – y_1$ = 17-7 = 10

m = ?y/?x = 10/10 = 1

Step 2: Now, we calculate the number of steps.

?x = ?y = 10

Then, the number of steps = **10**

Step 3: We get **m = 1,** Third case is satisfied.

Now move to next step.

The coordinates of drawn line are-

$P_1 = (2, 8)$

$P_2 = (3, 9)$

$P_3 = (4, 10)$

$P_4 = (5, 11)$

$P_5 = (6, 12)$

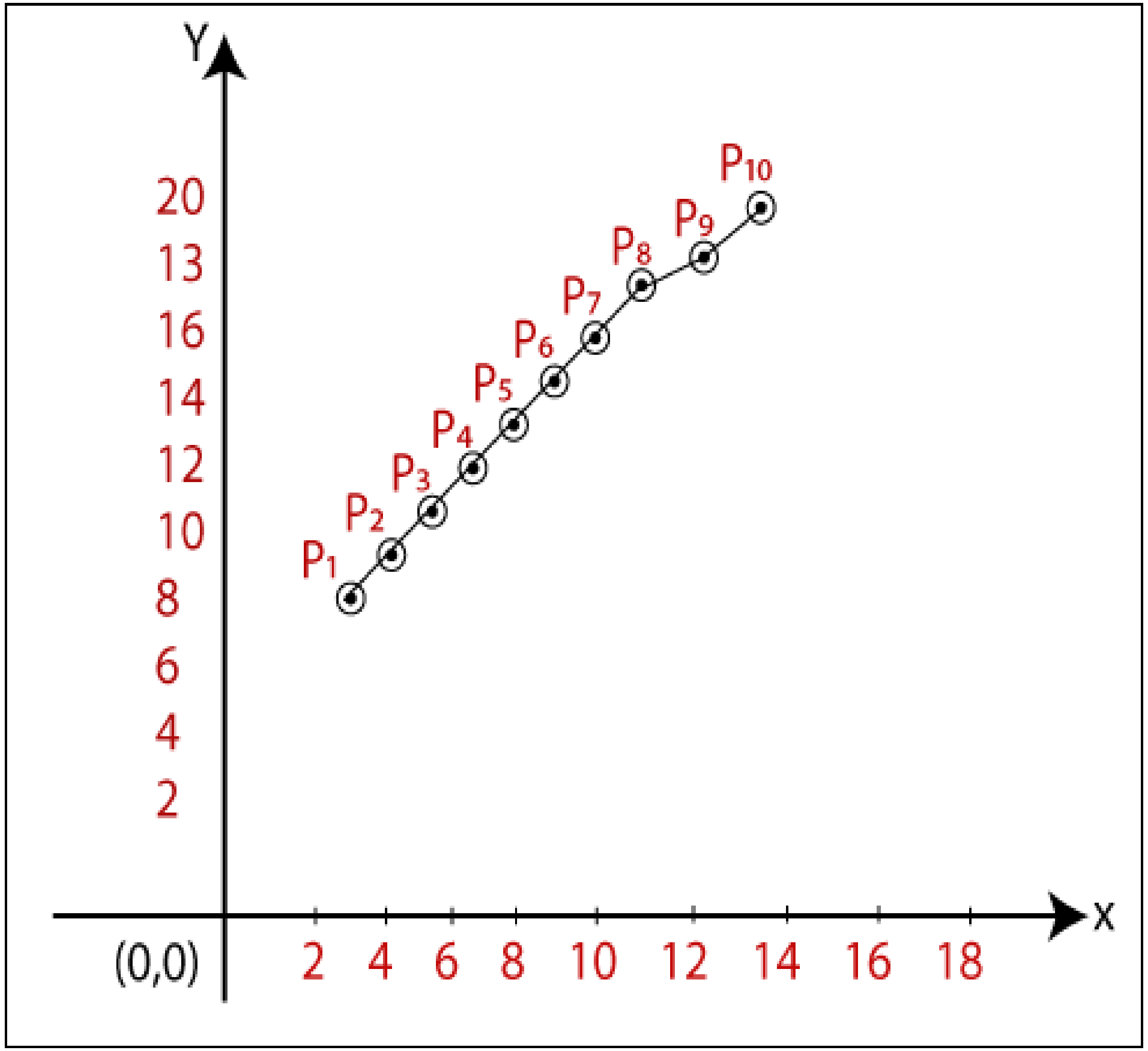
$P_6 = (7, 13)$

$P_7 = (8, 14)$

$P_8 = (9, 15)$

$P_9 = (10, 16)$

$P_{10} = (11, 17)$


Y
20
13
16
14
12
10
8
6
4
2
(0,0)
2
4
6
8
10
12
14
16
18
X
P_1
P_2
P_3
P_4
P_5
P_6
P_7
P_8
P_9
P_{10}

Advantages of Digital Differential Analyzer

It is a simple algorithm to implement.

It is a faster algorithm than the direct line equation.

We cannot use the multiplication method in Digital Differential Analyzer.

Digital Differential Analyzer algorithm tells us about the overflow of the point when the point changes its location.

Disadvantages of Digital Differential Analyzer

The floating-point arithmetic implementation of the Digital Differential Analyzer is time-consuming.

The method of round-off is also time-consuming.

Sometimes the point position is not accurate.

Bresenham’s Line Drawing Algorithm in Computer Graphics

- This algorithm was introduced by **“Jack Elton Bresenham”** in **1962.** This algorithm helps us to perform scan conversion of a line. It is a powerful, useful, and accurate method. We use incremental integer calculations to draw a line. The integer calculations include addition, subtraction, and multiplication.

Step 1: Start.

