

# COMPUTAÇÃO GRÁFICA E REALIDADE VIRTUAL

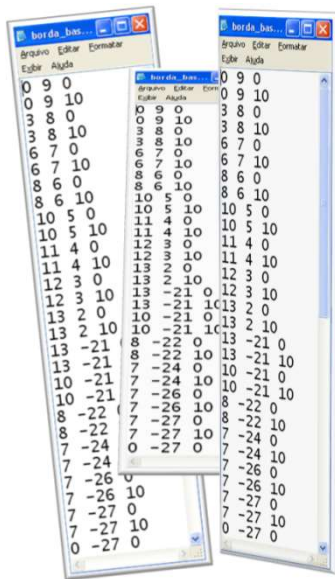
## Origens e Definições

**Prof. Dr. Fernando Kakugawa**

[fernando.kakugawa@animaeducacao.com.br](mailto:fernando.kakugawa@animaeducacao.com.br)

# O que é Computação Gráfica?

- Área da Ciência da Computação que estuda a **geração, manipulação e interpretação** de modelos de objetos e de imagens utilizando o computador
  - É o conjunto de métodos e técnicas para transformar **dados** em **imagens** através de um dispositivo gráfico



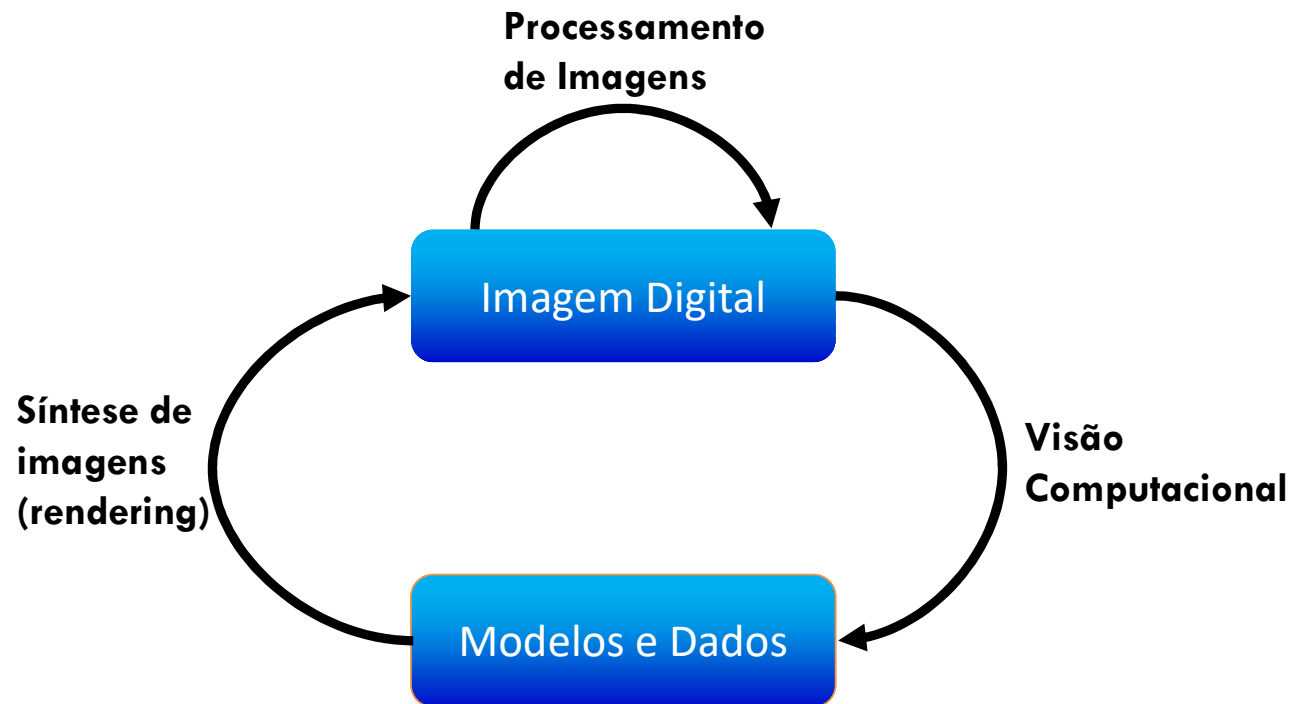
# Subáreas da Computação Gráfica

Síntese de  
Imagens  
(*Rendering*)

Processamento  
de Imagens

Visão  
Computacional

# Subáreas da Computação Gráfica



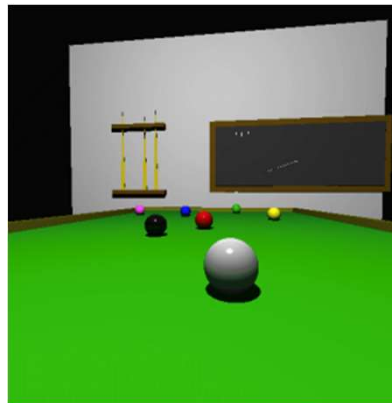
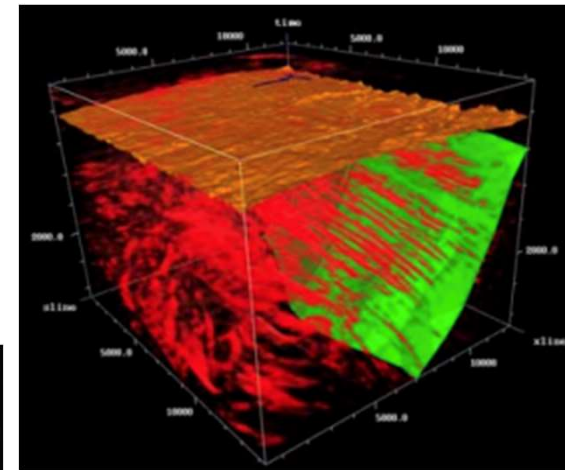


# Síntese de Imagens

- Produz imagens a partir de dados e modelos computacionais de objetos, câmeras e luz

