



A Mathematical Overview of Bresenham Algorithms in the Determination of Active Pixel Positions

Makanjuola Daniel*

Institute of Postgraduate Research and Studies, Cyprus International University, Nicosia, North Cyprus

E-mail: odundaniel@yahoo.com

Abstract: A computer display monitor, usually called simply a monitor, is a piece of electrical equipment which displays viewable images generated by a computer without producing a permanent record. The monitor comprises the display device, circuitry to generate a picture from electronic signals sent by the computer, and an enclosure or case. Within the computer, either as an integral part or a plugged-in-interface, there is circuitry to convert internal data to a format compatible with a monitor. The CRT or Cathode Ray Tube is the picture tube of a monitor. The back of the tube has a negatively charged cathode. The electron gun shoots electrons down the tube and onto a charged screen. The screen is coated with a pattern of dots that glow when struck by the electron stream. Each cluster of three dots, one for each colour, is one pixel. Several algorithms and models have been staged up at determining the position of pixels resourceful at displaying an image on the screen. This paper is inclined to illustrate a bird-view of mathematical-based bresenham's algorithms used in determining the positions of active pixels in forming images on the screen of a display system.

Keywords: Image processing; Bresenham's algorithm; Computer vision

I. INTRODUCTION

1.1 Formation of Pixels

The image on the screen is usually made up from at least tens of thousands of such tiny dots (phosphor) glowing on command of the computer. The closer together the pixels are, the clearer the image resolution. The distance between pixels on a computer monitor screen is called its dot pitch and is measured in millimetres. The maximum distance between two given pixel is 0.28 mm. As noted by Mc Graw Hills [1], the picture analysis of pixel formation is further explained in Figure 1.

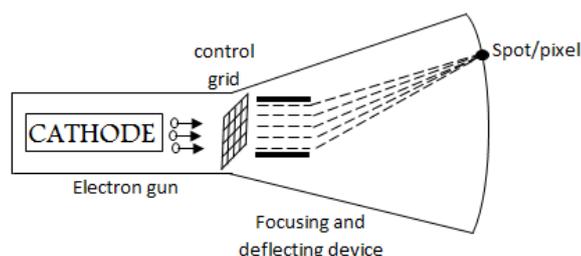


Figure 1: Cathode ray tube.

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 10, October 2017

There are two electromagnets around the collar of the tube which deflect the electron beam. The beam scans across the top of the monitor from left to right is then blanked and moved back to the left-hand side slightly below the previous trace (on the next scan line), scans across the second line and so on until the bottom right of the screen is reached. The beam is again blanked, and moved back to the top left to start again. This process draws a complete picture, typically 50 to 100 times a second. The number of times in one second that the electron gun redraws the entire image is called the refresh rate and is measured in hertz (cycles per second). All pixels are illuminated whenever an image display system is switched on, but not all the pixels are used to display images on the screen [2]. Several algorithms have been staged up at determining the positions of active pixels resourceful at displaying images on the screen.

II. RELATED RESEARCHES

In my recently published paper, titled, “The Formation of Graphical Images Using Resampling Pixel Addressing Mode and its Effects on 12 X 12 Block Memory”, various contributory researches with respect to pixel formation were highlighted [2]. Part of the reports in line with my interest is the contribution [3] which is based on models of low-level human vision in detecting perceptually important regions on the image by building important maps based on various visual characteristics Another contribution is a study undertaken [4], which is an illustration of a multi-scale centre-surround contrast. The method presented by Syeda-Mahmood uses segmentation in homogenous colour regions [5]. The method proposed by Di Gesu et al. uses symmetry transforms, Stentiford used dissimilarities in local neighbourhoods, Kadir and Brady [6-8] used a local measure of entropy to detect salient features in an image [8]. Wang et al. [9] used a wavelet transform and Carlotto and Stein used a fractal model [10]. Most of these methods are based on different structural characteristics of the image without explicit reference on pixel formation. The first pioneer algorithm aims at determine pixel position is called Digital Differential Analyzer (DDA) Algorithm [1]. In this algorithm, the line is sampled at unit intervals in one coordinate and find the corresponding values nearest to the path for the other coordinate.

For a line with positive slope less than one $\Delta x > \Delta y$ (where $\Delta x = x_2 - x_1$ and $\Delta y = y_2 - y_1$). Hence we sample at unit x intervals and compute each successive y values as;

$$Y_{k+1} = y_k + m$$

Where, m is the gradient (Slope).

