COMPUTER GRAPHICS

INTRODUCTION

OUTPUT PRIMITIVE

1. Points and lines
2. DDA algorithms
3. Circle Generation
4. Ellipse
5. Filled Area Primitive

Transformation (2D)

2D Viewing

3D Concepts

3D Object representation

Visible – Surface detection methods / hidden object elimination

Computer annotation

Software – Auto Desk, 3d Mars

AREAS OF APPLICATION

1. Visualization
2. Education and Training
3. Entertainment
4. Advertisement
5. C.A.D

Assignment:

Write brief notes on the application areas of computer graphics in the society

Y=mx+b

Starting point (x1, y1)

Next point (x2, y2)

Y=mx+b

M= gradient

Y2-y1 / x2-x1 = m

M is the gradient

Example (3,2), (9,14)

Y=mx+b

M = 14-2/9-3 = 12/6 = 2

Y = 2x + b;

B= y-2x;

y=mx+b

y = 2;

b= 4;

x(k+1) = y(k+m)

DDA, Line drawing algorithm

Digital Differential Analyzer

Brute force line drawing algorithm,

When we have a half pixel, like 0.5, we approximate, go to lower value or upper value

Examples are, 0.5, 1.5, .45

Assumption is that lines are drawn from left to right,

Y(k+1) = x + (1/m)

When moving from right to left, change the sign in between

Y(k+1) = x - (1/m)

Bresenham Line Drawing Algorithm.

1. Input the two line endpoints
2. Load (x0+y0) into the frame buffer, that is plot the first point
3. Calculate constants x, y, 2y, and 2y – 2x and obtain the first value for the decision parameter as P(0) = 2y-x(k), along the line, starting at k=0, perform the following test

If P(k) < 0, the next point to plot is x(k + 1), y(k) and P(k+1) = P(k) + 2y

Otherwise the next point to plot is X(k+1), y(k+1) and P(k+1) = P(k) + 2y – 2x;

For example

1. our initial points (2, 3), and (9, 14) – This is input of the line endpoints
2. 2. Load the points on a frame buffer
3. Then get the constants, (delta x) (change in x), (9-3) = 6;
   1. Change in y (delta y) – change in y was (14 – 2) = 12
   2. 2 delta y = (2 times 12) = 24
   3. P(0) = 24 – 6 = 18
   4. Therefore, our initial decision parameter is 18
   5. This is only applicable if M < 1

2nd assignment,

Derive bressenham line drawing algorithm