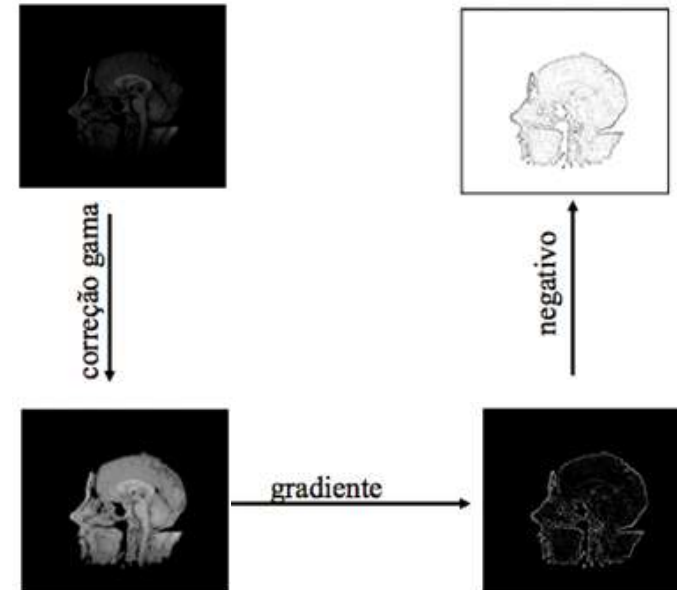


Visualização científica dados sísmicos



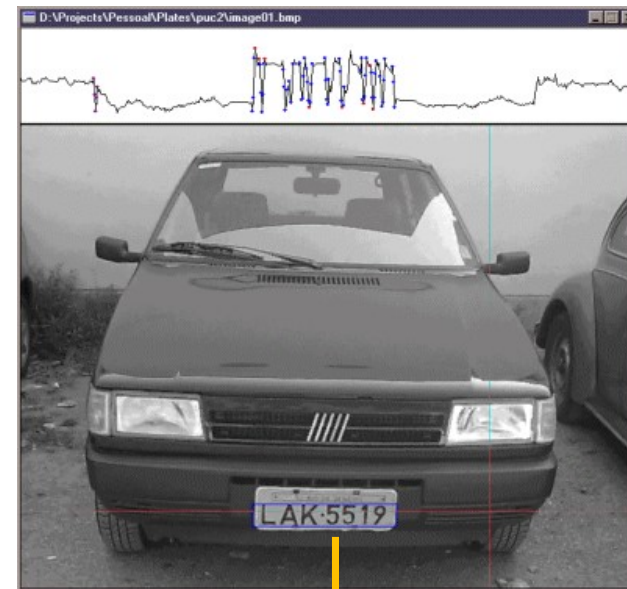
# Processamento de Imagens

- Algoritmos modificam uma imagem para melhorar a qualidade, eliminar ruídos e realçar características
- Parte de uma imagem e como resultado tem-se uma imagem melhor



# Visão Computacional (Análise de Imagens)

- Processa imagens buscando obter informações sobre objetos presentes nela
- Reconhecimento de padrões



LAK 5519



The background of the slide features a blue gradient with diagonal lines of binary code (0s and 1s) and light rays emanating from the left side, creating a digital and technological atmosphere.

# **HISTÓRICO E APLICAÇÕES**



# Histórico

## Euclides

fez uma formulação inicial da geometria

300 – 250 AC

## Descartes

matemático e filósofo que formulou a geometria analítica e os sistemas de coordenadas 2D e 3D

Séc. XVII

## Brunelleschi

arquiteto e escultor que usou de forma criativa a noção de perspectiva visual

Séc. XV

## Sylvester

matemático que inventou as matrizes e a notação matricial, uma das ferramentas mais comuns da Computação Gráfica

Séc. XIX

# Histórico

- **1885:** iniciou-se o desenvolvimento da tecnologia do tubo de raios catódicos

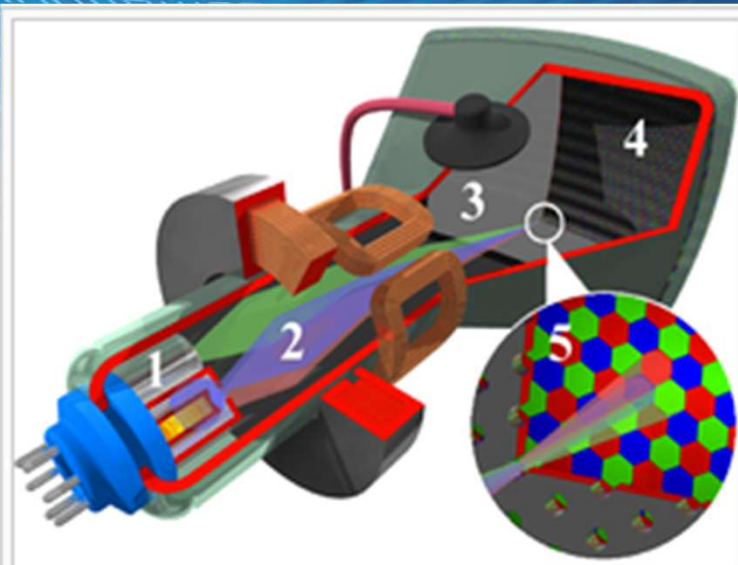
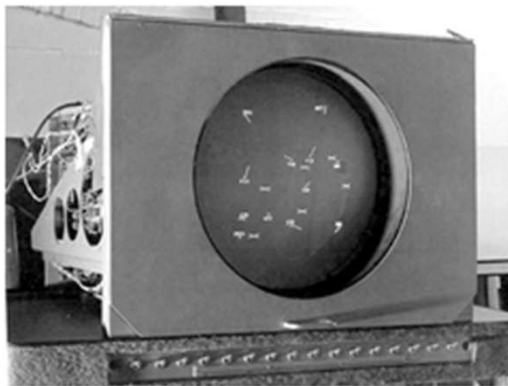


Diagrama em corte de um tubo de raios catódicos **de deflexão eletromagnética**, usado em **televisões** e **monitores** coloridos.

1: Canhões de elétrons e lentes eletrônicas de focalização  
2: Bobinas defletoras (deflexão eletromagnética)  
3: Anodo de alta tensão  
4: Máscara de sombra  
5: Detalhe da matriz de pontos coloridos RGB (vermelho, verde, azul)

# Histórico

- **1950:** Início da Computação Gráfica, pela primeira vez, um tubo de raios catódicos é ligado à um computador, no MIT e Projeto Sage (sistema de monitoramento e controle de voos), onde um radar foi ligado à um computador, num projeto do exército americano, utilizando interação com Light Pen;



SAGE system radar display. Picture used with the permission of the MITRE Corporation (copyright © The MITRE Corporation. All rights reserved).



SAGE operator uses the light-sensing gun.

