Topics to be Covered

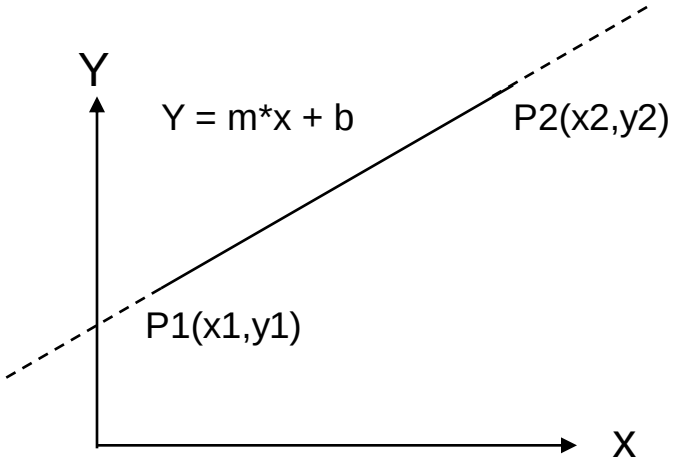
- Scan Conversion –Definition
- Line Algorithms
  - Slope-Intercept Method
  - Bresenhams
  - DDA
- Circle Drawing Algorithms
- Ellipse Drawing Algorithm
- Area Filling Techniques
- Character Generation

Scan Conversion

- Converting the geometric definition of a primitive form into a set of pixels that make up the primitive in the image space. This conversion task is generally referred to as scan conversion or rasterization

Scan Converting a Line

- Mathematical Representation of a line
  - A line segment is defined by its two endpoints and the line equation
    - Y = mx + b, where m is the slope and b is the intercept



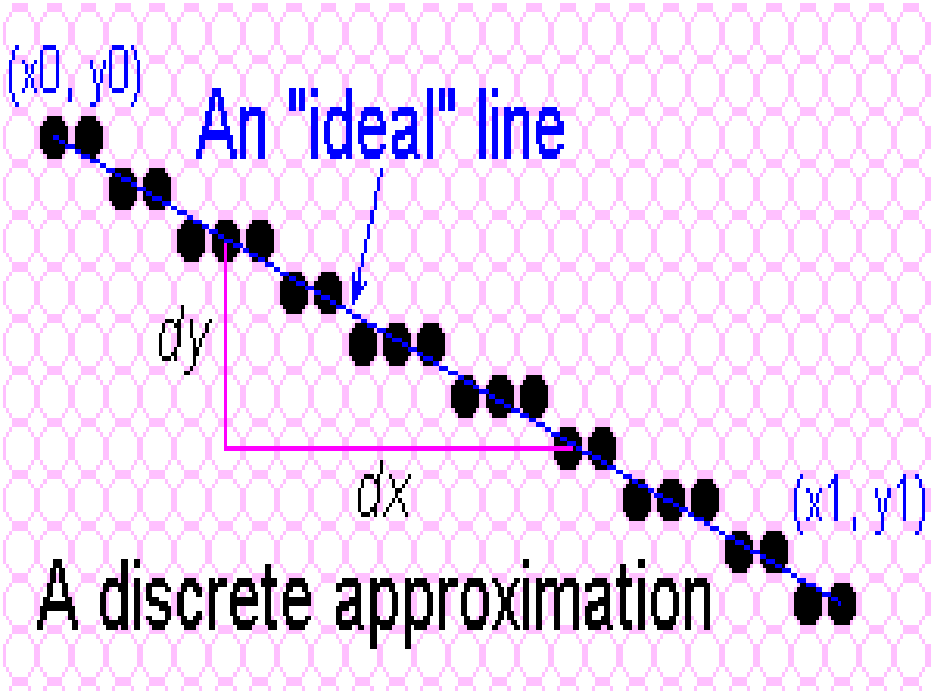
Defining a Line

Line Drawing Algorithms

- Slope-Intercept method
- Digital Differential (DDA) Algorithm
- Bresenham's Line Algorithm

Quest for the *Ideal Line*

The best we can do is a discrete approximation of an ideal line.



