CSE- 4105

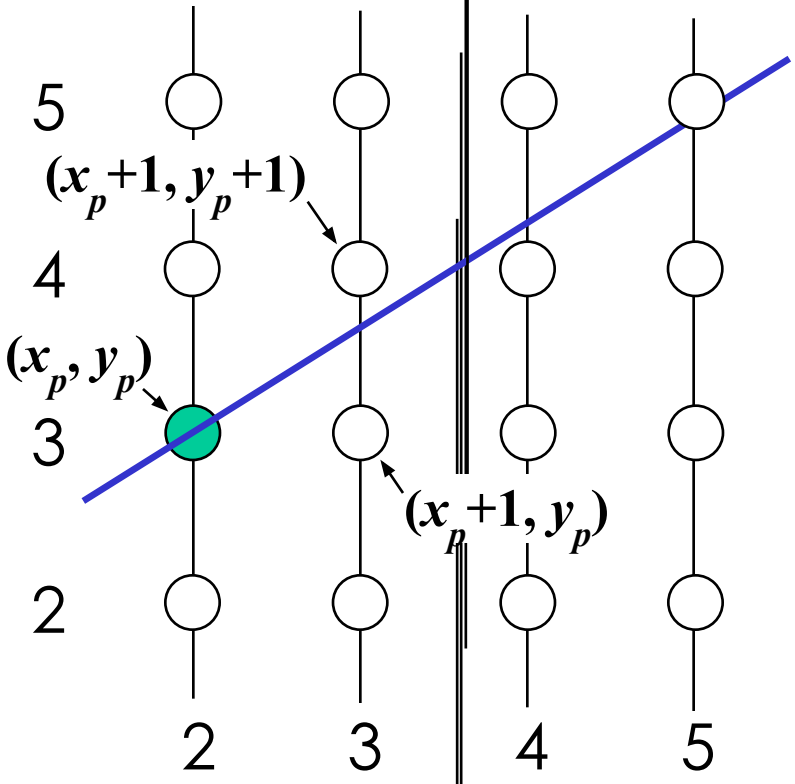
Lecturer- 04

Line Drawing Algorithms

- Bresenham's line drawing
  - Efficient line drawing algorithm using only incremental integer calculations
  - Can be adapted to draw circles and other curves
- Principle
  - Vertical axes show scan line positions
  - Horizontal axes show pixel columns
  - At each step, determine the best next pixel based on the sign of an integer parameter whose value is proportional to the difference between the vertical separations of the two pixel positions from the actual line.

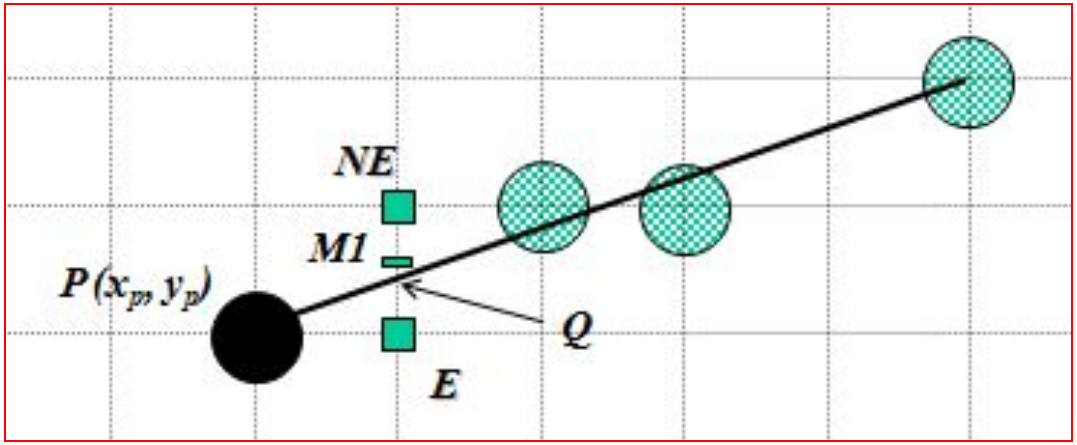
Big idea.

