Maharaja Agrasen Institute of Technology
ETCS 211

Computer Graphics & Multimedia
UNIT 1

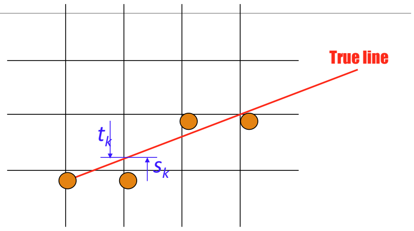
BRESENHAM'S LINE DRAWING ALGORITHMS

The Bresenham's Line Algorithm

The Bresenham's algorithm is another incremental scan conversion algorithm

The big advantage of this algorithm is that it uses only integer calculations

Bresenham's Line Algorithm cont..



- For a given value of *x*
  - one pixel lies at distance t_k above the line, and
  - one pixel lies at distance s_k below the line

Bresenham's Line Algorithm cont..

Decision parameter

$$p_k = (s_k - t_k)$$

- If $p_k < 0$, then closest pixel is below true line (s_k smaller)
- If $p_k \geq 0$, then closest pixel is above true line ($t_i$ksmaller)

- We must calculate the new values for p_k as we move along the line.

Bresenham's Line Algorithm cont..

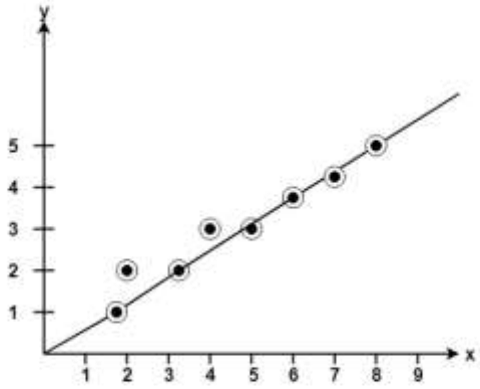
Algorithm

1. Input line end points
2. Load (x_0,y_0) to plot the first point.
3. Calculate dx,dy, 2dy and 2dy-2dx and obtain the starting value of the decision parameter as p_0=2dy-dx
4. At each x_k along the line, starting at k=0, perform the following test
   if p_k <0, the next point plot is (x_k+1, y_k) and
   $p_{k+1}=p_k+2dy$
   Other wise, the next point to plot is (x_k+1, y_k+1) and
   $p_{k+1}=p_k+2dy-2dx$
5. Repeat step-4 dx times.

Starting and Ending position of the line are (1, 1) and (8, 5). Find intermediate points.

x_1=1
y_1=1
x_2=8
y_2=5
dx= x_2-x_1=8-1=7
dy=y_2-y_1=5-1=4
I_1=2* Δy=2*4=8
I_2=2*(Δy-Δx)=2*(4-7)=-6
d = I_1-Δx=8-7=1

x	y	d=d+I_1 or I_2
1	1	d+I_2=1+(-6)=-5
2	2	d+I_1=-5+8=3
3	2	d+I_2=3+(-6)=-3
4	3	d+I_1=-3+8=5
5	3	d+I_2=5+(-6)=-1
6	4	d+I_1=-1+8=7
7	4	d+I_2=7+(-6)=1
8	5	

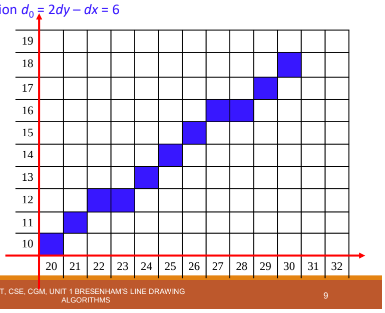


Example

(Bresenham's Line Algorithm)

Draw a line from (20,10) to (30,18)
dx = 10 *dy* = 8 initial decision $d_0 = 2dy - dx = 6$
Also 2*dy* = 16, 2(*dy* – *dx*) = -4

k	p_k	(x_{k+1},y_{k+1})
0	6	(21,11)
1	2	(22,12)
2	-2	(23,12)
3	14	(24,13)
4	10	(25,14)
5	6	(26,15)
6	2	(27,16)
7	-2	(28,16)
8	14	(29,17)
9	10	(30,18)



Bresenham's Line Algorithm cont..

Special cases

- Special cases can be handled separately
  - Horizontal lines (Δy = 0)
  - Vertical lines (Δx = 0)
  - Diagonal lines (|Δx| = |Δy|)
- directly into the frame-buffer without processing them through the line-plotting algorithms.