Step 2: Now, we consider Starting point as **(x_1, y_1)** and endingpoint **(x_2, y_2).**

Step 3: Now, we have to calculate **?x** and **?y.**

$$?x = x_2 - x_1$$

$$?y = y_2 - y_1$$

$$m = ?y/?x$$

Step 4: Now, we will calculate the decision parameter **p_k** with following formula.

$$p_k = 2?y - ?x$$

Step 5: Theinitial coordinates of the line are **(x_k, y_k),** and the next coordinatesare **(x_{k+1}, y_{k+1}).** Now, we are going to calculate two cases for decision parameter **p_k**

Case 1: If

$p_k < 0$

Then

$p_{k+1} = p_k + 2?y$

$x_{k+1} = x_k + 1$

$y_{k+1} = y_k$

Case 2: If

$p_k >= 0$

Then

$p_{k+1} = p_k + 2?y - 2?x$

$x_{k+1} = x_k + 1$

$y_{k+1} = y_k + 1$

Step 6: We will repeat step 5 until we found the ending point of the line and the total number of iterations =**?x-1.**

Step 7: Stop.

Example: A line has a starting point (9,18) and ending point (14,22). Apply the Bresenham's Line Drawing algorithm to plot a line.

Solution: We have two coordinates,

Starting Point = **(x_1, y_1)** = **(9,18)**

Ending Point = **(x_2, y_2)** = **(14,22)**

Step 1: First, we calculate **?x**, **?y.**

?x = $x_2 - x_1$ = 14-9 = 5

?y = $y_2 - y_1$ = 22-18 = 4

Step 2: Now, we are going to calculate the decision parameter **(p_k)**

p_k = 2?y-?x

= 2 x 4 – 5 = 3

The value of **p_k = 3**

Step 3: Now, we will check both the cases.

If

$p_k >= 0$

Then

Case 2 is satisfied. Thus

$p_{k+1} = p_k +2?y-2?x =3+ (2 x 4) – (2 x 5) = 1$

$x_{k+1} =x_k +1 = 9 + 1 = 10$

$y_{k+1} =y_k +1 = 18 +1 = 19$

Step 4: Now move to next step. We will calculate the coordinates until we reach the end point of the line.