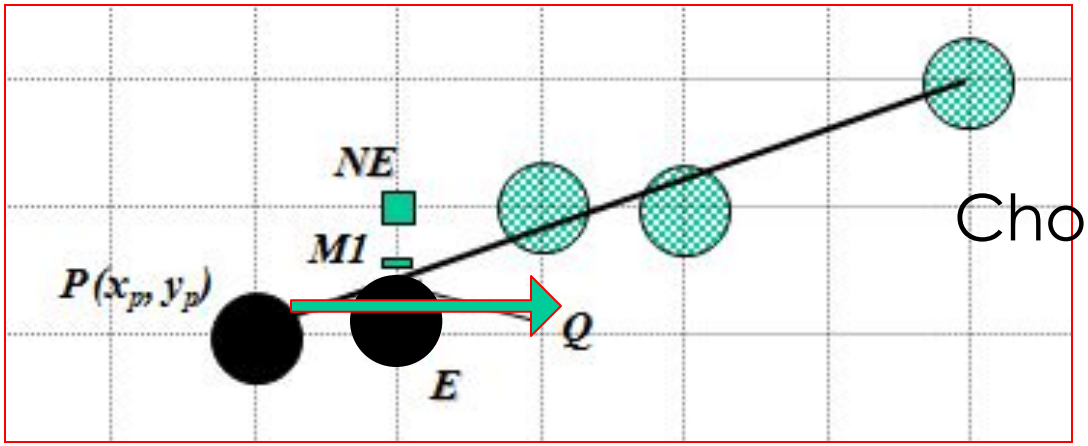
- Move across the x axis in unit intervals and at each step choose between two different y coordinates
- For example, from position (2, 3) we have to choose between (3, 3) and (3, 4)



Choosing a pixel

- We can move either to East (E)or North east (NE)

