Improved Line Drawing Algorithm: An Approach and Proposal

Muhammad.Usman khan
Computer Science & Engineering
Integral University
Lucknow, India
usmanintegral@gmail.com

Prof (Dr) Md.Rizwan Beg
Computer Science & Engineering
Integral University
Lucknow, India
rizwanbeg@gmail.com

Mohd Zunnun Khan
Computer Science & Engineering
Integral University
Lucknow, India
Zunnunkhan@gmail.com

***Abstract*: This paper investigates aliasing along straight line segments or edges and its origin, and how it is affected by the orientation or slope of the segment. A method for anti-aliasing or smoothing the straight line segments by modifying the intensity of the pixels is presented and this paper proposes a line drawing scheme to draw a line by calculating deflection-angle and error to draw the next possible pixel more accurately both starting from co-ordinate point (x_1,y_1) (x_2,y_2) to meet mid (m), so that there will be a pattern of pixels nearer to the imaginary line. The basis of the algorithm is to draw pairs of pixels from both the end points straddling the line.**

Keyword: **DDA algorithms, jaggies, Anti-aliasing, Angle deflection, slope, pixel.**

I. INTRODUCTION

We know that lines are very important in the field of science and mathematics, with the advent of pixels concept in the field of computer graphics line can be draw on pixels based movement, but it is very important to draw a line without any error,almost all displays are raster displays where raster graphics use a matrix of pixels to represent images [1,2]. The rasterization of a straight line segment can be accomplished using the line drawing algorithm called a Digital Differential Analyzer (DDA). it can be done using Bresenham's algorithm (a modified DDA) which uses integer mathematics only [3]. However, both produce the same pixels with the same aliasing effect. A line segment is defined by an infinite set of points which lie between two end points or vertices but its rasterization represents it with its samples or defined fixed positions only based on the screen resolution. Smooth straight lines appear stair like lines when displayed for variety of reasons, the most common being that the output device (display monitor) does not have enough resolution to portray a smooth line as mentioned above. This means that sampling is done below the under sampling (Nyquist rate)

[4,5,6]. This algorithms projects the endpoints to their pixel locations in the screen co-ordinates as integers and find a path of pixels between the two starting points and plots the line on the monitor from frame buffer (video controller).Rounding causes all lines except horizontal or vertical to be displayed as jigsaw appearance because of low resolution. This algorithm improves present projection of an algorithm which reduces round off error to the negligible limit.

So, it is very important to know, how the line drawing concept work, we have some basic ideas of line drawing algorithm are as follows.

Line Equation

The Cartesian slop-intercept equation for a straight line is

$$y=mx+b \qquad \text{------} \qquad (1)$$

The 2 end points of a line segment are specified at a position(x_1, y_1), (x_2, y_2).

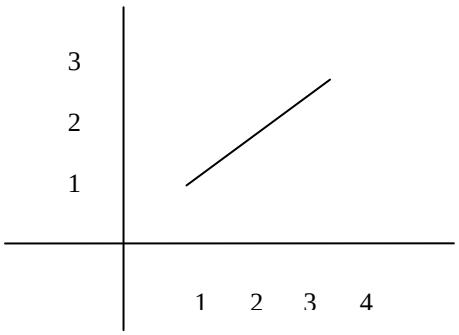

Figure 1. Line drawn between co-ordinate points (x_1,y_1) (x_2,y_2)

Determine the values for the slope m and y intercept b with the following calculation.

here, slope m:

$$m = (y_2 - y_1) / (x_2 - x_1)$$

$$m= dy / dx \qquad \text{------} \qquad (2)$$

y intercept b

b=y1-mx1 ------ (3)

Algorithms for displaying straight line based on this equation & y interval dy from the above equation.

m= dy / dx
dy= m. dx ------ (4)

Similarly x interval dx from the equation

m = dy / dx
dx = dy /m ------- (5)

A. Line DDA Algorithm:

The digital differential analyzer (DDA) is a scan conversion line algorithm based on calculation either dy or dx. The line at unit intervals is one co-ordinate and determine corresponding integer values nearest line for the other co-ordinate[7].