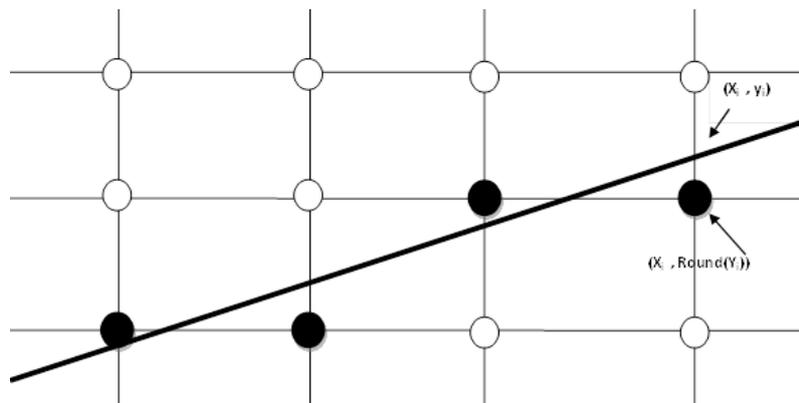


Figure 2: Determining pioneer algorithm.

Furthermore, for lines with positive slope greater than one, $\Delta y > \Delta x$. Hence, we sample at unit y intervals and compute each successive x values as $x_{k+1} = x_k + 1/m$.

Since the slope, m can be any real number; the calculated value must be rounded to the nearest integer (Figure 2).

For a line with negative slope, if the absolute value of the slope is less than one, we make unit increment in the direction and calculate y values as;

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 10, October 2017

$$Y_{k+1} = y_k + m$$

Again for a line with negative slope, if the absolute value of the slope is greater than one, we make unit decrement in the y-direction and calculate x values as;

$$X_{k+1} = x_k - 1/m$$

III. BRESEHAM'S ALGORITHM

Bresenham's algorithm is developed by Jack Bresenham, in this method; we look at just the center of the pixels. We determine d1 and d2 which is the "error", i.e. the difference from the "true line".

Procedures in Bresenham's Algorithm

- i. Determine the error term.
- ii. Define a relative error term such that the sign of this term tells which pixel to choose.
- iii. Derive equation to compute successive error terms from the first.
- iv. Compute the first error term.

IV. MATHEMATICAL CONCEPTS OF BRESEHAM'S ALGORITHMS

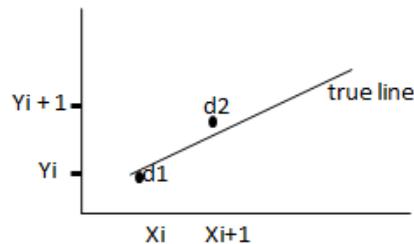


Figure 3: Mathematical concepts of bresenham's algorithms.

From the above (Figure 3), the y coordinate on the mathematical line at pixel position xi+1 is calculated as;

$$y = m(x_i + 1) + b \dots\dots\dots (i)$$

and the distances are calculated as ;

$$d_1 = y - y_i = [m(x_i + 1) + b] - y_i \dots\dots\dots (ii)$$

$$d_2 = (y_i + 1) - y = y_i + 1 - [m(x_i + 1) + b]$$

Then,

The error term is given as;

$$d1 - d2 = m(x_i + 1) + b - y_i - [y_i + 1 - [m(x_i + 1) + b]]$$

$$= m(x_i + 1) + b - y_i - y_i - 1 + m(x_i + 1) + b$$

To define the relative error term, we have;

$$P_i = dx (d_1 - d_2)$$

$$P_i = dx [2m(x_i + 1) - 2y_i + 2b + 1] \dots\dots\dots (iii)$$

Take $m = dy/dx$ and substituting into equation (iii), we have;

$$P_i = dx[2dy/dx(x_i + 1) - 2y_i + 2b + 1]$$

$$P_i = 2dy(x_i + 1) - 2y_i dx + 2bdx + dx$$

$$P_i = 2dyx_i + 2dy - 2y_i dx + 2bdx + dx$$

$$P_i = 2*dy*x_i - 2*dx*y_i + 2dy + dx(2b - 1) \dots\dots\dots (iv)$$



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 10, October 2017

Let $K = 2dy + dx(2b - 1)$

$$P_i = 2*dy*x_i - 2*dx*y_i + K$$

Now to derive the successive point on x terms, we have;

$$P_{i+1} = 2dy(x_{i+1}) - 2dx(y_{i+1}) + K$$

To derive equation for the successive error term, we have;

$$P_{i+1} - P_i = 2dy(x_{i+1} - x_i) - 2dx(y_{i+1} - y_i) \dots\dots\dots (v)$$

With $x_{i+1} = x_i + 1$ or $y_{i+1} = y_i + 1$ and inserting this in equation (v), we have;

$$P_{i+1} - P_i = 2dy - 2dx(y_{i+1} - y_i)$$

Hence,

$$P_{i+1} = P_i + 2dy - 2dx(y_{i+1} - y_i) \dots\dots\dots (vi)$$

Computing the first error term i.e computing $P_i(x_i, y_i)$ from equation (i)

