Computer Graphics

Lecture-00

Md Imtiaz Ahmed

Lecturer,DIIT

Primary Focus

Develop thinking ability.

- problem solving skills.
- formal thinking.

Goals

- Be very familiar with a collection of ***Computer Graphics algorithms***.
- Be intimately familiar with mathematics for Computer Graphics
- Be able to ***apply*** techniques in practical problems.

Textbook & References

- *Schaum's ouTlines series, THEORY AND PROBLEM OF COMPUTER GRAPHICS ,Second Edition,* McGraw Hill.

Computer Graphics
Chapter 3
Scan Conversion
Lecture-01

Md Imtiaz Ahmed

Lecturer,DIIT

Scan Conversion

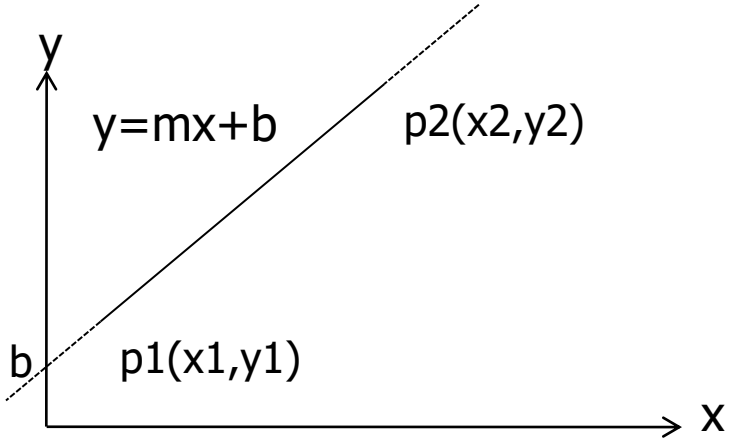
- Graphics System convert
  - each primitive(points ,lines, circles, ellipses etc.) from its geometric definition into a set of pixels that make up the picture in the image space.
  - This Conversion task is referred to as Scan Conversion.

Scan Converting a point

- (x,y) where x and y are real numbers within an image area.
- Need to be scan converted at pixel location (x', y')

Scan Converting a line

- Line drawing is accomplished by calculating intermediate positions along the line path between two specified end points.
- It is defined by the line equation y=mx+b, m is slope and b the y intercept of the line.
- Two end points are p1(x1,y1) and p2(x2,y2)
- Line equation describes the coordinates of all the points that lie between the two end points.



Direct use of the Line Equation

- Scan convert p1 and p2 to pixel coordinates (x'_1, y'_1) and (x'_2, y'_2) then set $m = (y'_2 - y'_1)/(x'_2 - x'_1)$ and $b = y'_1 - mx'_1$.
- *If* $|m| \leq 1$, then for every integer value of x between and excluding $x'_1 and x'_2$, calculate the corresponding value of y using the equation and scan convert (x,y).

- *If* $|m| \geq 1$, then for every integer value of x between and excluding $y'_1 and y'_2$, calculate the corresponding value of x using the equation and scan convert (x,y).