?x -1 = 5 – 1 = 4

Advantages of Bresenham’s Line Drawing Algorithm

It is simple to implement because it only contains integers.

It is quick and incremental

It is fast to apply but not faster than the Digital Differential Analyzer (DDA) algorithm.

The pointing accuracy is higher than the DDA algorithm.

Disadvantages of Bresenham’s Line Drawing Algorithm

The *Bresenham’s Line drawing algorithm* only helps to draw the basic line.

The resulted draw line is not smooth.

Algorithm of Mid-Point Subdivision Line Drawing

Step 1: Start.

Step 2: Consider the starting point as **(x_1, y_1)** and ending point as **(x_2, y_2).**

Step 3: Now, we will calculate **?d.**

?d = **2 (?y -?x)**

Step 4: Now, we will calculate the decision parameter **d_i** with the following formula.

d_i = **2?y – ?x**

Step 5: The increment of x or y coordinate depends on the following two cases-

$d_i < 0$

Then

$x_{k+1} = x_k + 1$

$y_{k+1} = y_k$

$d_n = d_i + 2?y$

Case 2: If

$d_i >= 0$

Then

$x_{k+1} = x_k + 1$

$y_{k+1} = y_k + 1$

$d_n = d_i + ?d$

Step 6: We will repeat step 5 until we found the ending point of the line.

Step 7: Stop.

Step 3: Now, we will check both the cases.

If

$p_k >= 0$

Then

Case 2 is satisfied. Thus

$p_{k+1} = p_k + 2?y - 2?x = 3 + {}_{(}2 \text{ x } 4) – (2 \text{ x } 5) = 1$

$x_{k+1} = x_k + 1 = 9 + 1 = 10$

$y_{k+1} = y_k + 1 = 18 + 1 = 19$

Step 4: Now move to next step. We will calculate the coordinates until we reach the end point of the line.

?x -1 = 5 – 1 = 4

Example: A line has a starting point (6,10) and ending point (13,17). Apply the Mid-point Line Drawing algorithm to draw a line.