Bresenham's Algorithm Example Where m<=1

Suppose we want to draw a line starting at pixel (2,3) and ending at pixel (12,8).

dx = 12 – 2 = 10
dy = 8 – 3 = 5
p0 = 2dy – dx = 0

2dy = 10
2dy – 2dx = -10

Algorithm

1.Input the two line endpoints and store left endpoint as (x0,y0)
2.Pre-calculate the values dx, dy, 2dy and 2dy - dx
3.Color pixel (x0,y0)
4.Let p_0 = 2dy –dx
5.At each x_k along the line, starting with k=0:

If p_k<0, then the next point to plot is (x_k + 1,y_k), and p_{k+1} = p_k + 2dy (Down pixel will be selected)
Otherwise, the next point to plot is (x_k + 1, y_k + 1), and p_{k+1} = pk + 2dy – 2dx (Upper pixel will be selected)

6.Repeat Step-4 dx times

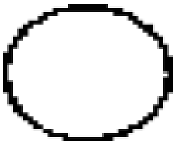
t	p	P(x)	P(y)
0	0	2	3
1	-10	3	4
2	0	4	4
3	-10	5	5
4	0	6	5
5	-10	7	6
6	0	8	6
7	-10	9	7
8	0	10	7
9	-10	11	8
10	0	12	8

Differentiate between DDA Algorithm and Bresenham's Line Algorithm:

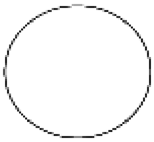
DDA Algorithm	Bresenham's Line Algorithm
1. DDA Algorithm use floating point, i.e., Real Arithmetic.	1. Bresenham's Line Algorithm use fixed point, i.e., Integer Arithmetic
2. DDA Algorithms uses multiplication & division its operation	2.Bresenham's Line Algorithm uses only subtraction and addition its operation
3. DDA Algorithm is slowly than Bresenham's Line Algorithm in line drawing because it uses real arithmetic (Floating Point operation)	3. Bresenham's Algorithm is faster than DDA Algorithm in line because it involves only addition & subtraction in its calculation and uses only integer arithmetic.
4. DDA Algorithm is not accurate and efficient as Bresenham's Line Algorithm.	4. Bresenham's Line Algorithm is more accurate and efficient at DDA Algorithm.
5.DDA Algorithm can draw circle and curves but are not accurate as Bresenham's Line Algorithm	5. Bresenham's Line Algorithm can draw circle and curves with more accurate than DDA Algorithm.

Anti-aliasing

- Anti-aliasing is a method of fooling the eye that a jagged edge is really smooth.
- Due to low resolution aliasing effect will occur, which can be removed by increasing the screen resolution.

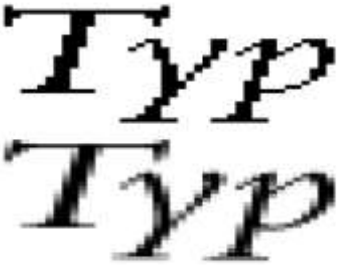


Jagged edges due to aliasing



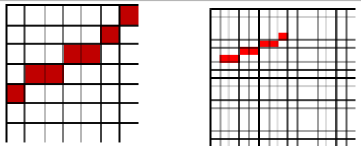
Circle after applying antialiasing

Some more Examples



Reducing Aliasing

- By increasing Resolution



The aliasing effect can be minimized by increasing resolution of the raster display.

Disadvantage of improving resolution

- More memory requirement (Size of frame buffer will become
- Large) More scan conversion time

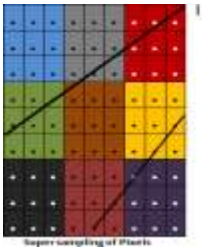
Anti-aliasing Methods

- Super-sampling method or post filtering
- Area sampling or Pre filtering

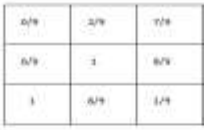
Super-sampling Method (Cont….)

- In this method every individual pixel is subdivided in to sub-pixel.
- In this method we count the number of pixel which are overlapped by the object.
- The intensity value of a pixel is the average of the intensity values of all the sampled sub-pixels with in that pixel.
- In this method every pixel on the screen have different intensity.

Super-sampling for a line object having Non-Zero width.



Super-sampling of Pixels



0/9	2/9	7/9
6/9	1	8/9
1	8/9	1/9

Super-sampling Method (Cont….)

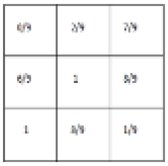
- Pixel at upper right corner is assigned 7/9 because seven of its nine-sub pixels are inside the object area.
- Suppose the color of object is RED(1,0,0), and the background color is light yellow (.5,.5,.5).
- At what intensity the pixel will glow?