Consider first a line with positive slope.

1) Step 1:

If the slope is less than or equal to 1, the unit x intervals dx=1 and compute each successive y values.

dx=1

m = dy / dx
m = (y_2-y_1) / 1
m = (yk+1 – yk) /1

yk+1 = yk + m -------- (6)

k takes integer values starting from 1,for the first point and increment by 1 until the final end point is reached. m = any real numbers between 0 and 1[7].

Calculate y values must be rounded to the nearest integer

2) Step 2:

If the slope is greater than 1 ,the roles of x any y at the unit y intervals dy=1 and compute each successive y values.

dy=1

m= dy / dx
m= 1/ (x_2-x_1)
m = 1 / ($x_{k+1} – x_k$)

x_{k+1} = x_k + (1 / m) ------- (7)

Equation 6 and Equation 7 that the lines are to be processed from left end point to the right end point.

3) Step 3:

If the processing is reversed, the starting point at the right
d x= -1

m= dy / dx
m = ($y_2 – y_1$) / -1
y_{k+1} = y_k – m -------- (8)
Intervals dy=1 and compute each successive y values.

4) Step 4:

Here,
dy = -1

m= dy / dx
m = -1 / (x2 – x1)
m = -1 / (xk+1 – xk)

xk+1 = xk + (1 / m) ---- (9)

Equation 6 and Equation 9 used to calculate pixel position along a line with –ve slope[7].

II. MOTIVATION

A lot of works has already been done in the field of computer graphics on line drawing, but unfortunately we couldn't remove the introduction of errors during the line drawing, described as Round-off errors where line diverges more and more from straight line and from original co-ordinate points. Round-off operation it time taking process, it performs integer arithmetic by storing float as integers in numerator and denominator and performing integer arithmetic has deflection is caused by aliasing. Aliasing in a straight line segment generated by the DDA is a function of its orientation [8]. When the line is horizontal or vertical, no aliasing appears and all the generated pixels are exactly located along the straight line. This means that the error value for each generated pixel is zero. This is also true when the slope of the line is +1 or -1[8] . All other cases, aliasing is produced as a function of two parameters. The first parameter is the DDA accumulating error, which is repeatable in nature. The second parameter is the slope or orientation of the line. In figure (2) a demonstration example of the aliasing along a single straight line segment is shown (up) with the corresponding error function (f) is shown down[8].

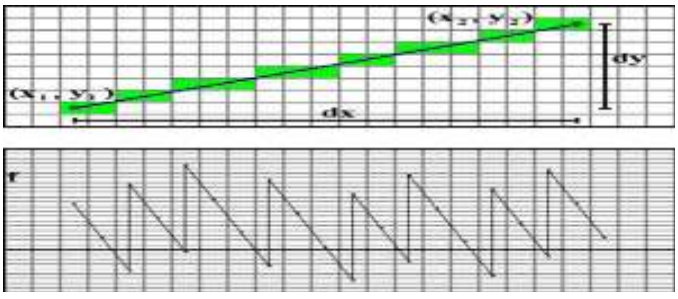

Figure 2. Aliasing example with accumulating error f (borrowed from reference [8])

After studying of DDA & Bresenham's line drawing algorithms, we proposes a modified scheme to reduce the errors which draws line by calculating deflection angle and error to draw next possible pixel more accurately. so that there will be a pattern of pixel nearer to the imaginary line by following parameter. The accumulation of round of error is successive addition of the floating point increments is used to find the pixel position but it take lot of time to compute the pixel position[9].