<https://www.ll.mit.edu/about/History/capecodprototype.html>

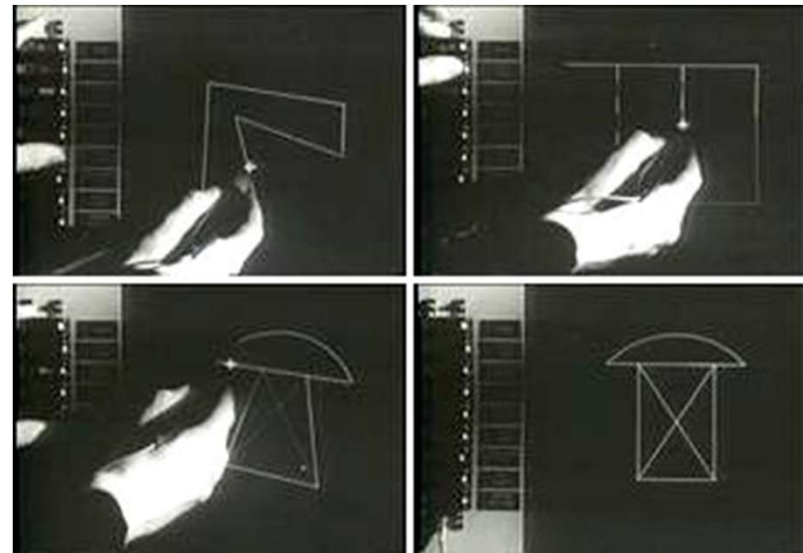


# Histórico

- **1959:** Surge o termo *Computer Graphics* – Criado por Verne L. Hudson, quando ele coordenava um projeto para a Boeing de simulação de fatores humanos em aviões;
- **1962:** No MIT o projeto TX2 com o aluno Ivan Sutherland usa um computador para fazer desenhos em sua tese de doutorado;
  - Sistema de desenho interativo de primitivas gráficas 2D baseado em caneta luminosa

# Histórico

- Sketchpad – (*A Man-Machine Graphical Communication System*)



# Histórico

- **1965:** O precursor dos programas de CAD (*Computer Aided Design*) é desenvolvido pela General Motors



- **1970:** Desenvolvimento de novas técnicas e algoritmos que são utilizados até hoje (métodos de sombreamento e algoritmo de z-buffer)

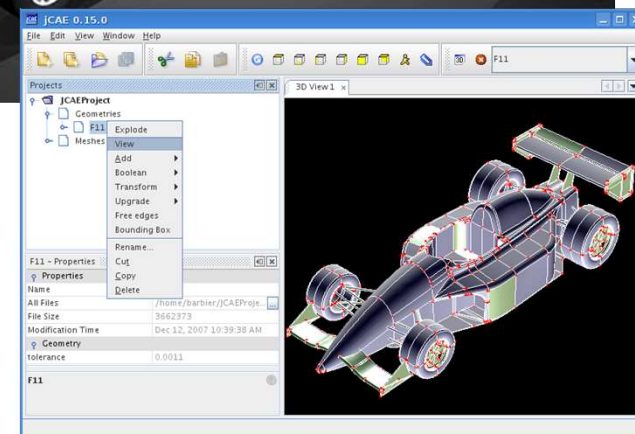
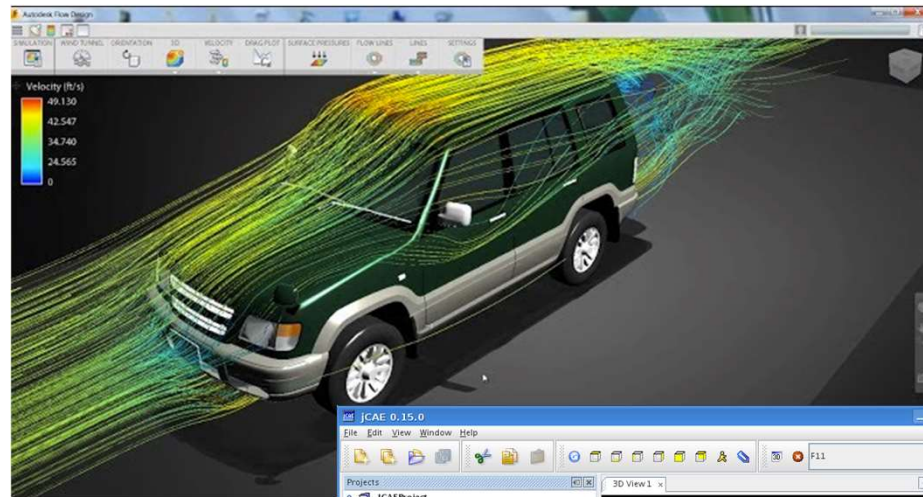
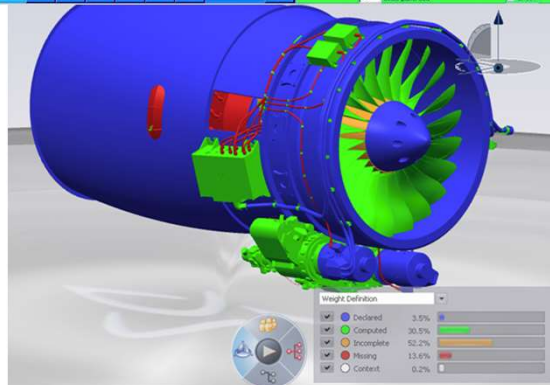
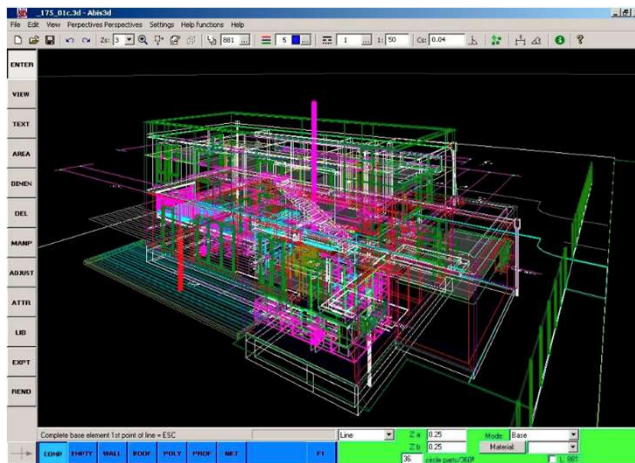


# Aplicações

- Desenho Assistido por Computador (CAD)
- Sistemas de Informações Geográficas (GIS)
- Visualização Científica
- Visualização Médica
- Educação
- Entretenimento