(1 X 7/9 +.5 X 2/9, 0X7/9+ 0.5 X 2/9, 0X7/9+.5X2/9)
R G B

Blending of background color and object color will occur only in area of pixel where object overlaps.

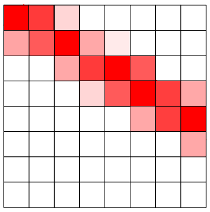
Question-

1.What will be the intensity of center pixel?
Answer- (1 X 1+.5X0, 0X1+.5X0, 0X1+.5X0)
2.What will be the intensity of lower right side pixel?
Answer- (1X1/9 + .5X8/9, 0X1/9+.5X8/9, 0X1/9+.5X8/9)



0/9	2/9	7/9
6/9	1	8/9
1	8/9	1/9

Intensity Variation on pixels after Super sampling method

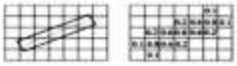


Area Sampling

Area sampling is a pre-filtering technique in which we superimpose a pixel grid pattern onto the continuous object definition. For each pixel area that intersects the object, we calculate the percentage of overlap by the object. This percentage determines the proportion of the overall intensity value of the corresponding pixel that is due to the object's contribution. In other words, the higher the percentage of overlap, the greater influence the object has on the pixel's overall intensity value.

In Fig. 3-26(a) a mathematical line shown in dotted form is represented by a rectangular region that is one pixel wide. The percentage of overlap between the rectangle and each intersecting pixel is calculated analytically. Assuming that the background is black and the line is white, the percentage values can be used directly to set the intensity of the pixels [see Fig. 3-26(b)]. On the other hand, had the background been gray (0.5, 0.5, 0.5) and the line green (0, 1, 0), each blank pixel in the grid would have had the background gray value and each pixel filled with a fractional number f would have been assigned a value of $[0.5(1-f), 0.5(1-f)+f, 0.5(1-f)]$—a proportional blending of the background and object colors.

Although the resultant discrete approximation of the line in Fig. 3-26(b) takes on a blurry appearance, it no longer exhibits the sudden transition from an on pixel to an off pixel and vice versa, which is what we



Write Formula for Blending of Colors for following Conditions

1. Object Background is (.5,.5,.5)

2.Object Color is (1,1,0)

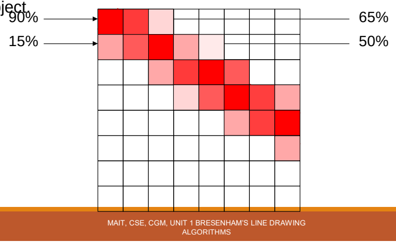
Ans:

Write Formula for Blending of Colors for following Conditions

1. Object Background is (.5,.5,.5)
2. Object Color is (1,1,1)

Area Sampling Method

- This figure shows how line with a non-Zero width have different intensity value at each pixel on the screen
- In this Area sampling method intensity value is determined according to area covered by the object.



Q & A

1. The Cartesian slope-intercept equation for a straight line is
a) y = m.x + b
b) y = b.x + m
c) y = x.x + m
d) y = b + m.m

2. For lines with slope magnitude |m|<1, ?x can be_________
a) A set corresponding vertical deflection
b) A set proportional to a small horizontal deflection voltage
c) Only a
d) All of the mentioned

3. On raster system, lines are plotted with
a) Lines
b) Dots
c) Pixels
d) None of the mentioned

4. Expansion of line DDA algorithm is
a) Digital difference analyzer
b) Direct differential analyzer
c) Digital differential analyzer
d) Data differential analyzer

5. Which algorithm is a faster method for calculating pixel positions?
a) Bresenham's line algorithm
b) Parallel line algorithm
c) Mid-point algorithm
d) DDA line algorithm

6. The disadvantage of lineDDA is
a) Time consuming
b) Faster
c) Neither a nor b
d) None of the mentioned

7. An accurate and efficient raster line-generating algorithm is
a) DDA algorithm
b) Mid-point algorithm
c) Parallel line algorithm
d) Bresenham's line algorithm

8. In Bresenham's line algorithm, if the distances d1 < d2 then decision parameter Pk is______
a) Positive
b) Equal
c) Negative
d) Option a or c

9. Which is the best line algorithm to balance the processing load among the processers?
a) Parallel line algorithm
b) DDA line algorithm
c) Bresenham's line algorithm
d) Position Bresenham's line algorithm

10. The algorithm which uses multiple processors to calculate pixel positions is
a) Midpoint algorithm
b) Parallel line algorithm
c) Bresenham's line algorithm
d) All of the mentioned

11. Coordinate references in the polyline function are stated as
a) Relative coordinate values
b) Absolute coordinate values
c) Current position
d) Real coordinate values