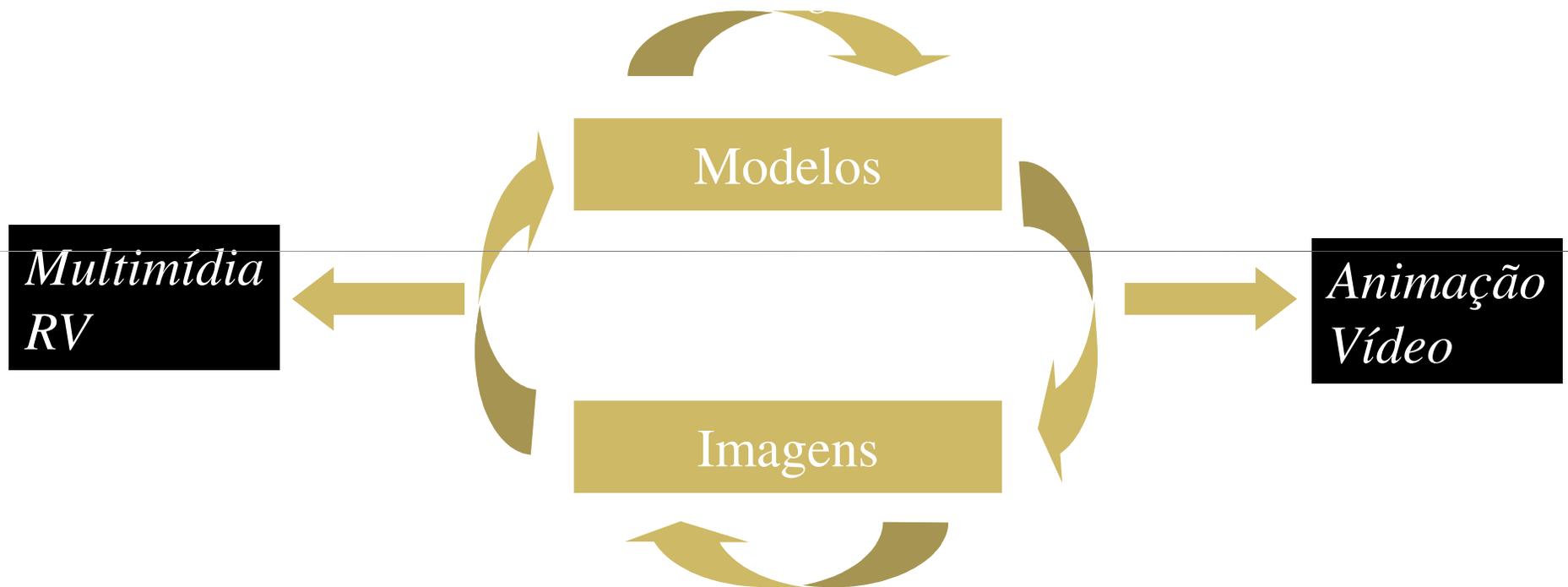


# ***REVISÃO P1***

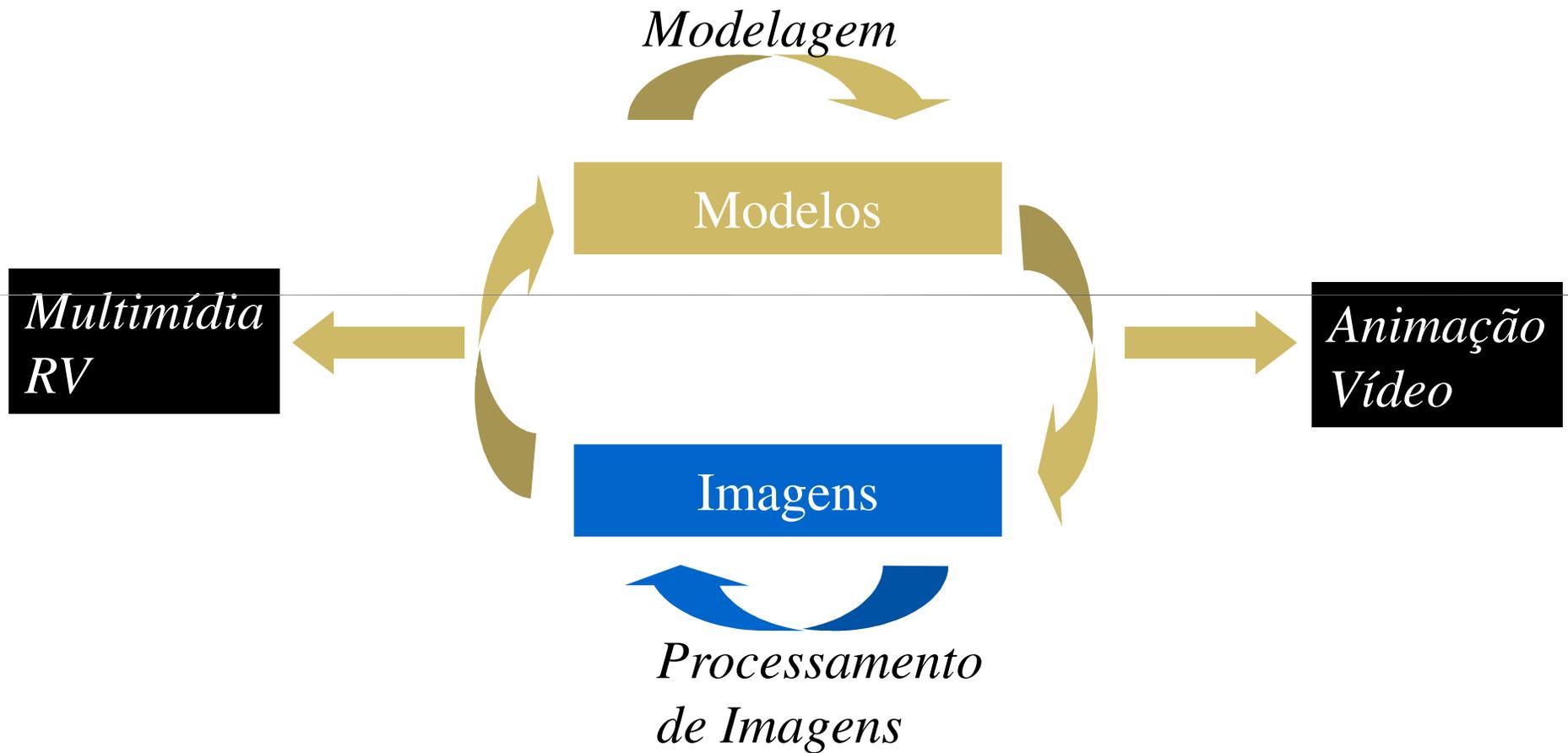
---



# Processamento Gráfico



# Processamento de Imagens



# Compressão de Imagens

**Objetivo:** diminuir o tamanho da imagem para facilitar o armazenamento/transmissão.

- **Compressão sem perda:** imagem reconstruída e idêntica a original. Importante no arquivamento de imagens medicas, ou de satélite.

- **Compressão com perda:** imagem reconstruída apresenta diferenças com relação a original (as vezes imperceptíveis). Uso para imagens em geral (Web, fotografias digitais, etc.)

**Imagem 153 x 204**



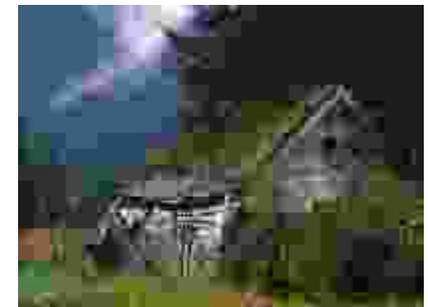
BMP (sem compressão) - 92 KB  
( 153 x 204 x 3 bytes)