III. PRIVIOUS WORK

In 1987, the researcher Roman P.Molla designed different algorithms to implement scan conversion for a straight line segment. The Digital Differential Analyzer (DDA) and Bresenham's algorithms were designed using serial processing in addition to the implementation of the DDA algorithm using parallel processing. The research discussed the performance, cost and the error ratio for the above mentioned systems [10]. Andreas Schilling presented in 1991 a hardware realization of an algorithm for antialiasing. He mainly used PLA's in his design. The algorithm is based on sub pixel mask look up table [11].In1993, Andreas Schilling and Wolfgan Straber introduced an algorithm that deals with hidden surface elimination problem at pixels level. The algorithm provided the solution of the aliasing problem resulted in the scan conversion operation. The hardware implementation was divided into three stages in order to apply the pipeline technique to improve the performance. The designed architecture costs 12000 gate and the chip has ability to produce 27pixel/sec[8] In 2004, P. Beaudoin and P. Poulin proposed a mechanism to compress the antialiasing buffer and limit the bandwidth requirements for hardware edge antialiasing. The presented method supports the usual OpenGL fragment-related functions [5]. In 2004 also, Y. K. Liu et al presented an integer one-pass algorithm for voxel traversing along a line. The proposed approach is based on a modification of the well known Bresenham's algorithm [2]. D. Wang et al presented in 2006 an antialiasing method using a DSP-based display system for removing the undesired jaggies occurred in the line drawing [8].

IV. PROPOSED WORK

Fundamentally, there are three methods which can be used for antialiasing. Since aliasing in computer-generated graphics is a spatial aliasing, an obvious solution is to increase the sampling rate or raster resolution which decreases aliasing effect. This method is limited by the display hardware [12, 13]. In the second method, super sampling is used which is a technique of collecting data points at greater resolution (usually by a power of two) than the final data resolution. These data points are then combined or averaged (down sampling) to the desired resolution. This technique is referring to as post-filtering the image [6, 14, 15,16,17,18]. The third method of antialiasing treats each pixel as a finite area rather than a point and is known as pre-filtering the image [17, 19,20]. The last two methods can only be implemented using systems with a display of more than two intensities per pixel. The third method has a computational advantage over the second one since it does not require a large memory for storing the image at the sub-pixel stages. This is why pre-filtering techniques are cheaper, but still effective, when implemented [11]. By the study of various papers ,this paper proposes a new line drawing scheme to draw a line by calculating deflection-angle and error to draw the next possible pixel more accurately both starting from co-ordinate point (x_1,y_1) (x_2,y_2) to meet mid (m), so that there will be a pattern of pixels nearer to the imaginary line. The basis of the algorithm is to draw pairs of pixels from both the end points straddling the line and we are defining two starting co-ordinate points.

i.e

` Line 1 (x_1,y_1) &

Line 2 (x_2,y_2)

Let us assume two starting point to draw line after calculating mid (m).We apply DDA on both starting point that reaches to mid (m).

For the calculation of meeting point (m) of line1 & line 2

$$m=(x_1+x_2/2,\ y_1+y_2/2)$$

It draws lines quickly; it can be handled by the case where the line endpoints do not lie exactly on integer points of the pixel grid. A naïve approach to anti-aliasing the line would take an extremely long time. The basis of the algorithm is to draw pairs of pixels straddling the line by starting with two point and both are the starting point so line meet at the mid of the both line-1and line-2. Pixels at the line ends are handled separately [21,22,23].

DDAs are used for rasterization of lines, in its simplest implementation the DDA algorithm interpolates values in interval $[(x_{start}, y_{start}), (x_{end}, y_{end})]$ by computing for each x_i the equations [24]

$x_i = x_{i-1}+1/m,$

$y_i = y_{i-1} + m,$

Where $\Delta x = x_{end} - x_{start}$ and

$\Delta y = y_{end} - y_{start}$

and $m(\text{mid}) = \Delta y/\Delta x$

Therefore some cases which also exists in current problem statement. To solve this issue we assume three cases i.e. as follows

i) Case 1 :

When both lines divert in same +ve direction from mid point (m).

