



Bresenham's Line Drawing Algorithm

Mrs. Pournima Abhishek Kamble¹, Mrs. Sujata Shankar Gawade²

Lecturer, Computer Technology, BVIT, Navi Mumbai, India¹

Lecturer, Computer Technology, BVIT, Navi Mumbai, India²

Abstract: The basic principle of this algorithm is to select the optimum raster location to represent a straight line. To accomplish this algorithm always increments either x or y one unit depending on the slope of line. The increment in the other variable is determined by examining the distance between the actual line location and the nearest pixel. This distance is called decision variable or error.

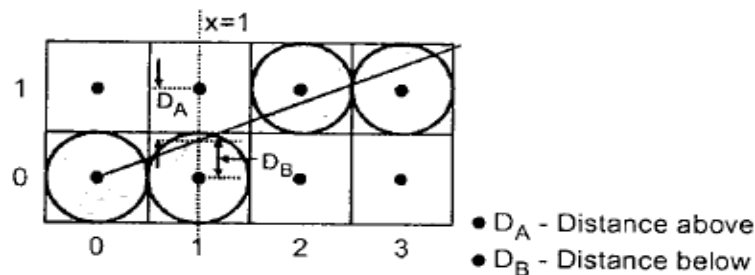
Keywords: Pixel, Decision Variable, Error.

I. INTRODUCTION

In computer graphics, a line drawing algorithm is an algorithm for approximating a line segment on discrete graphical media, such as pixel-based displays and printers. On such media, line drawing requires an approximation (in nontrivial cases). Basic algorithms rasterize lines in one colour. Bresenham's line drawing algorithm uses the optimum raster location to represent a straight line. Bresenham's algorithm uses only integer addition and subtraction and multiplication by 2, So bresenham's algorithm is time efficient.

II. WORKING

Bresenham's line drawing algorithm uses the optimum raster location to represent a straight line as follows:-



- First pixel is drawn at (0,0)
- as $D_B < D_A$, (1,0)
- as $D_B > D_A$, (2,1)
- as $D_B < D_A$, (3,1)

This decisions can be taken depending on the error or decision Variable.

-In mathematical terms error or decision variable is defined as $e = D_B - D_A$

Now if $e > 0$ (means if $D_B > D_A$) the pixel above the actual line is closer than the pixel below the actual line. So, increment $x = x + 1$ and $y = y + 1$.

And if $e < 0$ (means if $D_B < D_A$) the pixel below the actual line is closer than the pixel above the actual line. So, increment $x = x + 1$ and $y = y$.

The error term is initially set as $e = 2dy - dx$ where $dx = x_2 - x_1$ & $dy = y_2 - y_1$

Then according to value of e following actions can be taken as:



```

While(e>=0)
{
y=y+1;
e=e-2dx
}
x=x+1
e=e+2dy

```

III.ALGORITHM

1. Read the line and points (x1,y1) and (x2,y2) such that they are not equal(if equal then plot that point and exit)
2. $dx=x2-x1$ and $dy=y2-y1$
3. Initialize starting point
 $x=x1$
 $y=y1$
4. Initialize e as ; $e=2dy-dx$
5. Initialize counter $i=1$
6.

```

while(e>=0)
{
y=y+1
e=e-2dx
}
x=x+1
e=e+2dy

```
7. plot(x,y)
8. $i=i+1$
9. If($i \leq dx$) then go to step 6.
10. Stop.

IV.PROBLEM

Consider the line from (0,0) to (8,5). Use the bresenham's algorithm to rasterize this line.

solution:

Step 1-4:

$$x1=0, x2=8, y1=0, y2=5$$

$$dx=x2-x1=8-0=8$$

$$dy=y2-y1=5-0=5$$

$$e=2dy-dx=2*5-8=2$$

Initial $x=0$ and $y=0$



i	e	x	y	(x,y)
0	2	0	0	(0,0)
1	-14/-4	1	1	(1,1)
2	6	2	1	(2,1)
3	-10/0	3	2	(3,2)
4	-16/-6	4	3	(4,3)
5	4	5	3	(5,3)
6	-12/-2	6	4	(6,4)
7	8	7	4	(7,4)
8	-8/2	8	5	(8,5)

V. GENERALIZED BRESENHAM'S ALGORITHM

- The bresenham's algorithm only works for the first quadrant.
- The generalized bresenham's algorithm requires modification for line lying in the other octant.
- This can be done by considering the quadrant in which the line lies.

1st quadrant $x=x+1$ & $y=y+1$
 2nd quadrant $x=x-1$ & $y=y+1$
 3rd quadrant $x=x-1$ & $y=y-1$
 4th quadrant $x=x+1$ & $y=y-1$

- This can be done by introducing A sign function as $s1=\text{sign}(x2-x1)$
 $s2=\text{sign}(y2-y1)$

And depending upon the slop of A function the value of x or y will Be incremented .

```

If(dx>dy)
{
Exchange=1;
}
else
EXCHANGE=0;

```

Algorithm:

Step 1 :Read the line end point $(x1,y1)$ and $(x2,y2)$ such that they are not equal.

Step 2: $dx=x2-x1$ and $dy=y2-y1$

Step 3: initialize starting point

```

x=x1
y=y1
plot(x,y)

```



Step 4: $s1 = \text{sign}(x2 - x1)$
 $s2 = \text{sign}(y2 - y1)$ [sign function returns -1,0,1 depending upon argument is <0, >0]

Step 5: if $dy > dx$ then exchange dx by dy
 exchange=1
 else exchange=0

Step 6: $e = 2 * dy - dx$

Step 7: $i = 1$

Step 8: while($e \geq 0$)
 {
 if(exchange=1) then $x = x + s1$
 else $y = y + s2$
 end if $e = e - 2 * dx$
 }
 if(exchange=1) then $y = y + s2$
 else $x = x + s1$
 end if $e = e + 2 * dy$

Step 9: plot(x,y)

Step 10: $i = i + 1$

Step 11: if($i \leq dx$) then go to step 8

Step 12: stop

REFERENCES

- [1]. Donald Hearn, Baker M. Pauline, "Computer Graphics", Pearson Education, New Delhi, June 2012, ISBN: 817758765X.
- [2]. Maurya Rajesh K., "Compute Graphics", Wiley-India 2011, Delhi ISBN: 978-81-265-3100-4.
- [3]. Dr. Chopra Rajiv, "Computer Graphics", S. Chand 2016, New Delhi, ISBN: 978-93-856-7633-8.
- [4]. Foley James, "Computer Graphics principles and practices", Pearson Education, New Delhi 2014, ISBN: 978-0-321-39952-6.