JPEG - 6 KB



JPEG - 3 KB



JPEG - 2 KB

# Visão Computacional

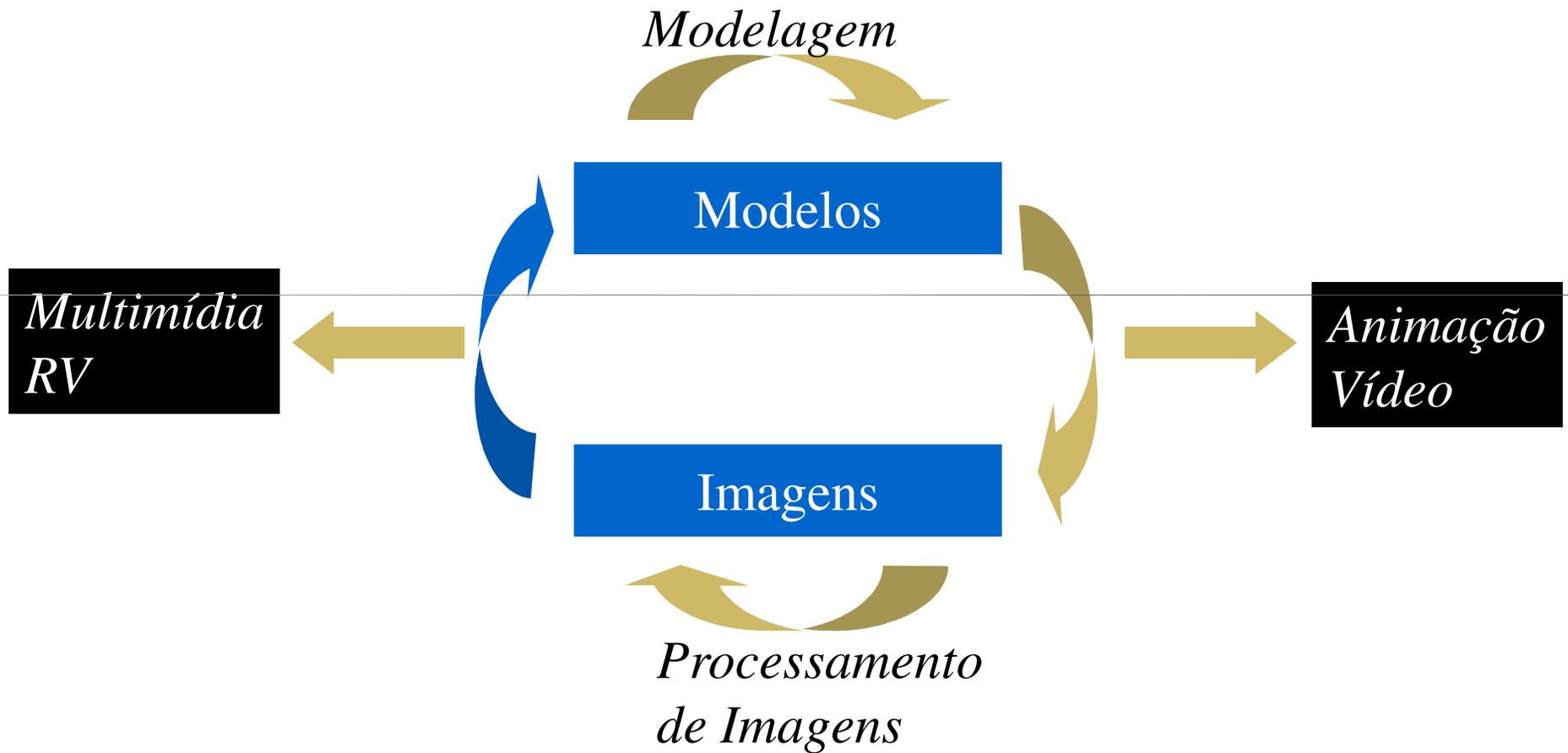


Image denoising and enhancement



Rectangle detection

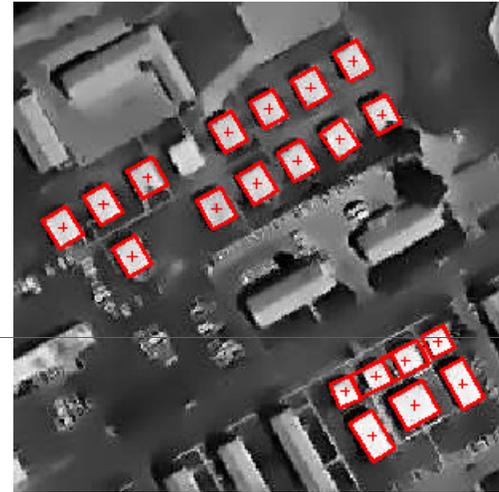
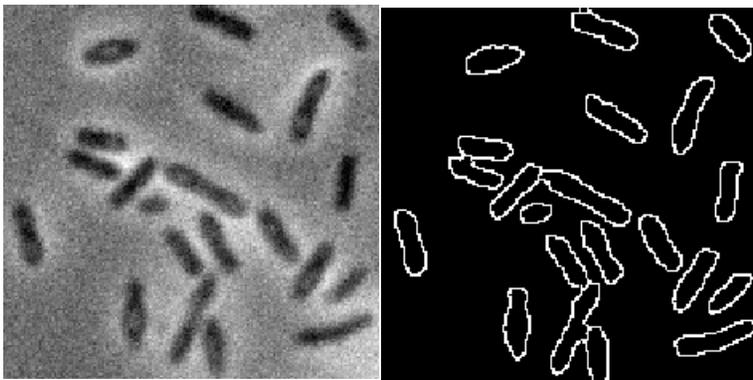


Image Segmentation



People tracking



# O que é Computação Gráfica?

Dados

- Objetos
- Fontes de Luz
- Interação
- ...