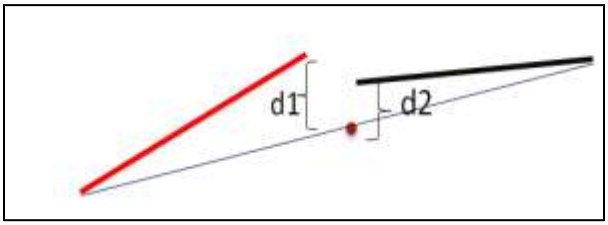

Figure 3. Diflaction in both +ve Diarection.

ii) Case 2 :

When both lines divert in same -ve direction from mid point (m).

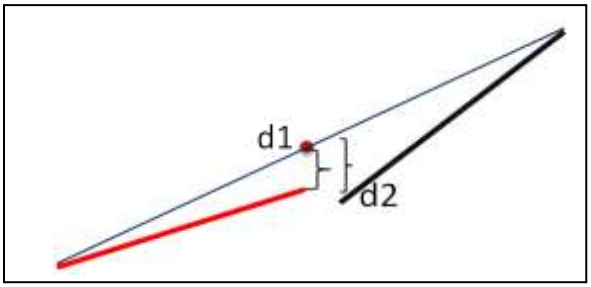

Figure 4. Diflaction in both -ve Diarection.

iii) Case 3 :

When both lines divert in both +ve & -ve direction from mid point (m).

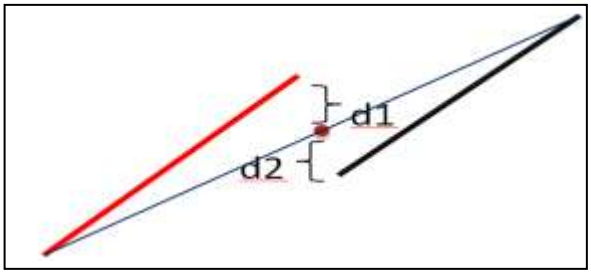

Figure 5. Diflaction in both +ve & -ve Diarection.

For the perfect line of both the side we have to find deflection-angles.

A. Calculation of angles

These angles are calculated by the Pythagoras theorem and Now calculate the deflection from mid point by Cosθ1, Cosθ2 It can be calculate by the Pythagoras theorem

- Cosθ1 = *B/H* = *am/ah1* .
- Cos θ2 = *B/H* = *bm/bh2* .

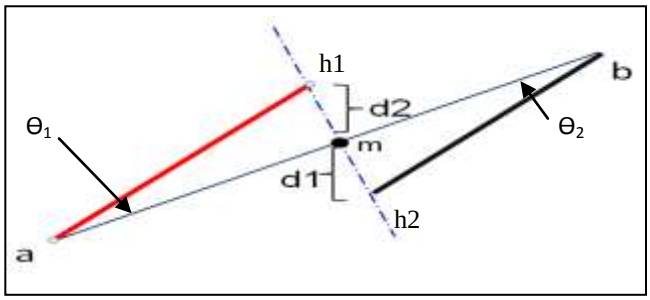

Figure 6. calculation of diflaction-anglke angle tan θ1 tan θ2.

we can find out the angle tan θ1 tan θ2,and directly find out the deflection angles θ1, θ2 or d1,d2.