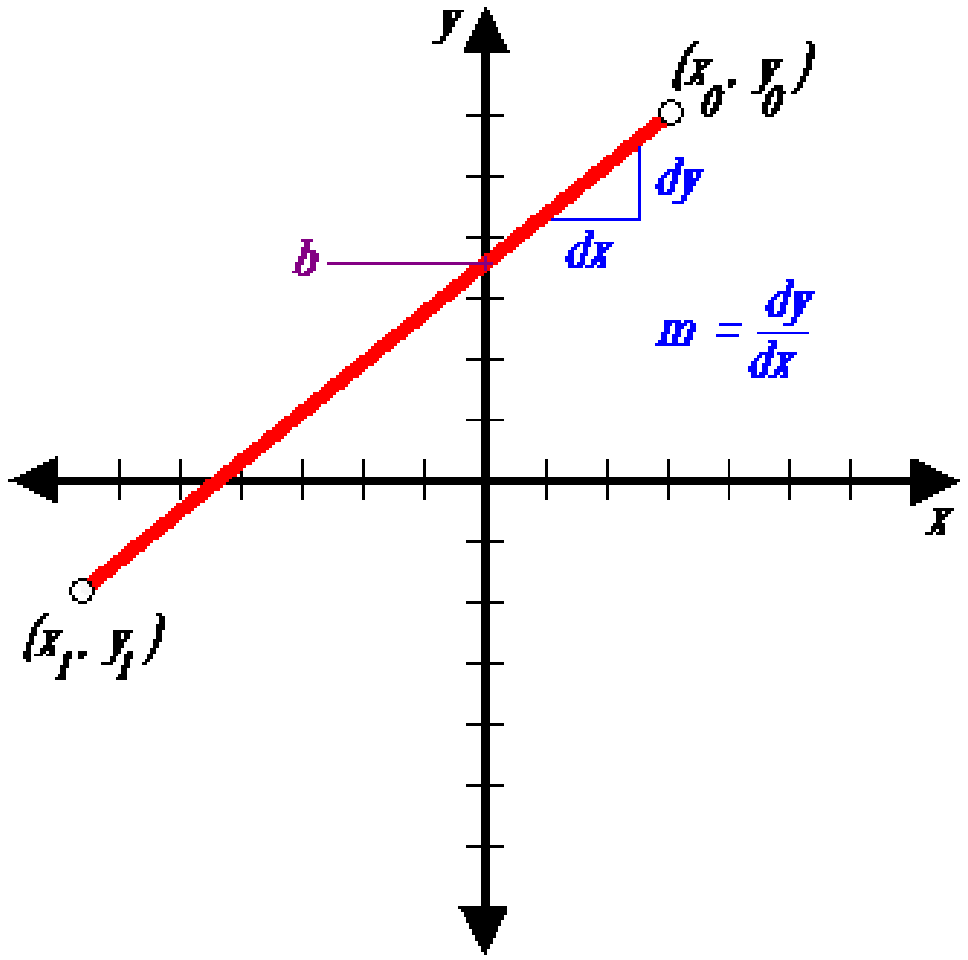
Important line qualities:

- Continuous appearance
- Uniform thickness and brightness
- Accuracy (Turn on the pixels nearest the ideal line)
- Speed (How fast is the line generated)

Simple Line

Based on the simple *slope-intercept algorithm* from algebra

$$y = m x + b$$



An algorithm to draw a line using the slope-Intercept methods (slope is between 0^o to 45^o)

1. Compute
   dx = x2-x1
   dy = y2-y1
   m = dy/dx
   b = y1-m*x1
2. If dx < 0 x = x2, y = y2 and xend =x1
3. if dx > 0 x=x1, y=y1 and xend = x2
4. If x< xend, stop
5. Plot a point at (x, y)
   x = x+1
   Y=mx+b
6. Go to step 4

Direct Use Line Equation

- Involves floating point computation (multiplication and addition) at every step leading to increase in the computation time

DDA

- Incremental scan-conversion method
- Faster than the direct use of the line equation
- However, a floating point operation is still required
- The line drifts away from the original line when the line is relatively long.