IMAGEM



# Temas da Computação Gráfica

## Forma

Modelagem Geométrica

## Aparência

Renderização

---

## Ação

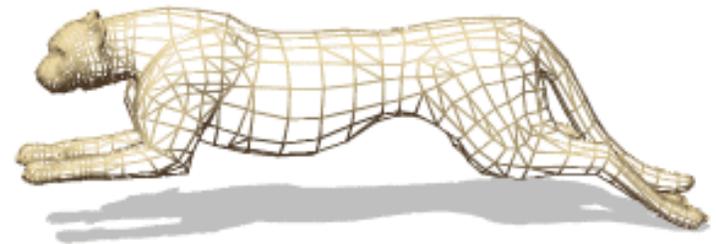
Animação

## Interfaces

RV

# Modelagem Geométrica

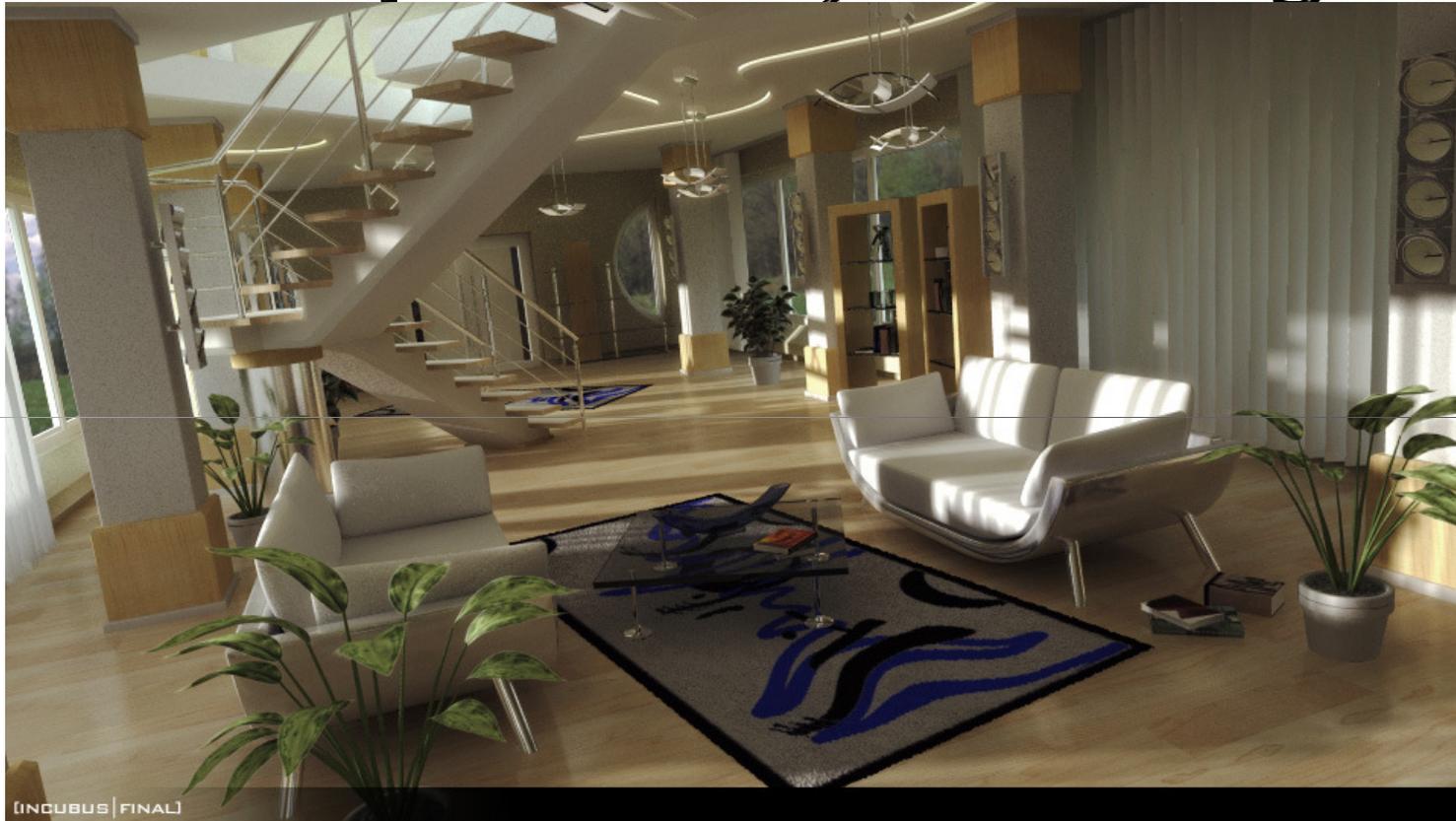
- ❑ Como criar/projetar/representar objetos
- ❑ Como representar coisas e ambientes complexos (um bicho de pelúcia é complexo?)



# Renderização

- ▣ Uma imagem é uma distribuição de energia luminosa num meio bidimensional (o plano do filme fotográfico, por exemplo)
- ▣ Dados uma descrição do ambiente 3D e uma câmera virtual, calcular esta energia em pontos discretos (tirar a fotografia)
- ▣ Resolver equações de transporte de energia luminosa através do ambiente