- *tan θ1* = *P/B* = *d2/am* ,
- *tan θ2* = *P/B* = *d1/bm*

by given method we can rotate the line on calculated angle cosθ1,cosθ2 by using rotation and redraw the both lines on calculated angles θ1, θ2 and deflections d1,d2. So,proposed line drawing scheme is able to draw line by calculating error and draw next possible pixel more accurately so that there will be a pattern of pixel nearer to the imaginary line by following DDA

B. Algorithm : A naive line-drawing algorithm

```
dx = x2 - x1
dy = y2 - y1
for x from x1 to x2
{
   y = y1 + (dy) * (x - x1)/(dx)
   pixel(x, y)
}
```

This proposed method is used to modify Bresenham's algorithm, when using a DDA to draw a single straight line segment on a raster display. To reduce aliasing, the intensity of each affected pixel is computed using a convolution integral [24]. The anti-aliasing must be needed by an intensity function that is used for exponential value which is symmetric around the origin and can be presented [11]. The equation that is used in solving intensity function are as follow:

I = exp (– | kf |)	k = 2	(1)
I = 1- |k f |	k = 1.264	(2)
I = cos (k f)	k = 2.388	(3)

Where: "f" represents the error function of a value range (-0.5 to 0.5).
"I" normalized intensity value between 0 to 1.

Line segments with different slopes are shown before antialiasing and after with a value of "k" being 2 in figure (8).

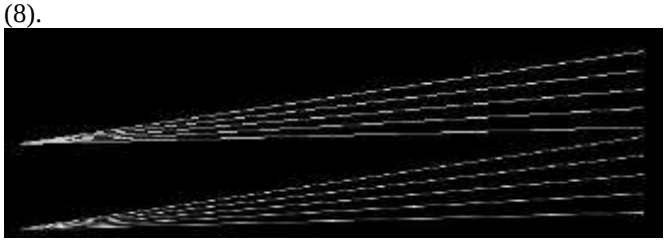

Figure 7. Lines with alisaing (UP) and antialiasing (Down).(figure borrowed from referance [8])

If "k" (k = 2, 4, 6, 8) as demonstrated in Figure (9, 10, 11, 12).

Figure 8. A

Figure 9. B

Figure 10. C

Figure 11. D

Figure 12. Different levels of antialiasing (A to D) demonstrated in figure (9,10,11,12).(figure borrowed from referance [8])

V. RESULT

The concept determines the best approximation to a desired line on the screen. Combination of rasterization and generation of the pixel on scan line is in order. It will show straight lines as straight lines. The following results are shown after applying this approach.

They must start and end accurately. Lines should have constant brightness along their length. Lines should be drawn rapidly

VI. CONLUSION & FUTURE WORK.

The error produced by scan-converting any straight line segment to its displayable pixels is repeatable in nature and it is a strong function of its slope or orientation. The same pixel error values are produced either using the DDA or Bresenham's algorithm. A variable level of antialiasing is feasible only via changing the value of one parameter (k) in the intensity function used for anti-aliasing[8],and it permits using any intensity function by just reprogramming the values with its discrete values. However, the main contributions in the field of antialiasing in computer graphics cannot be included in this paper i.e. novel analysis of the pixel error presented that is resolve only by hardware solution.

So, the fixed-point integer operation requires two additions per output cycle. The probability of fractional part overflows proportional to the ratio m of the interpolated start/end values [26], and probability of line drawing on screen is easy by this proposed paper approach.

In fact any two consecutive point(x,y) laying on this line segment should satisfy the equation. This line drawing implemented by using angle deflection between imaginary line and line drawing given by co-ordinate, floating-point or integer arithmetic.

REFERENCES.

[1] Joey Lawrance ," The History of Computer Graphics ", CS 551 Term Paper, University ofUtah, February 19, 2004.

[2] Y.K. Liu, B. Zalik and H.Yang," An Integer One-Pass Algorithm for Voxel Traversal ",Computer Graphics Forum, Volume 23 , Number 2, 2004, pp 167-172.

[3] Angel, E., and Morrison, D. Speeding up Bresenham's Algorithm. IEEE Computer,Graphics and Application, 11(6), pp. 16-17, 1991.

[4] Dusheng Wang , Hock Lye Toh , Xiang Chen , Fan Yang ," A Simple Anti-Aliased Method for Straight Line Drawing Based On DSP Platform ", Posters proceedings ISBN 80-86943-04-6 , WSCG' 2006 , January 30 – February 3, 2006.