An algorithm to draw a line using the DDA method

1. Compute

   dx = x2-x1 dy = y2-y1

2. If abs(dx) > abs(dy) then steps = abs(dx)
3. Else steps = abs(dy)

4 Plot a point at (x, y)

5. xinc = dx / steps;
6. yinc = dy/steps;
7. x = x1 and y = y1
8. Plot a point at (x, y)
9. k=1
10. if k = steps , stop
11. x = x +xinc
12. y = y + yinc
13. Plot a point at (x, y)
14. k = k+1
15. Go to step 7

Bresenham Line Algorithm

1. Highly efficient incremental method
2. Produces mathematically correct results using simple calculations

An algorithm to draw a line using the Bresenham's method (slope is between 0^o to 45^o)

1. Compute

   dx = x2-x1 dy = y2-y1
   inc1 = 2(dy-dx) Inc2 = 2 * dy
   d = inc1 - dx

2. If dx < 0 x = x2, y = y2 and xend =x1
3. if dx > 0 x=x1, y=y1 and xend = x2
4. Plot a point at (x, y)
5. If x= xend, stop
6. if d < 0 then d= d+inc1
7. If d>=0 then d= d+ inc2 y = y+1
8. x = x+1
9. Plot a point at (x, y)
10. Go to step 5

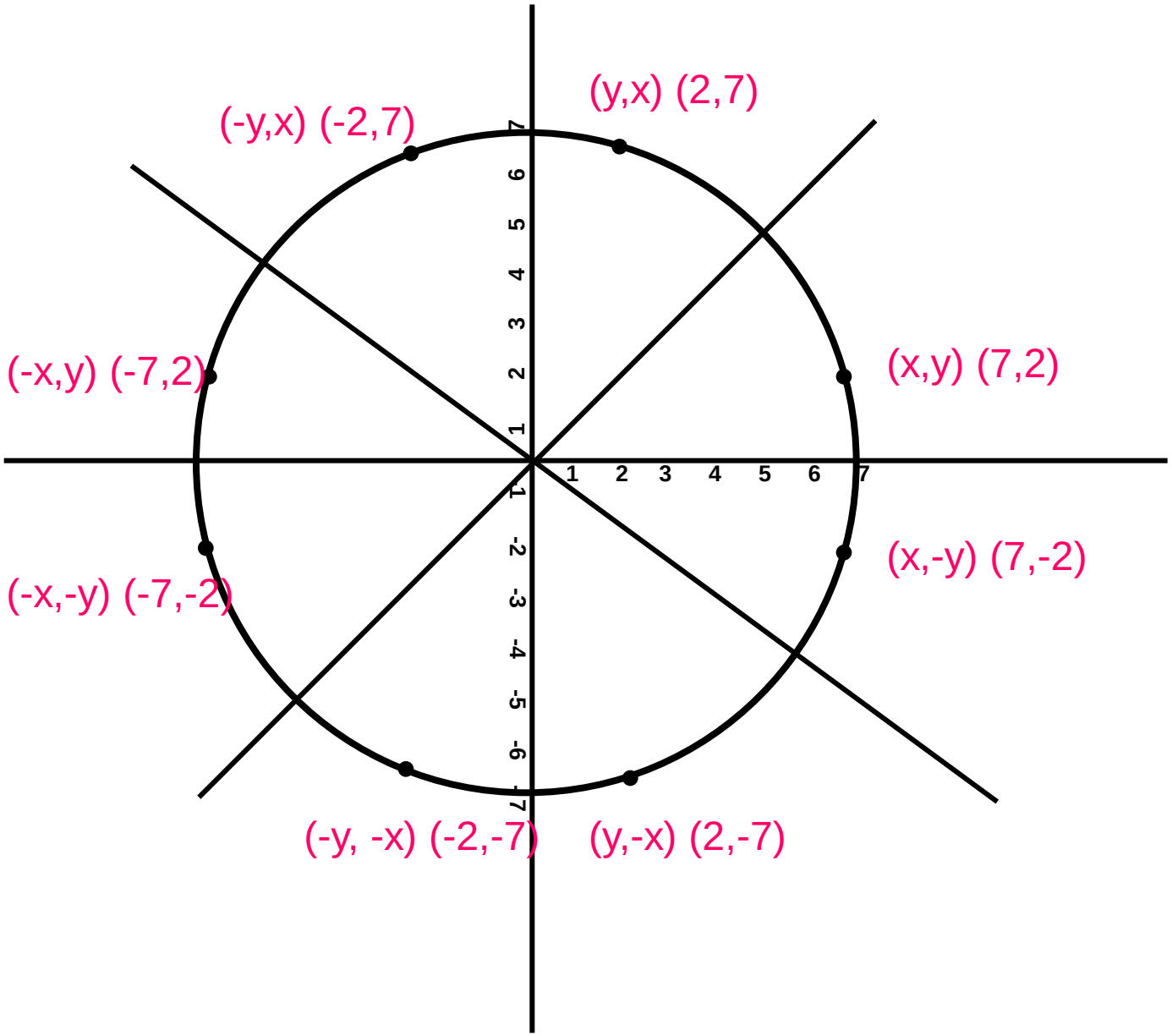
For implementing other cases in Bresenhams algo