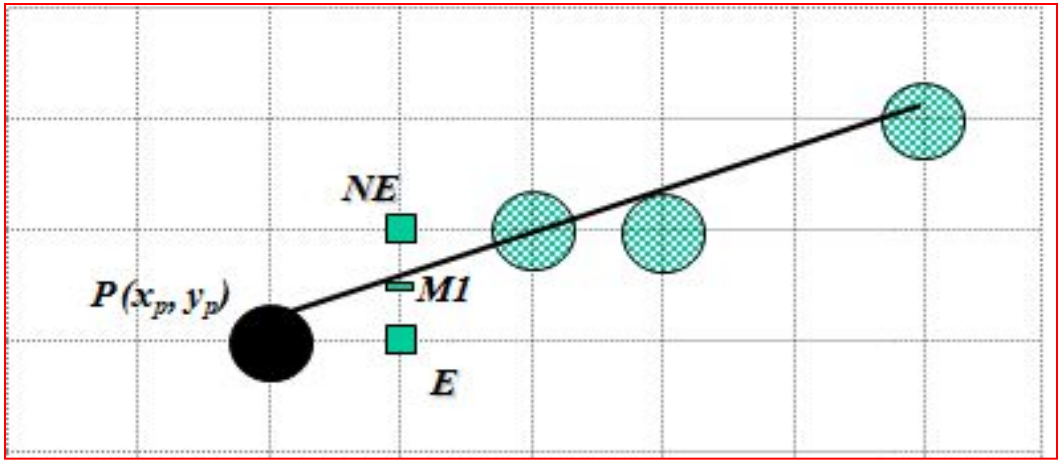


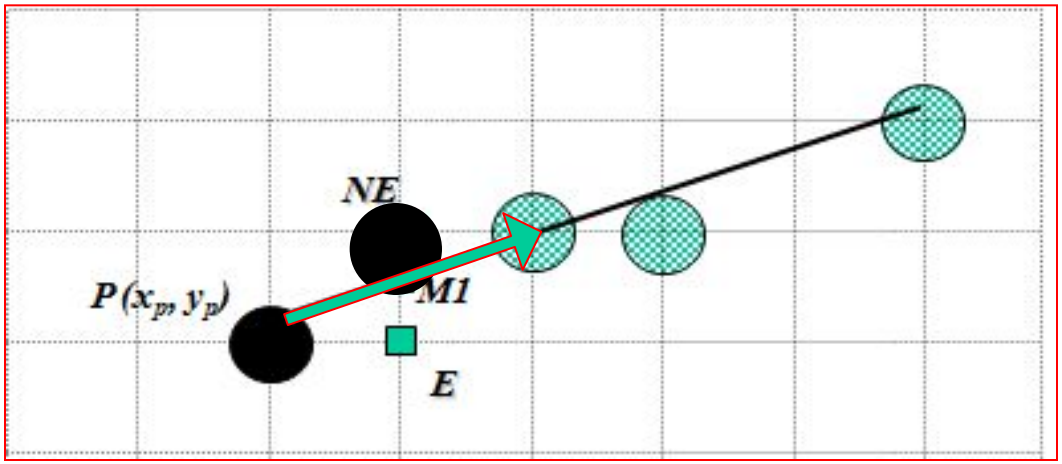


Choose E

Choosing a pixel

- We can move either to East (E)or North east (NE)

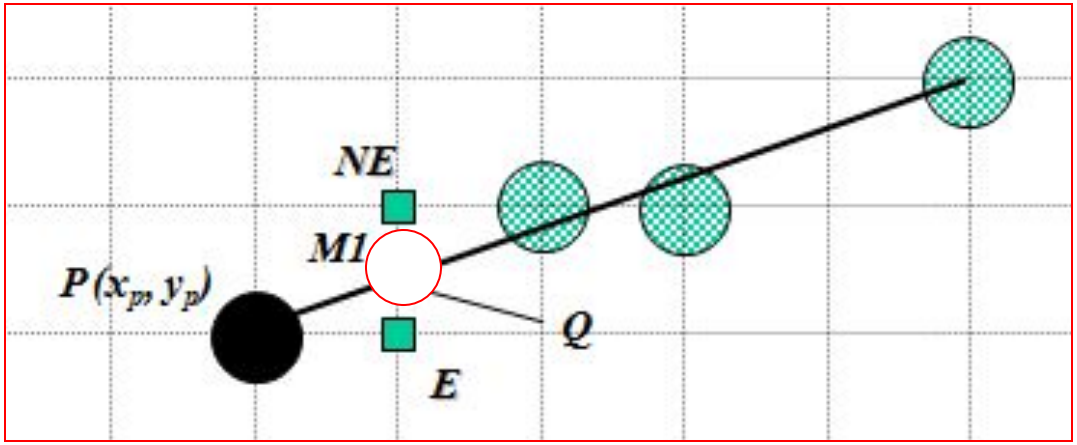




Choose NE

Choosing a pixel

- We can move either to East (E)or North east (NE)