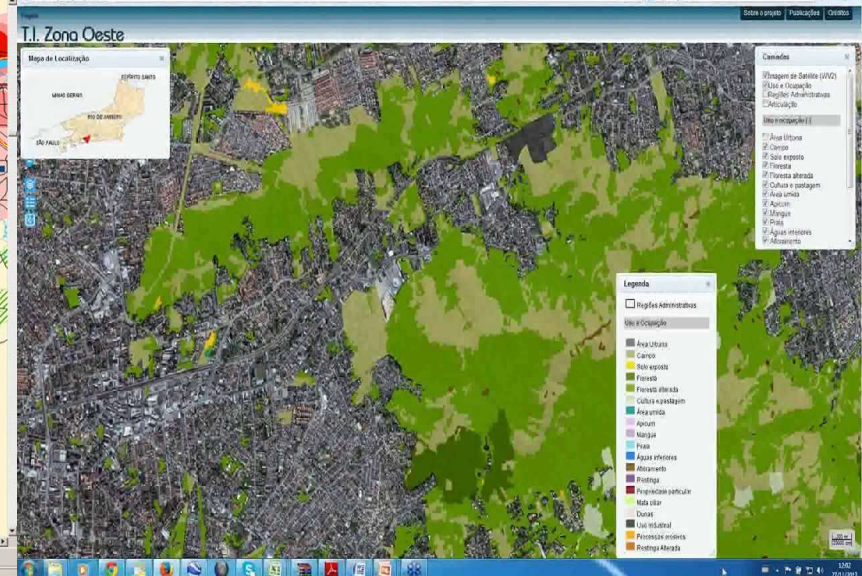
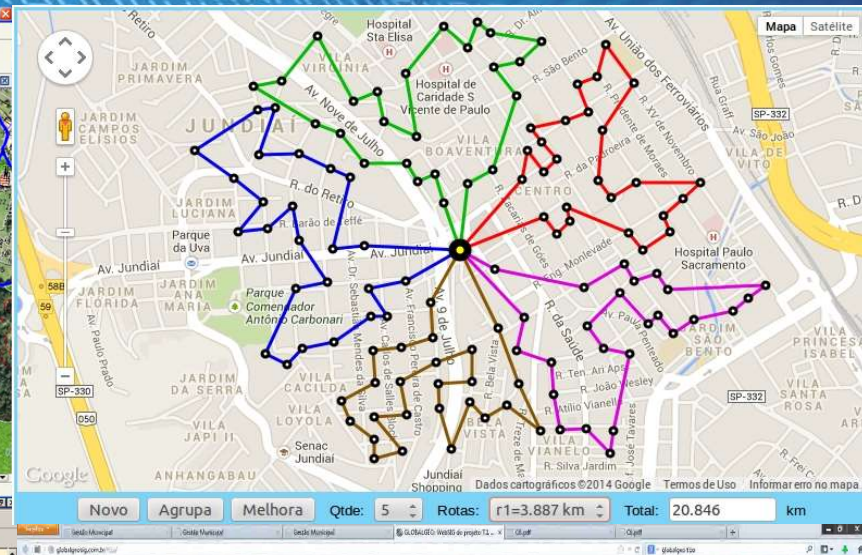
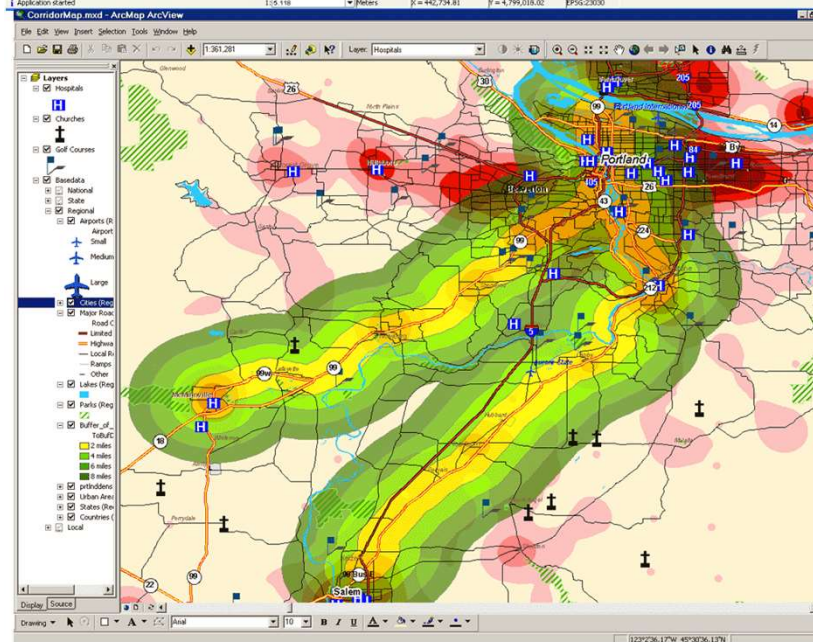
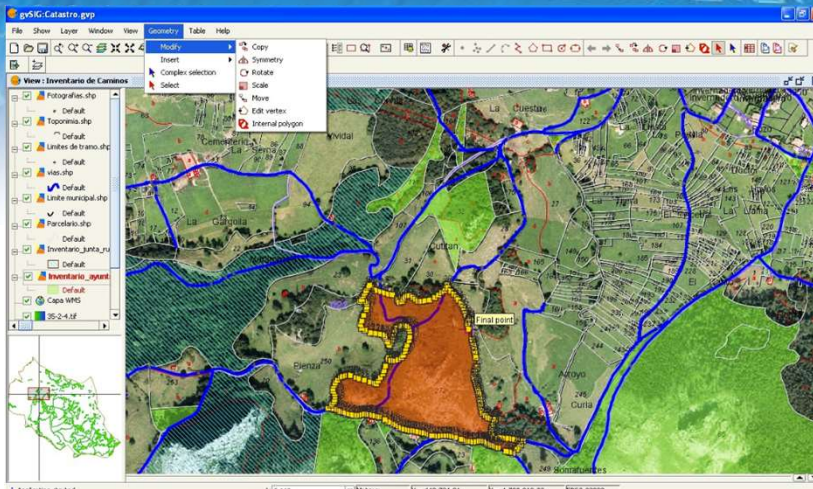
# CAD

- Manipulação de desenho e aplicações industriais, arquitetura, engenharia, construção de modelos virtuais, simulações, etc.



