



Computação Gráfica

Hitalo Nascimento

Primitivas Gráficas

Aula 03



Objetivos

Apresentar os principais conceitos relacionados a primitivas gráficas;

Apresentar os algoritmos para desenho de retas.



INTRODUÇÃO

- Uma imagem pode ser descrita de várias maneiras;
- Supondo que tenhamos uma tela raster, uma imagem é completamente especificada pelo conjunto de intensidades para as posições de pixels no visor.



INTRODUÇÃO

- Podemos descrever uma imagem como um conjunto de objetos complexos, como árvores e terreno ou móveis e paredes, posicionados em locais de coordenadas especificados dentro da cena.



INTRODUÇÃO

- As formas e cores dos objetos podem ser descritas internamente como matrizes de pixels ou com conjuntos de estruturas geométricas básicas, como segmentos de linha reta e áreas de cores de polígono.



INTRODUÇÃO

- As formas e cores dos objetos podem ser descritas internamente como matrizes de pixels ou com conjuntos de estruturas geométricas básicas, como segmentos de linha reta e áreas de cores de polígono.



INTRODUÇÃO

- Pontos e segmentos de linha reta são os componentes geométricos mais simples das imagens;
- As primitivas de saída adicionais que podem ser usadas para construir uma imagem incluem círculos e outras seções cônicas, superfícies quadriculares, curvas spline e superfícies, áreas de cores de polígono e cordas de caracteres



INTRODUÇÃO

Definimos um ponto em CG, como um pixel com propriedades dadas por:

- Posição e cor.



INTRODUÇÃO

Uma linha conecta dois pontos. Matematicamente é definida como:

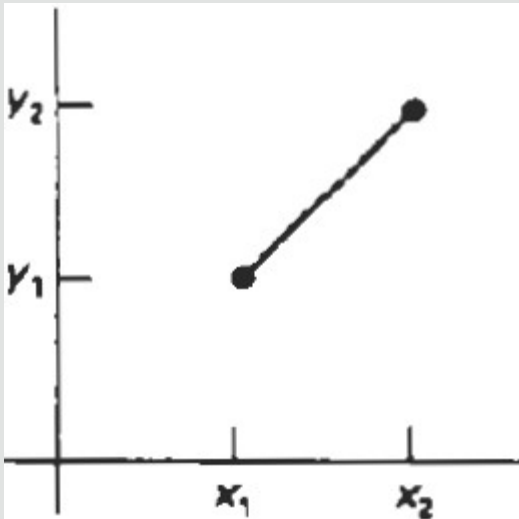
$$y = m \cdot x + b$$

Em que:

- m representa a inclinação da linha ou coeficiente angular;
- b é o ponto de intersecção da reta com o eixo- y ;
- x é a variável independente da função $y = f(x)$.

INTRODUÇÃO

Dados dois pontos definidos como $P(x_1, y_1)$ e $P(x_2, y_2)$, conforme a figura, podemos determinar m e b da seguinte forma:



$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

$$b = y_1 - m \cdot x_1$$

$$\Delta y = m \Delta x$$

$$\Delta x = \frac{\Delta y}{m}$$



INTRODUÇÃO

O algoritmo Digital Differential Analyzer (DDA) é o algoritmo de geração de linha simples baseado nas equações apresentadas.

Step 1 – Get the input of two end points

Step 2 – Calculate the difference between two end points.

Step 3 – Based on the calculated difference in step-2, you need to identify the number of steps to put pixel. If $dx > dy$, then you need more steps in x coordinate; otherwise in y coordinate.

Step 4 – Calculate the increment in x coordinate and y coordinate.

Step 5 – Put the pixel by successfully incrementing x and y coordinates accordingly and complete the drawing of the line.



INTRODUÇÃO

O algoritmo Digital Differential Analyzer (DDA) é o algoritmo de geração de linha simples baseado nas equações apresentadas.

```
#include "device.h"

#define ROUND(a) ((int)(a+0.5))

void lineDDA (int xa, int ya, int xb, int yb)
{
    int dx = xb - xa, dy = yb - ya, steps, k;
    float xIncrement, yIncrement, x = xa, y = ya;

    if (abs (dx) > abs (dy)) steps = abs (dx);
    else steps = abs (dy);
    xIncrement = dx / (float) steps;
    yIncrement = dy / (float) steps;

    setPixel (ROUND(x), ROUND(y));
    for (k=0; k<steps; k++) {
        x += xIncrement;
        y += yIncrement;
        setPixel (ROUND(x), ROUND(y));
    }
}
```



Bibliografia

- https://www.tutorialspoint.com/computer_graphics/computer_graphics_quick_guide.htm
- Donald Hearn and M. Pauline Baker. Computer Graphics C version. Second Ed.