Option depends on the midpoint M1

Equation

$$y = mx + c$$

$$y = \frac{dy}{dx}x + c$$

$$ydx = xdy + cdx$$

$$xdy - ydx + cdx = 0$$

$$A = dy$$

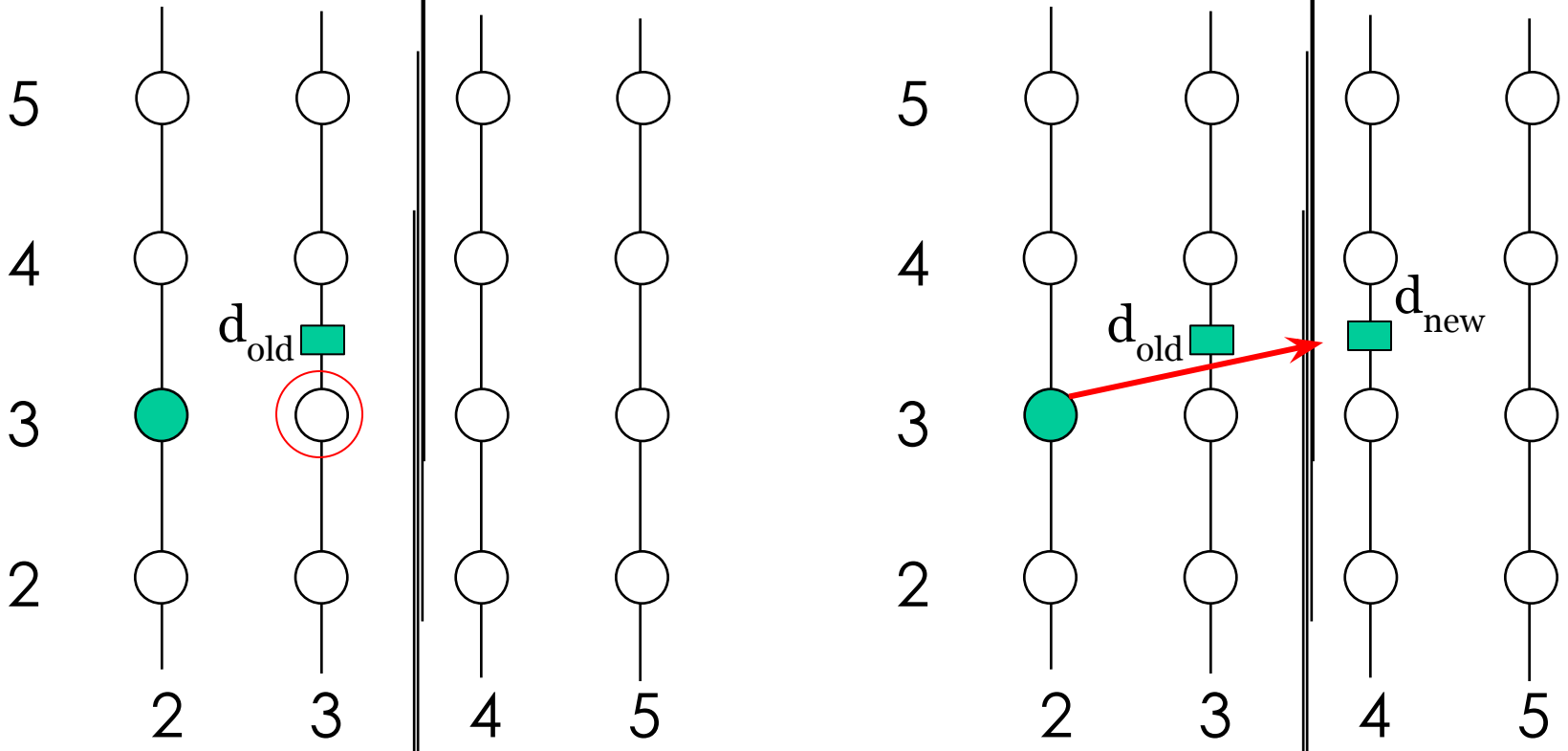
$$B = -dx$$