Solution: We have two coordinates,

Starting Point = $\mathbf{(x_1, y_1)}$ = **(6,10)**

Ending Point = $\mathbf{(x_2, y_2)}$ = **(13,17)**

Step 1: First, we calculate **?x**, **?y.**

$$?x = x_2 – x_1 = 13-6 = 7$$

$$?y = y_2 – y_1 = 17-10 = 7$$

Step 2: Now, we are going to calculate the **?d.**

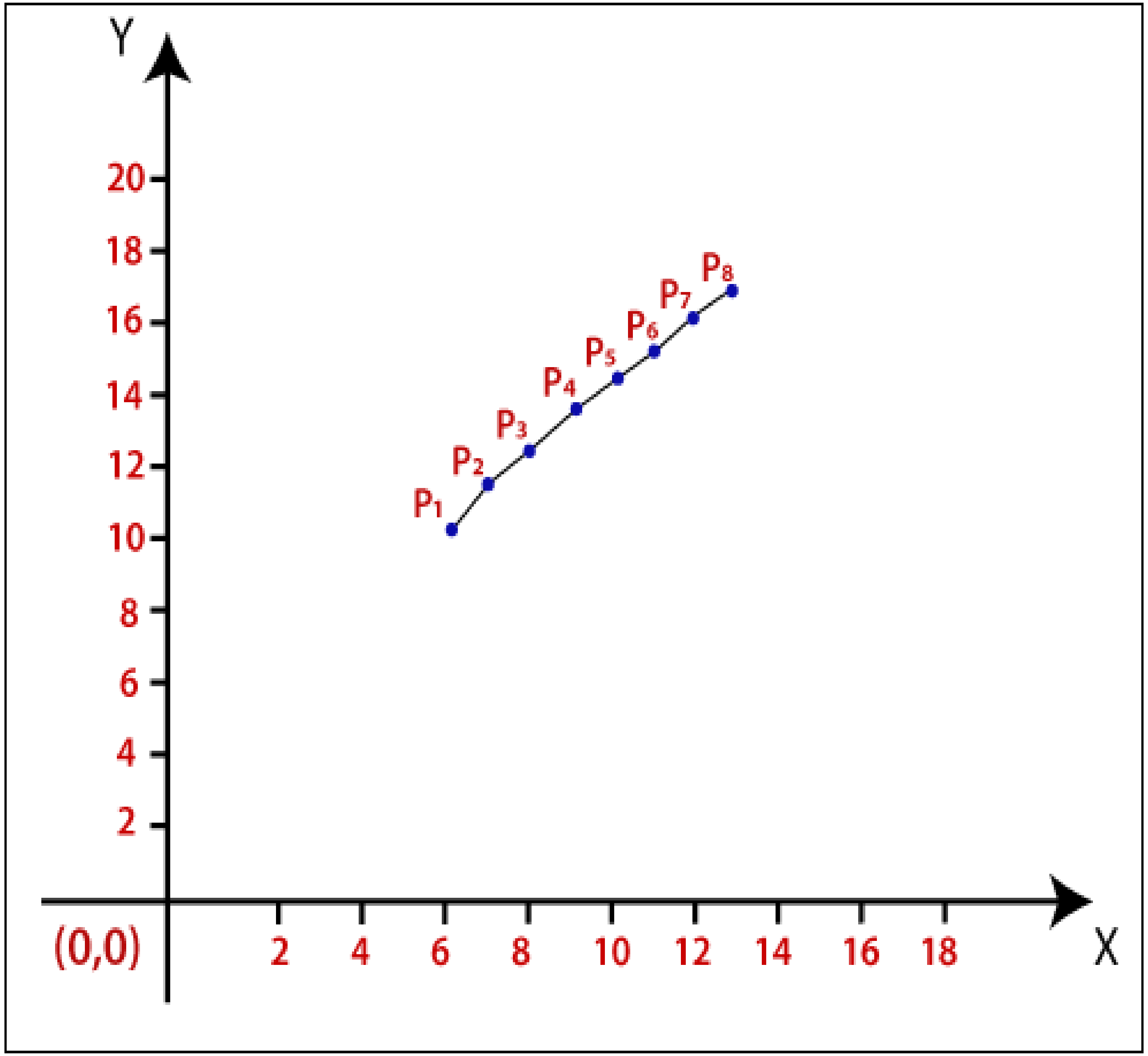
$$?d= 2(?y-?x)$$

$$= 2 (7 – 7) = 0$$

The value of **?d= 0**

Step 3: Now, we will calculate the decision parameter $\mathbf{d_i}$ with following formula.

$$d_i = 2?y – ?x$$


Y
20
18
16
14
12
10
8
6
4
2
(0,0)
2
4
6
8
10
12
14
16
18
X
P1
P2
P3
P4
P5
P6
P7
P8

Advantages of Mid-point Subdivision Line Drawing Algorithm

It is very easy to implement.

It is a less time-consuming algorithm.

This algorithm uses basic arithmetic operations.

It only requires integer data.

The drawn line is smooth than other line drawing algorithms.

Disadvantages of Mid-point Subdivision Algorithm

Sometimes this algorithm is not suitable due to critical images and graphics.

The points have less accuracy. There are improvements needed.

ASSIGNMENT ONE

Using C++ and an IDE of your own choice, illustrate the configuration of C++ to work with OpenGL and the chosen IDE
Write suitable C++ and openGl code to create a window (800 x 800) cleared with a color of your own choice.
Write suitable code to draw
1) Line with default color and thickness
2) Dotted line with a color and thickness of your own choice.
NB: provide screen shot showing the code and the output in each case.
Due date 29/6/2021