[5] Philippe Beaudoin and Pierre Poulin ," Compressed Multisampling for Efficient Hardware Edge Antialiasing", LIGUM, Dept. I.R.O., University of Montreal, March 2004.

[6] Peter Longhurst, Kurt Debattista, Richard Gillibrand, Alan Chalmers ," AnalyticAntialiasing for Selective High Fidelity Rendering ", Dept. of Computer Science , University of Bristol, BS8 1UB, UK, 2005.

[7] http://i.thiyagaraaj.com/tutorials/tutorial-downloads/dda-line-algorithm-tutorial-download

[8] Al-Rafidain Engineering Vol.17 No.2 April 2009

[9] http://i.thiyagaraaj.com/tutorials/tutorial-downloads/dda-line-algorithm-tutorial-download

[10] Roman P.Molla Vaya , "Parallel Fixed Point Digital Differential Analyzer " , ComputerJournal , Vol. 23, No. 1, pp:46-52, 1987.

[11] Andreas Schilling ," A New Simple and Efficient Antialiasing with Subpixel Masks ",Computer Graphics , Vol. 25, no. 4 , July 1991 (SIGGRAPH '91 Proceedings), pp. 133-141.12-Peter Young ," Coverage Sampled Antialiasing ", Technical Report , NIVIDIACorporation , October 30, 2006.

[12] WU, X. ***An Efficient Antialiasing Technique***. Computer Graphics, vol. 25, No.4, pp. 143-152, Jul 1991.

[13] M. Golipour-Koujali, " General Rendering and Antialiasing Algorithms for Conic Sections ", PhD thesis, London South Bank University, England, pp. 120-123 May 2005.

[14] Andreas Schilling ," A New Simple and Efficient Antialiasing with Subpixel Masks ",Computer Graphics , Vol. 25, no. 4 , July 1991 (SIGGRAPH '91 Proceedings), pp. 133-141.

[15] Peter Young ," Coverage Sampled Antialiasing ", Technical Report , NIVIDIACorporation , October 30, 2006.

[16] Fabris, A.E., Robust Antialiasing of Curves, PhD thesis, University of East Anglia,Norwich, November 1995.

[17] Fabris, A.E. and Forrest, A.R. Antialiasing of curves by discrete prefiltering. In Computer Graphics (SIGGRAPH'97 Proceedings), August 1997.

[18] Molnar, S. Efficient Super-sampling Antialiasing for High-performance Architecture.Technical report, 91-023, University of North Carolina, 1991.

[19] Lathrop, O., Kirk, D. and Voorhies, D. Accurate Rendering by Sub-pixel Addressing.IEEE Computer Graphics & Applications, pp. 45-53, Sep 1990.

[20] 17-Qyvind Ryan ," Application Of Antialiasing In An Image Processing Framework Setting ",Dept. of Informatics, University of Oslo, Norway, 2006.

[21] 3Abrash, Michael (June 1992). "Fast Antialiasing (Column)". *Dr. Dobb's Journal* **17** (6): 139(7).

[22] 4Wu, Xiaolin (July 1991). "An efficient antialiasing technique". *Computer Graphics* **25** (4): 143–152. doi:10.1145/127719.122734. ISBN 0-89791-436-8. http://portal.acm.org/citation.cfm?id=122734.

[23] 5Wu, Xiaolin (1991). "Fast Anti-Aliased Circle Generation". In James Arvo (Ed.). *Graphics Gems II*. San Francisco: Morgan Kaufmann. pp. 446–450. ISBN 0-12-064480-0.

[24] "http: // en.wikipedia.org/wiki/ Bresenham's_line_algorithm".

[25] "http://i.thiyagaraaj.com/articles/articles/dda-line-algorithm."

[26] "http://en.wikipedia.org/wiki/Digital_differential_analyzer_%28graphics_algorithm%29"