$$C = cdx$$

$$Ax + By + C = 0$$

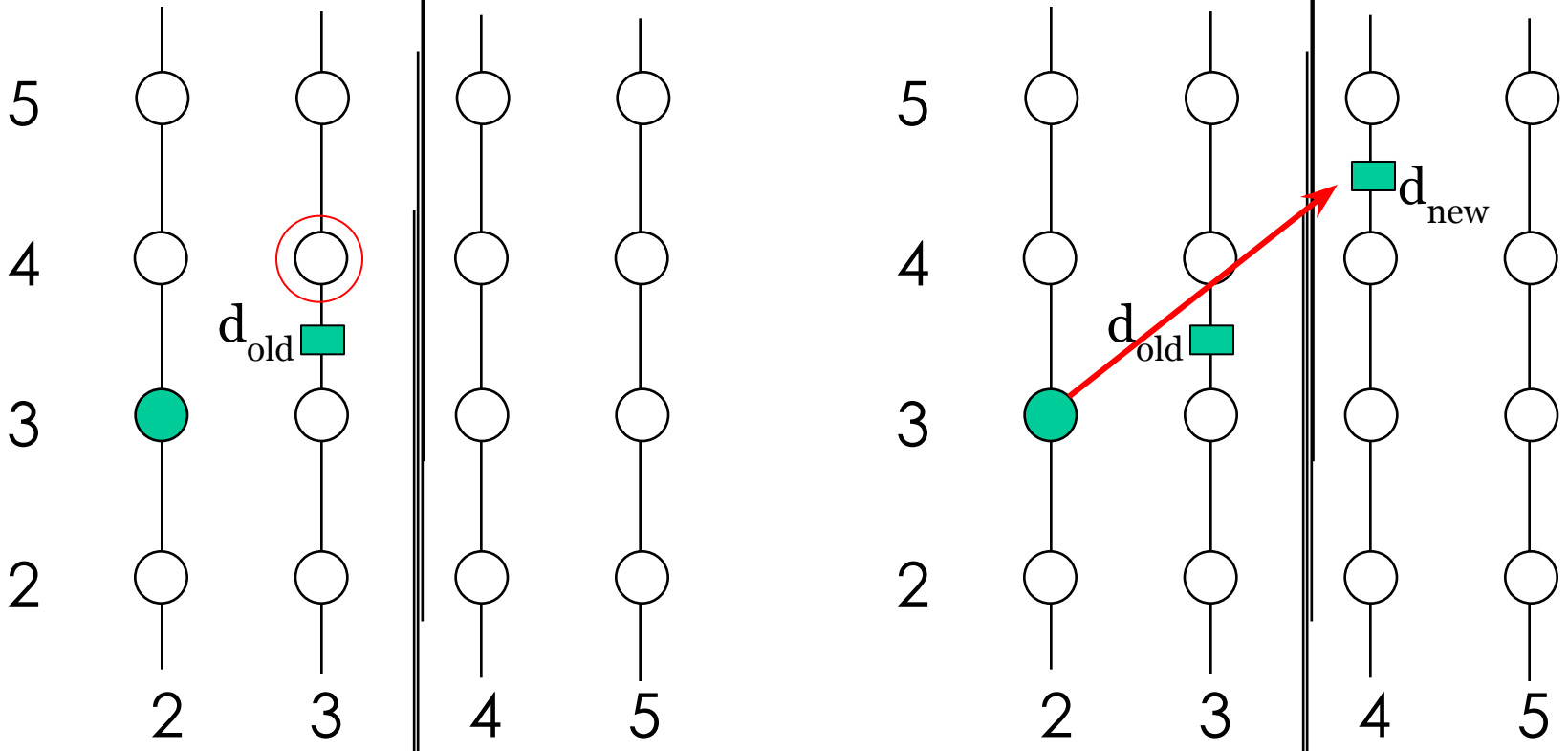
Big idea.

- If we choose E then we have to calculate the next mid point *dnew*

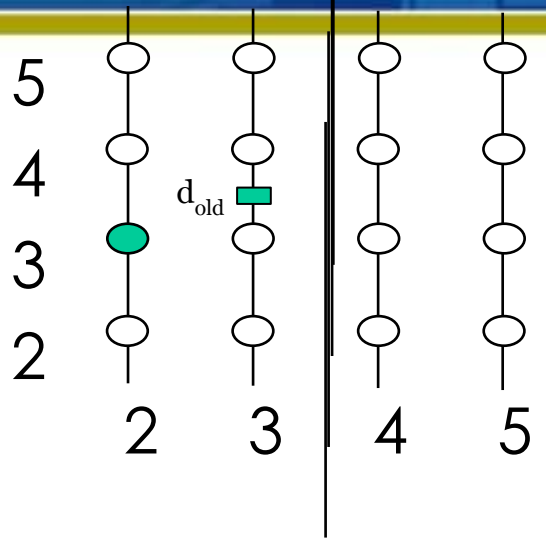


Big idea.

