**DEERWALK INSTITUTE OF TECHNOLOGY**

**Tribhuvan University**

**Faculties of Computer Science**

**A logo of a sea creature

Description automatically generated**

**Bachelors of Science in Computer Science and Information Technology (BSc. CSIT)**

**Course: Computer Graphics (CSC214)**

**Year/Semester: II/III**

**A Lab report on:**

**Implementation of Midpoint Ellipse Algorithm**

Submitted by:

Name: Parth Poudyal

Roll: 1317

Submitted to:

Binod Sitaula

Department of Computer Science

**Theory**

Unlike the Circle drawing algorithm, ellipse does not follow 8-point symmetry. Instead follows 4-point symmetry. Therefore, an incremented approach with a limit for changing regions is applied in the algorithm to print out a ellipse using this algorithm.

This algorithm prioritizes on scanning a ellipse that is centered at the origin; for which the major and the minor axis are parallel to the coordinate system.

The equation of a ellipse is given as :

Like circle

Dividing the curve into 2 parts from (0, b) to (a, 0) into 2 parts at point Q.

The point where the 2 regions are divided : the slope is – 1 i.e.

Slope can be provided by the method of partial derivatives by

We have . We monitor the slope value during the scan conversion process to detect for Q. once the slope is equal to -1, the region is changed. The starting point is (0, b).

Suppose that the coordinates of the last scan converted pixel upon entering step i are We are to select either T or S to be the next pixel. The midpoint of T & S is used to define the following decision parameter.

If < 0 ; the midpoint is inside the curve , this choose pixel T

If > 0 ; the midpoint is outside the curve , this choose pixel S

Decision parameters for the next step is:

Since we have

If T is chosen pixel (pi<0), we have

If S is chosen pixel (pi>0) we have . Thus, we can express in terms of and

if < 0 =

The initial value for the recursive expression can be obtained by evaluating the original

definition of with (0, b):

=

=

Suppose the pixel has just been scan converted upon entering step j. The next pixel is either U or V . The midpoint of the horizontal line connecting U & V is used to define the decision parameter:

If <0, the midpoint is inside the curve, and we choose pixel V.

If 0the midpoint is outside the curve and we choose pixel U. Decision parameter for the

The next step is:

Since ,We have

If V is chosen pixel , we have

If U is chosen pixel we have . Thus, we can express in terms of and ():

: if :

: if

The initial value for the recursive expression is computed using the original definition of .

And the coordinates of of the last pixel chosen for part 1 of the curve:

=

**Algorithm**

Step 1: Take input radius along x axis and y axis and obtain center of ellipse.

Step 2: Initially, we assume ellipse to be centered at origin and the first point as : (x, y 0)= (0, ry).

Step 3: Obtain the initial decision parameter for region 1 as: p10=ry2+1/4rx2-rx 2ry

Step 4: For every xk position in region 1 : If p1k<0 then the next point along the is (xk+1 , yk) and p1k+1=p1k+2ry2xk+1+ry2 Else, the next point is (xk+1, yk-1 ) And p1k+1=p1k+2ry2xk+1 – 2rx2yk+1+ry2

Step 5: Obtain the initial value in region 2 using the last point (x0, y0) of region 1 as: p20=ry2(x0+1/2)2+rx2 (y0-1)2-rx2ry2

Step 6: At each yk in region 2 starting at k =0 perform the following task. If p2k>0 the next point is (xk, yk-1) and p2k+1=p2k-2rx2yk+1+rx2

Step 7: Else, the next point is (xk+1, yk -1) and p2k+1=p2k+2ry2xk+1 -2rx2yk+1+rx2

Step 8: Now obtain the symmetric points in the three quadrants and plot the coordinate value as: x=x+xc, y=y+yc

Step 9: Repeat the steps for region 1 until 2ry2x>=2rx2y

Step 10: End

**Program**

#include <graphics.h>

#include <iostream>

using namespace std;

//By Parth Poudyal

void plotPoints(int xc, int yc, int x, int y) {

    putpixel(xc + x, yc + y, WHITE);

    putpixel(xc - x, yc + y, WHITE);

    putpixel(xc + x, yc - y, WHITE);

    putpixel(xc - x, yc - y, WHITE);

}

void drawEllipse(int xc, int yc, int rx, int ry) {

    int x = 0, y = ry;

    int rx2 = rx \* rx;

    int ry2 = ry \* ry;

    int tworx2 = 2 \* rx2;

    int twory2 = 2 \* ry2;

    // Initial decision parameter of region 1

    int p1 = ry2 - (rx2 \* ry) + (0.25 \* rx2);

    int dx = 0;

    int dy = tworx2 \* y;

    // Region 1

    while (dx < dy) {

        plotPoints(xc, yc, x, y);

        x++;

        dx += twory2;

        if (p1 < 0) {

            p1 += ry2 + dx;

        } else {

            y--;

            dy -= tworx2;

            p1 += ry2 + dx - dy;

        }

    }

    int p2 = (ry2 \* (x + 0.5) \* (x + 0.5)) + (rx2 \* (y - 1) \* (y - 1)) - (rx2 \* ry2);

    // Region 2

    while (y >= 0) {

        plotPoints(xc, yc, x, y);

        y--;

        dy -= tworx2;

        if (p2 > 0) {

            p2 += rx2 - dy;

        } else {

            x++;

            dx += twory2;

            p2 += rx2 - dy + dx;

        }

    }

}

int main() {

    int gd = DETECT, gm;

    initgraph(&gd, &gm, (char\*)"");

    int xc, yc, rx, ry;

    cout << "Enter the center of the ellipse (xc, yc): ";

    cin >> xc >> yc;

    cout << "Enter the x-radius (rx): ";

    cin >> rx;

    cout << "Enter the y-radius (ry): ";

    cin >> ry;

    drawEllipse(xc, yc, rx, ry);

    getch();

    closegraph();

    return 0;

}

**Output**

**A black background with white text

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.**