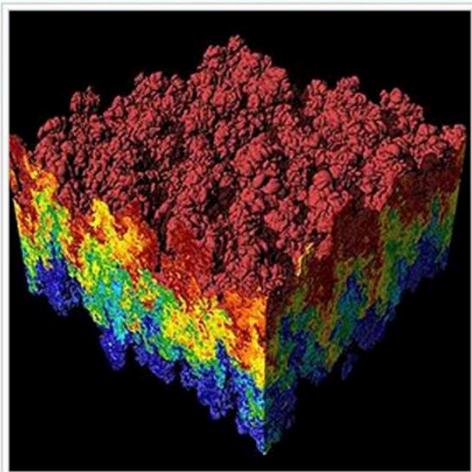


# GIS – Sistemas de Informações Geográficas

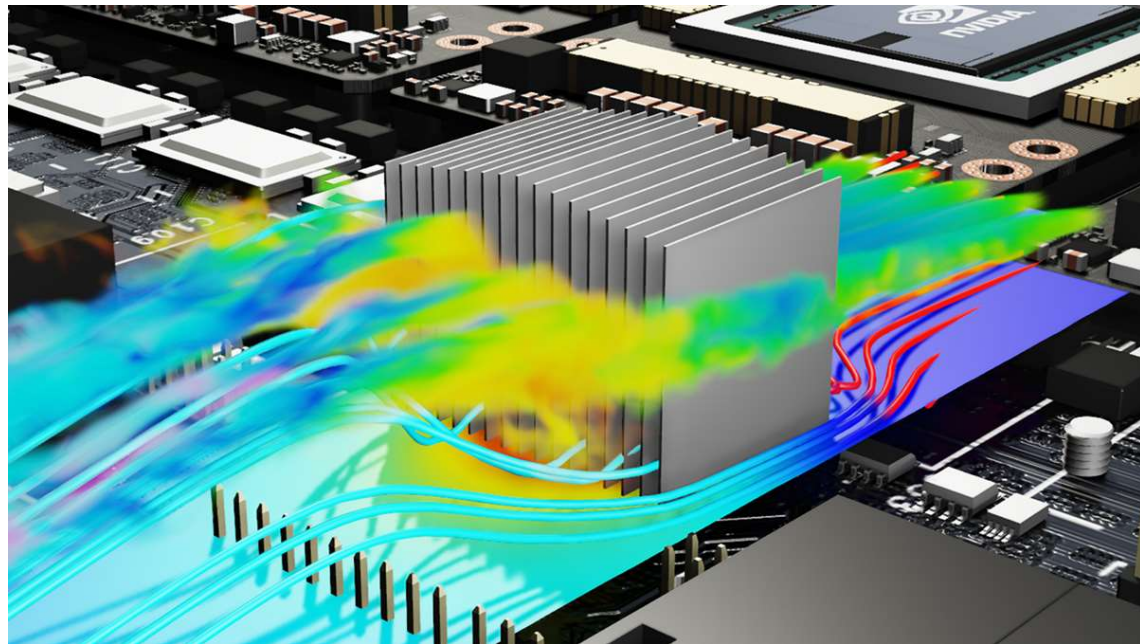




# Visualização Científica



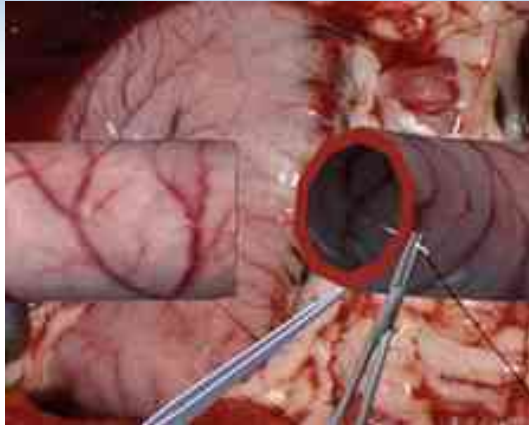
Visualização científica de uma simulação da instabilidade de Rayleigh-Taylor causada pela mistura de dois fluidos<sup>[1]</sup>



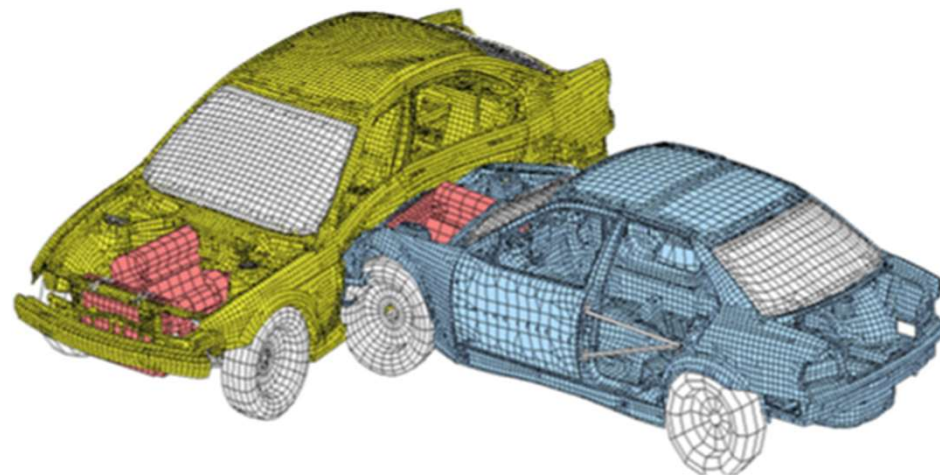
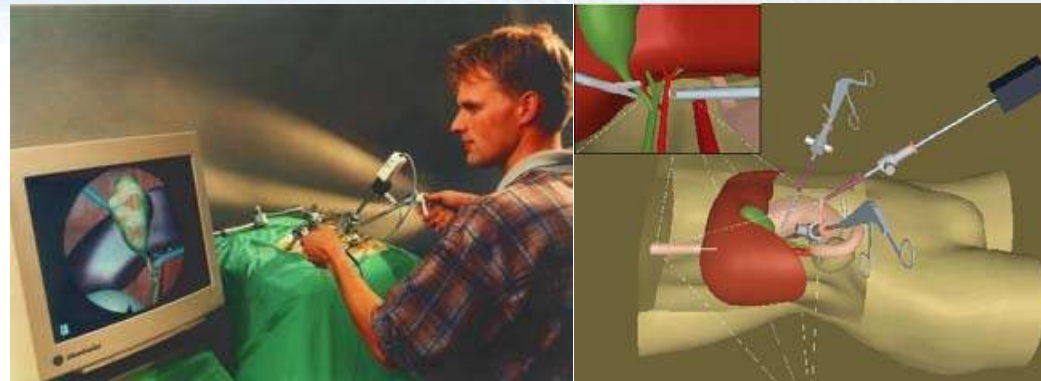


# Treinamento/Simulação

Simulação de sutura de vasos sanguíneos



Simulação de cirurgia endoscópica



# Entretenimento (Games/Filmes)

- Ilusão de uma cena em movimento utilizando sequência de imagens estáticas
- Criação de condições reais utilizando modelos físicos e matemáticos