- If we choose NE then we have to calculate the next mid point *dnew*



- We have to calculate the decision variable d
- If d<0 , E is chosen
- If d>0, NE is chosen
- D=F $(x_p+1, y_p+1/2)$
  $= A(x_p+1)+B(y_p+1/2)+C$



How to update d :

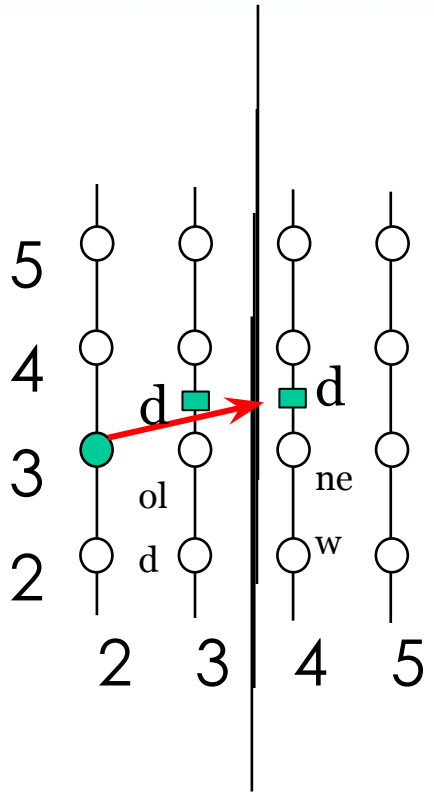
- On the basis of picking E or NE, figure out the location of M for that pixel, and the corresponding value of d for the next grid line.
- If E is chosen:
  - M is incremented by one step in the x direction

$$d_{new} = F(x_p + 2, y_p + ½)$$
$$= A(x_p + 2) + B(y_p + ½) + c$$

$$d_{old} = A(x_p + 1) + B(y_p + ½) + c$$

  - incremental difference $\mathbf{\Delta E = d_{new} - d_{old}}$

$$\Delta E = A = dy$$



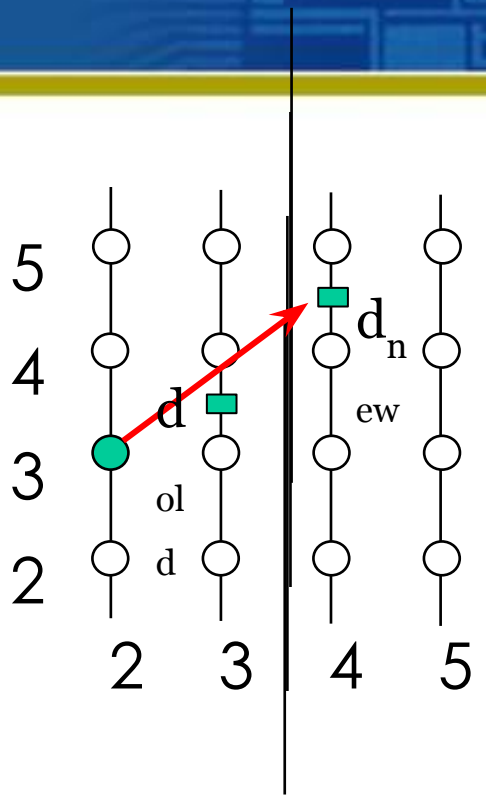
How to update d :

- If NE is chosen:
  - M is incremented by one step each in both the x and y directions

$$d_{new} = F(x_p + 2, y_p + 3/2)$$

$$= A(x_p + 2) + B(y_p + 3/2) + c.$$

$$d_{old} = A(x_p + 1) + B(y_p + ½) + c$$



- incremental difference $\mathbf{\Delta NE = d_{new} - d_{old}}$

$$\Delta NE = A + B = dy - dx.$$

Initial value of d_0

- Initial value of *d*

$$\begin{aligned} d_0 &= F(x_0 + 1, y_0 + 0.5) = a(x_0 + 1) + b(y_0 + 0.5) + c \\ &= F(x_0, y_0) + a + 0.5b \\ &= a + 0.5b \end{aligned}$$

Algorithm

```
DrawLine(int x1, int y1,
         int x2, int y2, int color)
{
    int dx, dy, d, incE, incNE, x, y;
    dx = x2 - x1;
    dy = y2 - y1;
    d   = 2*dy - dx;
    incE   = 2*dy;
    incNE = 2*(dy - dx);
    y = y1;

    for (x=x1; x<=x2; x++)
    {
        WritePixel(x, y, color);
        if (d>0) {
            d = d + incNE;
            y = y + 1;
        } else {
            d = d + incE;
        }
    }
}
```

Thank You