Slope of line	Predominant distance	Coord. which changes at each step	Coord. which may or may not change
0< m< 1	x	x	y
m = 1	x or y	x and y	-
1 < m <	y	y	x
0 > m> -1	x	x	y
m = -1	x or y	x and y	-
-1 > m > ∞	y	y	x

SCAN CONVERTING A CIRCLE

- Since circle is a symmetrical figure , eight points can be plotted for each value that the algorithm calculates

Eight way Symmetry of a circle



BRESENHAM CIRCLE ALGORITHM

- To plot the circle efficiently
  - Avoid the use of trigonometric functions and power functions.

1. Set (h,k)= coordinates of circle center, x=0, y= r and the decision parameter d= 3 – 2*r
2. If x>y , stop
3. Plot the eight points, with respect to the current coordinates (x,y)

   Plot (x + h, y + k) Plot (- x + h, - y + k)

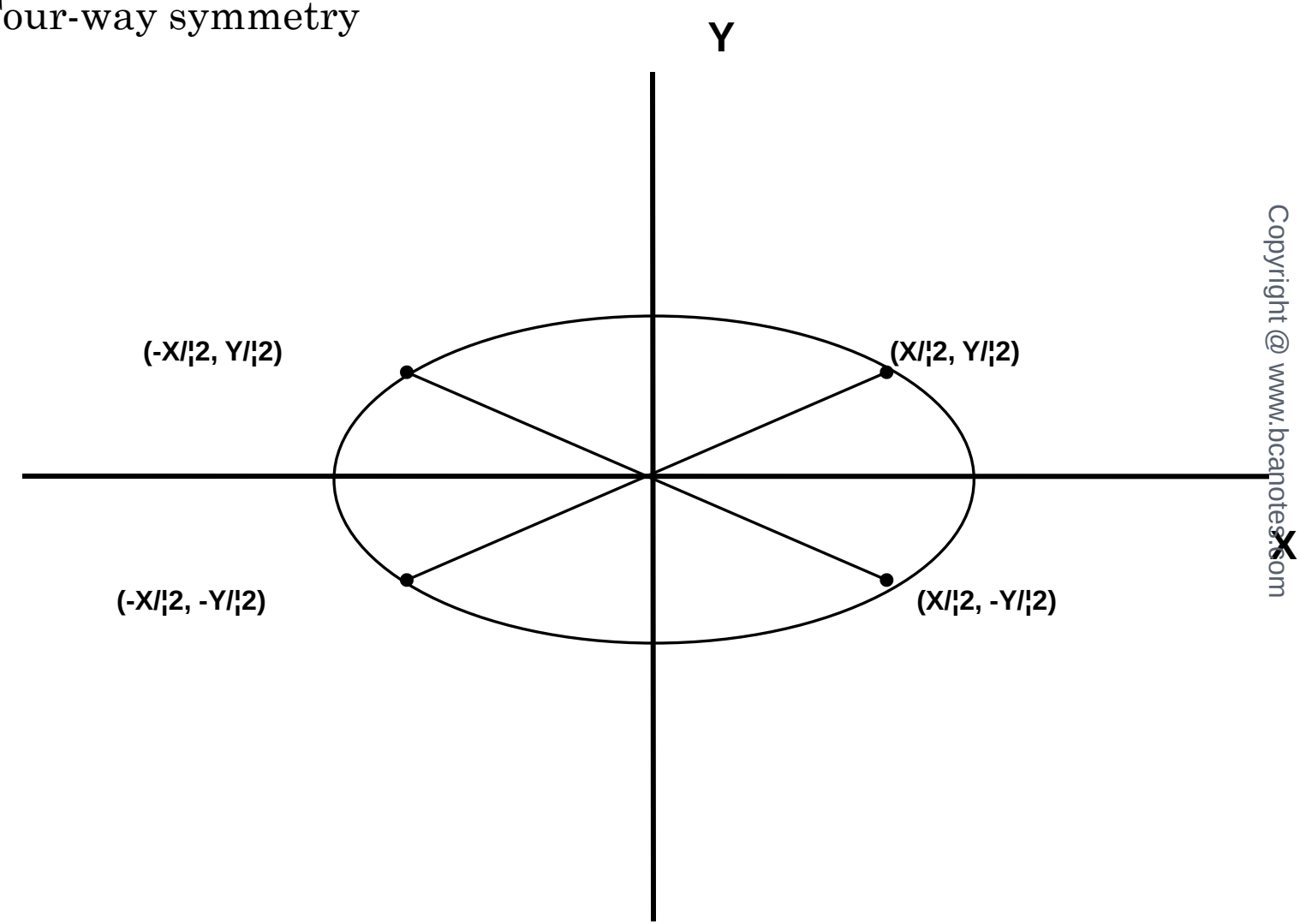
   Plot (y + h, x + k) Plot (- y + h, - x + k)

   Plot (- y + h, x + k) Plot (y + h, - x + k)

   Plot (- x + h, y + k) Plot (x + h, - y + k)
4. If d < 0 then d =d d+4*x + 6 and x = x + 1
5. If d >=0 then d= d + 4*(x-y) + 10 , x = x + 1 and y = y-1
6. Go to step 2

Ellipse

- Shows Four-way symmetry



REGION FILLING

- Process of coloring in a definite image or region
- Regions may be defined at
  - Pixel Level
    - Boundary defined
    - Algorithms are referred as Boundary Fill Algorithms
    - Interior Defined Region
    - Algorithms are referred as flood-fill Algorithms
  - Geometric Level
    - A region is defined or bounded by abstract elements as connected lines and curves

Boundary –Fill Algorithm

- Recursive Algorithm
- Begins with a initial pixel inside the region
- The Algorithm checks to see if this pixel is a boundary pixel or has already been filled.
- If No, It Fills the pixel and makes a recursive call to itself using each and every neighboring pixel as a new seed.
- If Yes, the algorithm simply return to its caller

Flood –Fill Algorithm

- Begins with a initial pixel inside the region
- The Algorithm checks to see if this pixel has the region's original color
- If Yes, It Fills the pixel with new color and uses each of the pixel's neighbors as a new seed in a recursive call.
- If No, the algorithm simply return to its caller

SCAN CONVERTING A CHARACTER

- Unit is Points – 1/72 inch and picas -2 points
- Two Approaches
  - Bitmap Font or Raster
  - Vector or Outline Font

ALIASING EFFECTS OF SCAN CONVERSION

- Staircase
- Unequal Brightness
- The Picket Fence Problem
- Anti-aliasing