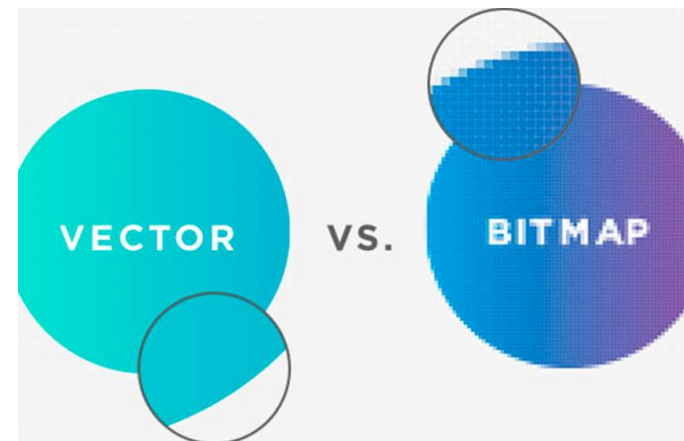
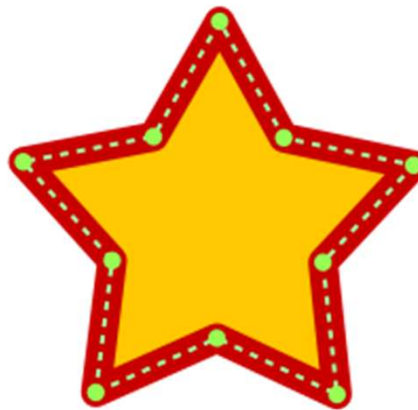
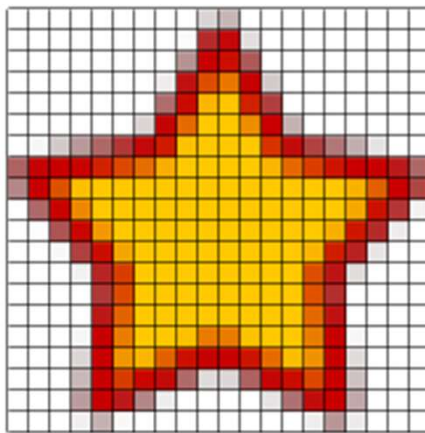
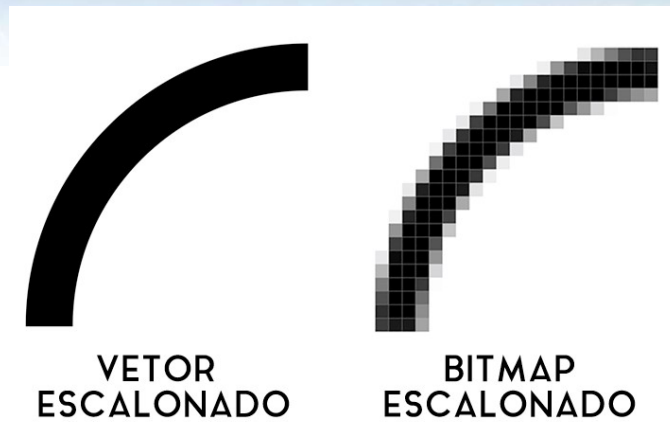
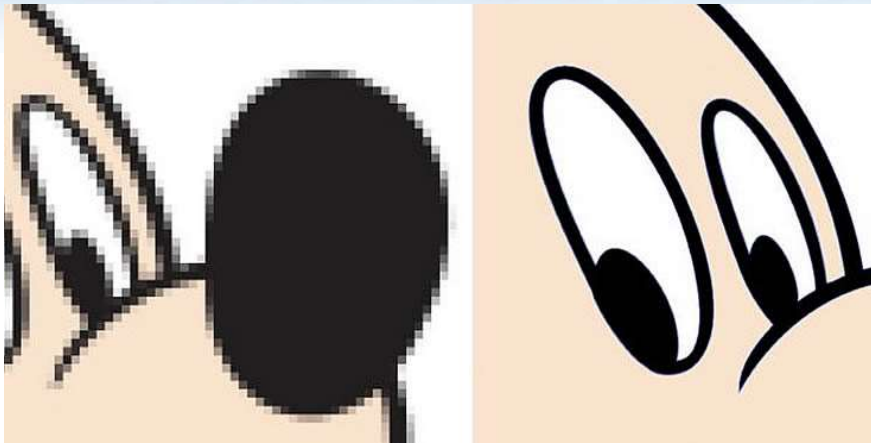
Toy Story  
Marco da Computação  
Gráfica em 3D





# **REPRESENTAÇÃO VETORIAL E MATRICIAL**

# Representação Vetorial e Matricial

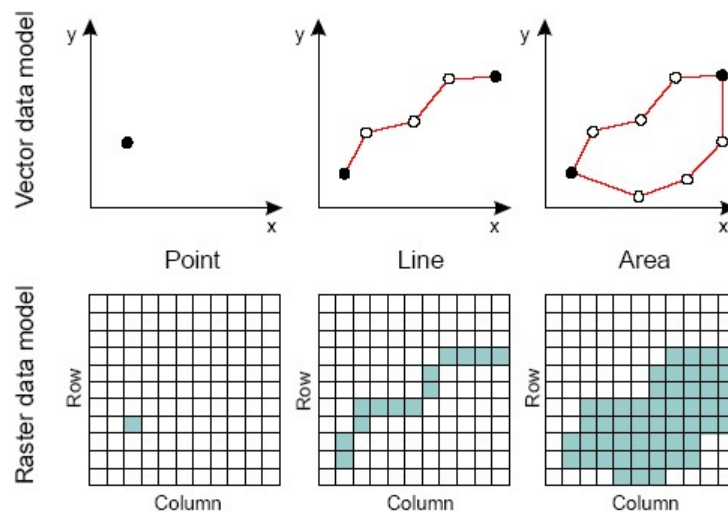
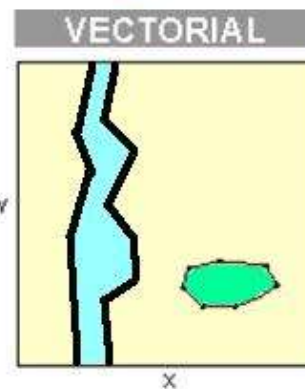
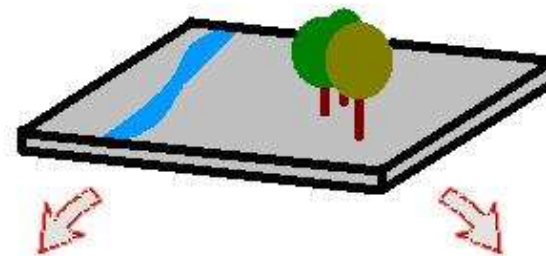
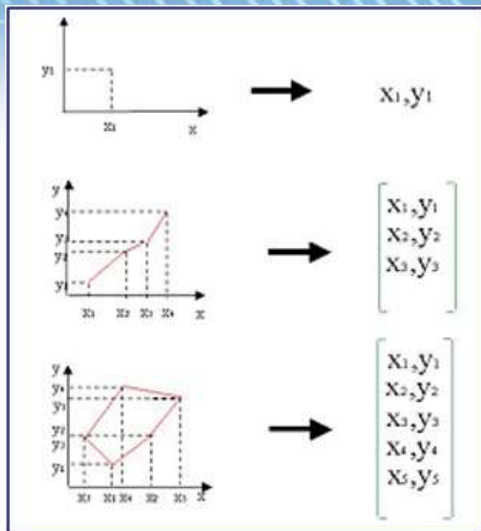