DDA Algorithm

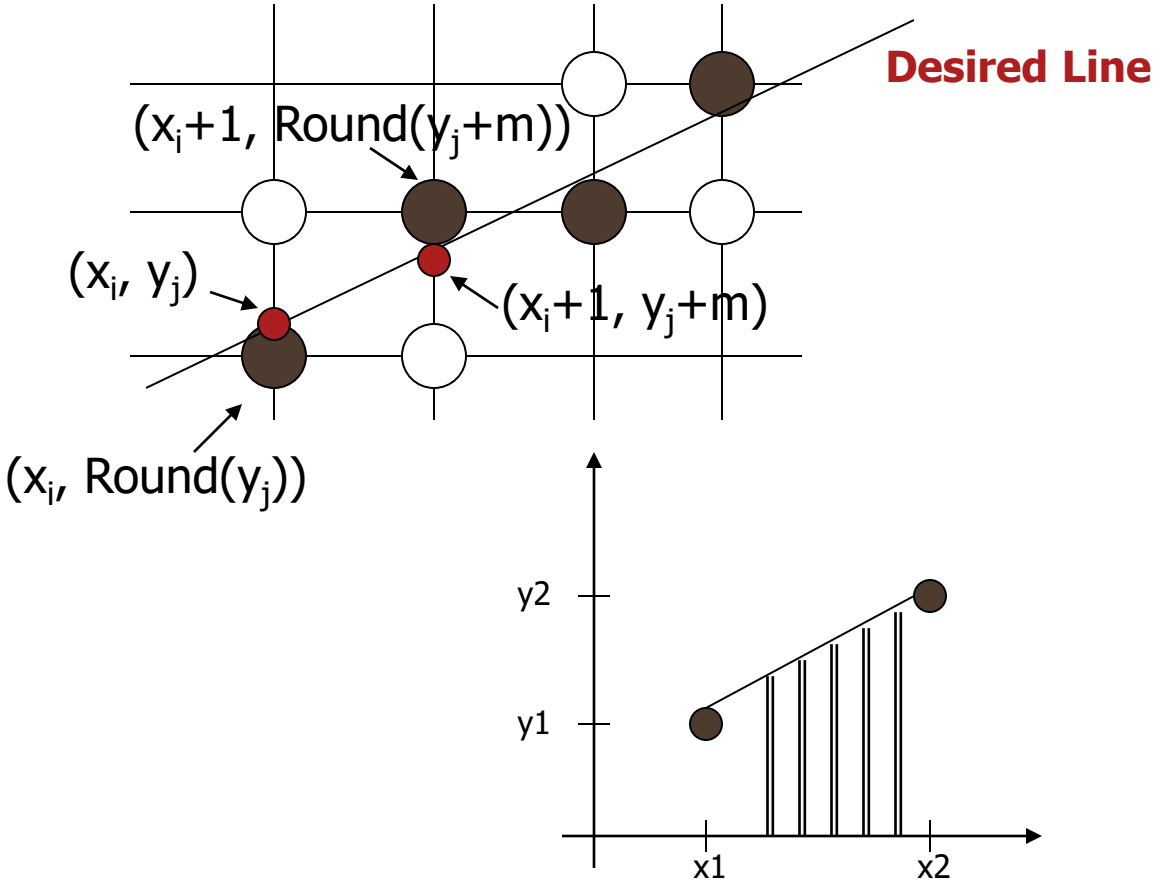
- *The digital differential analyzer (DDA)* is an *incremental scan-conversion method.*
- *Characterized by Performing calculations at each step using results from the preceding step.*
- *At each step i calculate (xi,yi) point on the line.*
- *Next point (xi+1,yi+1)*
- $\Delta y / \Delta x = m, where \Delta y = y_{i+1} - y_i and \Delta x = x_{i+1} - x_i, We\ have$

$$y_{i+1} = y_i + m\Delta x$$

$$or$$

$$x_{i+1} = x_i + \Delta y / m$$

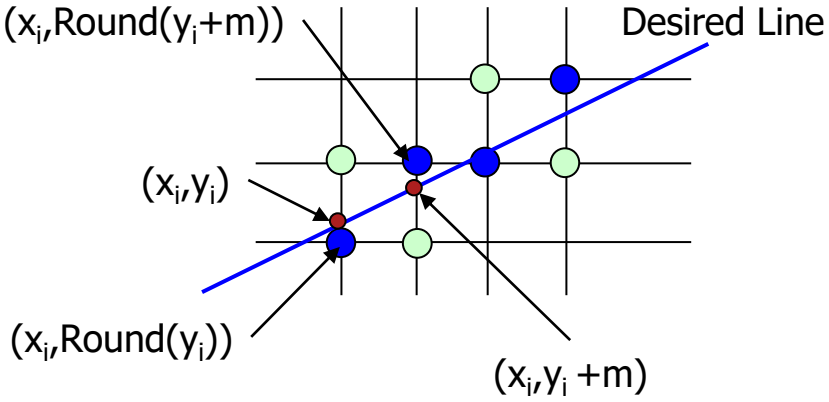
DDA Algorithm



DDA Algorithm

- DDA ("Digital Differential Analyzer")
- Assume $0 \le m \le 1$.

$$
\begin{aligned}
y_{i+1} &= mx_{i+1} + B \\
&= m(x_i + \Delta x) + B \\
&= y_i + m\Delta x \\
y_{i+1} &= y_i + m \quad [\Delta x = 1]
\end{aligned}
$$




```
void Line(int x0, int y0, int x1, int y1, int value) {
    double y = y0;
    double m = (y1 – y0) / (x1 – x0);      // 0 <= m <= 1
    for (int x = x0; x <= x1; x++) {
        WritePixel(x, Round(y), value);
        y += m;
    }
}
```

Require to eliminate floating point operations & variables

DDA ALGORITHM

- *DDA* algorithm is faster than the direct use of the line equation since it calculates points on the line without any floating point multiplication
- Cumulative error occurs.