Where $b = y - dy/dx * x$

Recall equation (iv),

$$P_i = 2dy*x_i - 2*dx*y_i + 2dy + dx [2(y - dy/dx*x) - 1]$$

$$P_i = 2*dy*x_i - 2*dx*y_i + 2dy + dx[(2y - 2dy/dx * x) - 1]$$

$$P_i = 2*dy* x_i - 2*dx*y_i + 2dy + 2ydx - 2dy*x - dx\dots\dots (vii)$$

Rearranging,

$$P_i = 2dy - dx$$

If $p_i < 0$, plot the pixel $(x_i + 1, y_i)$ and next decision parameter is $p_{i+1} = p_i + 2dy$ else, plot the pixel $(x_i + 1, y_i + 1)$ and next decision parameter is $p_{i+1} = p_i + 2dy - 2dx$

4.1 Bresenham Algorithm for 1st Octant

- I. Enter endpoints (x_1, y_1) and (x_2, y_2)
- II. Display x_1, y_1
- III. Compute $dx = x_2 - x_1$; $dy = y_2 - y_1$; $p_1 = 2dy - dx$
- IV. If $p_1 < 0.0$, display $(x_1 + 1, y_1)$, else display $(x_1 + 1, y_1 + 1)$
- V. If $p_1 < 0.0$, $p_2 = p_1 + 2dy$, else $p_2 = p_1 + 2dy - 2dx$
- VI. Repeat steps 4, 5 until reach x_2, y_2

V. CONCLUSION

In a CRT monitor, the electron beam is turned on to illuminate the screen phosphor at the selected location. Depending on the display technology, the positioning of the electron beam changes. In a random-scan (vector) system point plotting instructions are stored in a display list and the coordinate values in these instructions are converted to deflection voltages that position the electron beam at the screen location [1]. The screen is coated with a pattern of dots that glows when struck by the electron beam. Each cluster of three dots, one for each colour, is one pixel. Several reports and models have been developed to illustrate the formation of pixels but most of these models are based on different structural characteristics of the image without explicit reference on pixel formation and their corresponding positions. No significant attention has been paid towards determining the sequential positions of active pixels resulting in image formation. The first pionner algorithm aims at determining pixel positions is called Digital Differential Algorithm (DDA). DDA works by sampling the line at unit interval in one coordinate and finding the corresponding values nearest to the path for the other coordinate. More comprehensive algorithm was developed by Jack Bresenham. Bresenham's algorithm claims superiority over DDA by its capability to sense "error" i.e. the difference $d1 - d2$ (in terms of distance) from the true line – which is the deviation of active pixels contributing to image formation on the screen.

VI. REFERENCES

1. NW William, SF Robert, Principles of Interactive Computer Graphics. Principles of interactive computer graphics McGraw Hill 2000.



ISSN(Online): 2320-9801

ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 10, October 2017

2. D Mankanjuola, The Formation of Graphical Images Using Resampling Pixel Addressing Mode and its effects on 12 X 12 Block Memory. *British Journal of Mathematics and Computer Science* 2015; 9: 35-39.
3. W Osberger, AJ Maeder, Automatic Identification of Perceptually Important Regions in an Image. *IEEE International Conference on Pattern Recognition* 1998; 1-4.
4. I Laurent, K Christof, et al. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1998; 20: 1254-1259.
5. S Mahmood Tanveer Fathima, Data and model-driven selection using colour regions. *International Journal of Computer Vision* 1997; 21: 9-36.
6. G Di Vito, V Cesara, et al. Local Operators to Detect Regions of Interest. *Pattern Recognition Letter* 1997; 18: 1077-1081.
7. S Fred, An Estimator for Virtual Attention Through Competitive Novelty with Application to Image Compression. In: *Picture Coding Symposium, Seoul, Korea* 2001; 25-27
8. K Timor, B Michael, Saliency, Scale and Image Description. *International Journal Computer Vision* 2001; 45: 83-105.
9. JZ Wang, J Li, et al. Unsupervised Multi-resolution Segmentation for Images with low Depth of Field. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2001; 23: 85-90.
10. CJ Mark, MC Stein, A Method for Searching Artificial Objects on Planetary Surfaces. *Journal of the British Interplanetary Society* 1990; 43: 209-216.