# Representação Vetorial e Matricial

- **Vetorial:** é usada para a definição e modelagens de objetos que serão representados pela imagem;
  - Utilizam como elementos básicos as primitivas:
    - os pontos, as linhas, as curvas, etc.;
  - Cada primitiva possui um conjunto de atributos que define sua aparência e um conjunto de dados que define sua geometria
- **Matricial (Raster / bitmap):** é descrita por um conjunto de células em um arranjo espacial bidimensional, uma matriz;
  - Cada célula representa os pixels da imagem;
  - A representação matricial é usada para formar a imagem na memória e nas telas de computador.



# Representação Vetorial e Matricial



# Rasterização

- É uma aproximação de variáveis contínuas para um espaço discreto
- **Por exemplo:**
  - uma reta **descrita matematicamente** é infinitesimalmente contínua, é impossível determinar qual é o próximo ponto depois de um determinado ponto. Não existem quebras
  - Porém num **espaço discreto** existem quebras, e é possível visualizar cada ponto individualmente
- Portanto, a rasterização de retas é a discretização de um modelo matemático para um espaço de uma matriz quadrada: a tela

The background of the slide features a blue gradient with diagonal lines and a pattern of binary code (0s and 1s) that appears to be receding into the distance, creating a sense of depth and digital technology.

# **RASTERIZAÇÃO DE RETAS**



# Primitivas Gráficas



São comandos e funções que manipulam e alteram os elementos gráficos de uma imagem



Elementos básicos que compõem uma imagem

Pontos, segmentos de reta e círculos

# Pixel ou Ponto

- É uma unidade gráfica fundamental
- **Propriedades:**
  - posição no plano gráfico
  - cor
- Um pixel é um objeto concreto e não abstrato
  - Pode ter uma superfície, que é dada pela construção física do dispositivo no qual está sendo desenhado

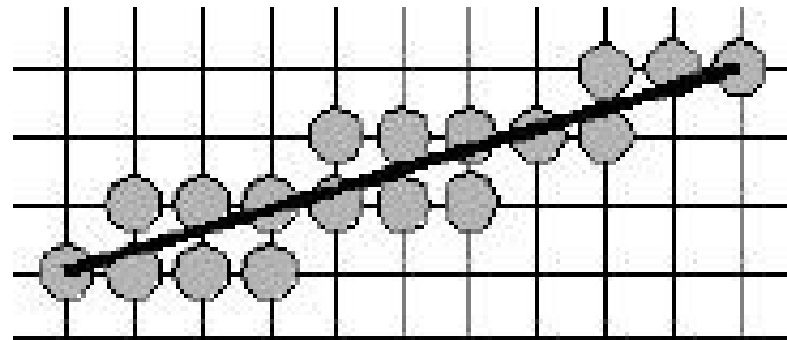
# Rasterização de Retas

- O traçado de primitivas gráficas elementares requer a construção de algoritmos capazes de determinar **quais pixels serão alterados** para simular a aparência do elemento gráfico desejado
  - Denominados: Algoritmos de Conversão Matricial
- Um elemento comum a ser traçado é o segmento de reta
  - Diversas aplicações de computação gráfica têm como primitiva básica o segmento de reta
  - Tais algoritmos estão disponíveis em hardware
  - O hardware recebe as coordenadas dos pontos extremos e calcula os pixels intermediários que farão parte do desenho da reta na tela



# Segmentos de reta

- Adotou-se o critério de selecionar os dois pixels imediatamente acima e abaixo do ponto de intersecção do segmento com cada vertical, menos quando o segmento passa por um pixel

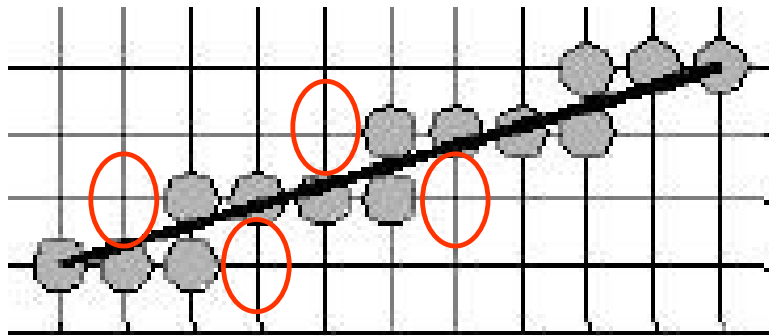


- **Restrição:** dessa forma obtém-se linhas densas, como se o segmento fosse espesso

# Segmentos de reta

- **Para melhorar:**

- O critério foi selecionar todos os pixels cujas coordenadas são obtidas arredondando-se os valores das coordenadas de algum ponto do segmento (ceil, floor, trunc ou round)



- **Restrição:** com segmentos a  $45^\circ$  o critério produz resultados semelhantes ao anterior

# Algoritmo de DDA

## Digital Differential Analyzer

- Estratégia simples: Uso da equação da reta

$$y = mx + B$$

- Onde:  $m = (y_2 - y_1) / (x_2 - x_1)$  //inclinação  
(coeficiente angular)
- Onde:  $B = (y_1 - m * x_1)$  //intersecção eixo y

$$y = y_1 + m * (x - x_1)$$

- Basta verificar qual é o inteiro mais próximo das coordenadas geradas pela equação da reta
- Inicia uma repetição de  $(x_1, y_1)$  até  $(x_2, y_2)$
- Variando o **x** e calculando o valor de **y** no intervalo



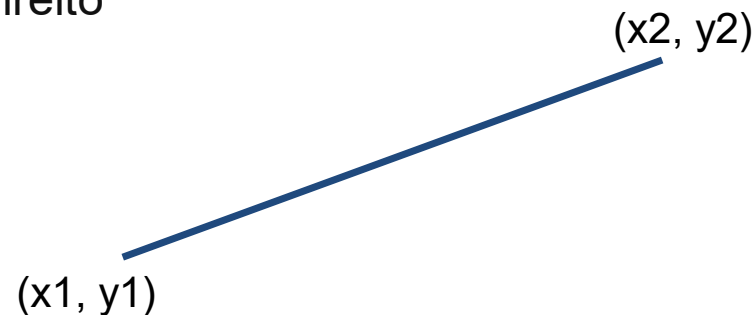
# Algoritmo de DDA

```
void linha(int x1, int x2, int y1, int y2, int cor){  
    float y, m;  
    int x;  
    m = (y2 - y1) / (x2 - x1); //coeficiente angular  
  
    for(x = x1; x <= x2; x++){  
        y = y1 + m * (x - x1);  
        writePixel(x, round(y), cor);  
    }  
}
```

- **Problema:** trabalha com multiplicação e soma de ponto flutuante na determinação dos pontos que compõem a reta, além de arredondamentos
- Os cálculos com ponto flutuante são lentos quando comparado com inteiros

# Algoritmo do Ponto Médio

- Um algoritmo equivalente ao da equação da reta, mas que dispensa as operações em ponto flutuante foi proposto por **Bresenham** (Década de 60)
- **Bresenham** desenvolveu um algoritmo clássico que usa apenas variáveis inteiras, e que permite que o cálculo seja feito de forma **incremental**
  - Assume que a inclinação da linha está entre 0 e 1 (i.e. 1º octante:  $0^\circ - 45^\circ$ )
  - O ponto  $(x_1, y_1)$  é o inferior esquerdo e o ponto  $(x_2, y_2)$  é o superior direito



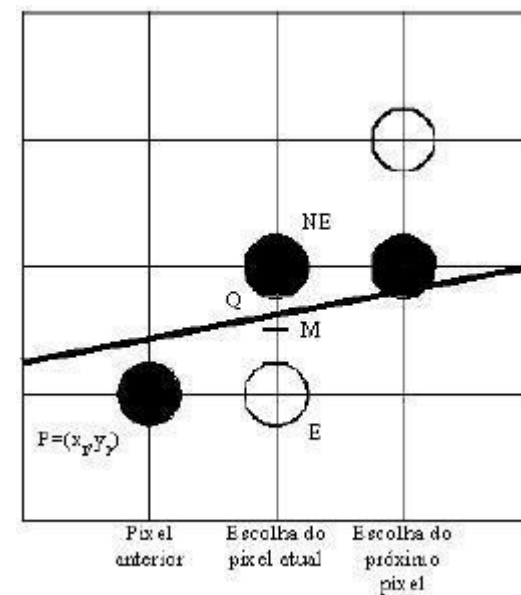
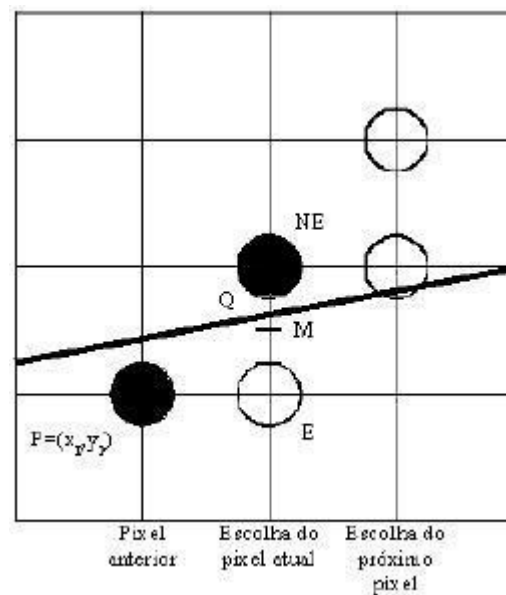
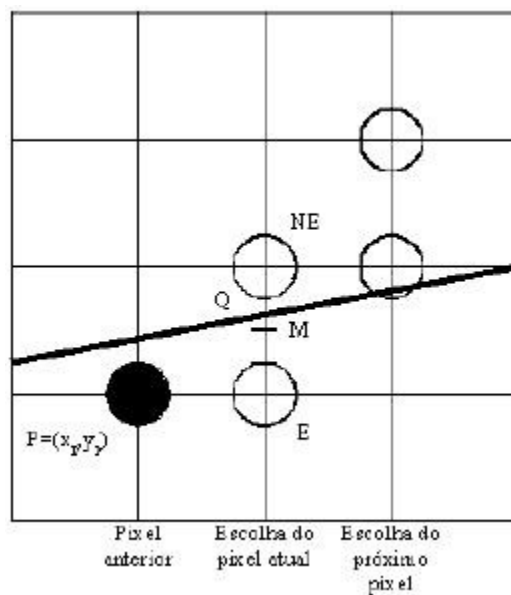
# Algoritmo do Ponto Médio

- É um algoritmo eficiente já que apenas recorre a adições e subtrações inteiras e multiplicações por 2
- A partir do pixel inicial é calculado o ponto médio entre os **próximos dois pixels** a serem selecionados
- **Funcionamento do algoritmo**
  - Assumindo que o pixel que acabou de ser selecionado é **P**, em  $(x_p, y_p)$ , e o próximo deve ser escolhido entre o pixel à direita (pixel E) e o pixel acima à direita (NE)
  - Seja:
    - **Q** o ponto de intersecção entre a reta e a coluna  $x = x_p + 1$  da malha, e
    - **M** o ponto intermediário entre os pixels E e NE



# Algoritmo do Ponto Médio

- O que se faz é observar de que lado da reta está o ponto  $M$ 
  - Se a reta estiver **acima de  $M$** , próximo pixel é **NE**
  - Se a reta estiver **abaixo de  $M$** , o próximo pixel é **E**
  - Dessa forma, o teste do ponto-médio permite a escolha do pixel mais próximo da reta



# Algoritmo do Ponto Médio

```
dx := x2 - x1;  
dy := y2 - y1;  
d := (2 * dy - dx);  
    { Valor inicial da variável de  
      decisão d }  
incE := 2 * dy; {Incremento de E }  
incNE := 2 * (dy - dx);  
    { Incremento de NE }  
x := x1;  
y := y1;  
Desenha_Pixel(x, y);
```

```
while (x < x2) do  
    begin  
        if (d <= 0) then  
            begin  
                { Escolhe E - Q abaixo de M}  
                d := d + incE;  
                x := x + 1;  
            end  
        else begin  
            { Escolhe NE - Q acima de M}  
            d := d + incNE;  
            x := x + 1;  
            y := y + 1;  
        end;  
        Desenha_Pixel(x, y);  
    end;
```

# Exercício

- Aplique o algoritmo para a reta:
  - a)  $P1: (2,1)$  e  $P2: (11,3)$
- Realize o teste de mesa e desenhe o traçado da reta no plano cartesiano



*Material elaborado por:*

**Prof. Ms. Simone de Abreu**

[siabreu@gmail.com](mailto:siabreu@gmail.com)

**Prof. Dr. Fernando Kakugawa**

[fernando.kakugawa@animaeducacao.com.br](mailto:fernando.kakugawa@animaeducacao.com.br)