# Exemplo - Ray Tracing



*Kirschner, Andre*

RENDERER USED: 3d studio max

RENDER TIME: approx 6 hours 30 minutes

HARDWARE USED: AMD1600+, ti4200

# Real ou Computação Gráfica?



<http://www.fakeorreal.com>

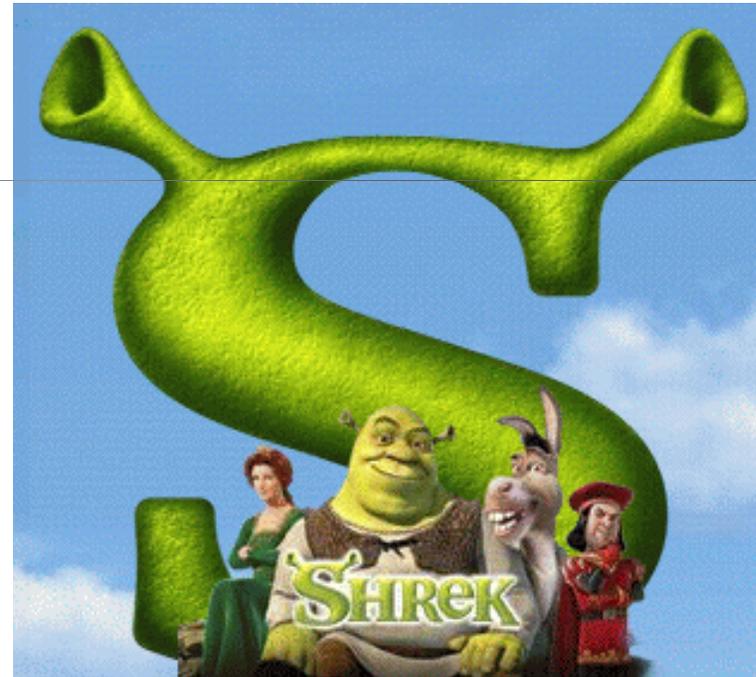
# Animação

- ❑ Modelar Ações dos objetos, ou seja, como objetos se MOVEM
- ❑ Como representar movimento de objetos?
- ❑ Como especificar movimento (interativamente ou através de um programa)?

---

- ❑ Animação Baseada em Física/regras
- ❑ Atores Autônomos
- ❑ Captura de movimento
- ❑ Onde a IA encontra

# Exemplos Monstros, Shrek



**MEDIA | DOWNLOADS**  
GALLERY | VIDEO CLIPS | TRAILERS

1 2 3

**THE STORY** **MEET THE CHARACTERS** **MEDIA AND DOWNLOADS** **FUN AND GAMES** **BEHIND THE FAIRYTALE**

Princess Fiona (CAMERON DIAZ) nervously introduces her new husband Shrek (MIKE MYERS) to her parents, King Harold (JOHN CLEESE) and Queen Lillian (JULIE ANDREWS), the rulers of Far Far Away, in DreamWorks Pictures' computer-animated comedy SHREK 2.

Register

HAPPILY EV... INGREDIENTS

**SHREK 2**  
ON DVD & VIDEO  
FRIDAY NOVEMBER 5TH

Close Book

CLOSE

TM and © 2004 DreamWorks, LLC. All rights reserved.

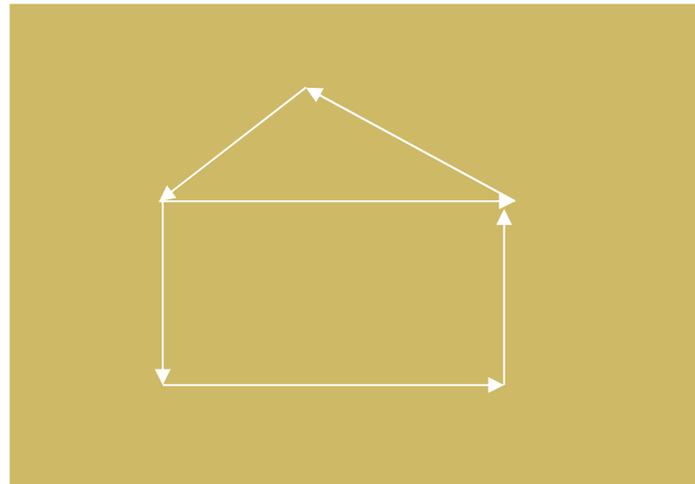
# **Um pouco de história**

---

- Hardware x Software
- Hardcopies devices
- (1963) Ivan Sutherland's PhD tese em sistemas de desenho
  - \* Estruturas de dados para armazenar símbolos e hierarquias que são replicados (projeto de circuitos)
  - \* Técnicas de interação usando teclado e light pen – fundamentando a matemática e os elementos ainda estudados hoje em dia

# Tecnologia de output

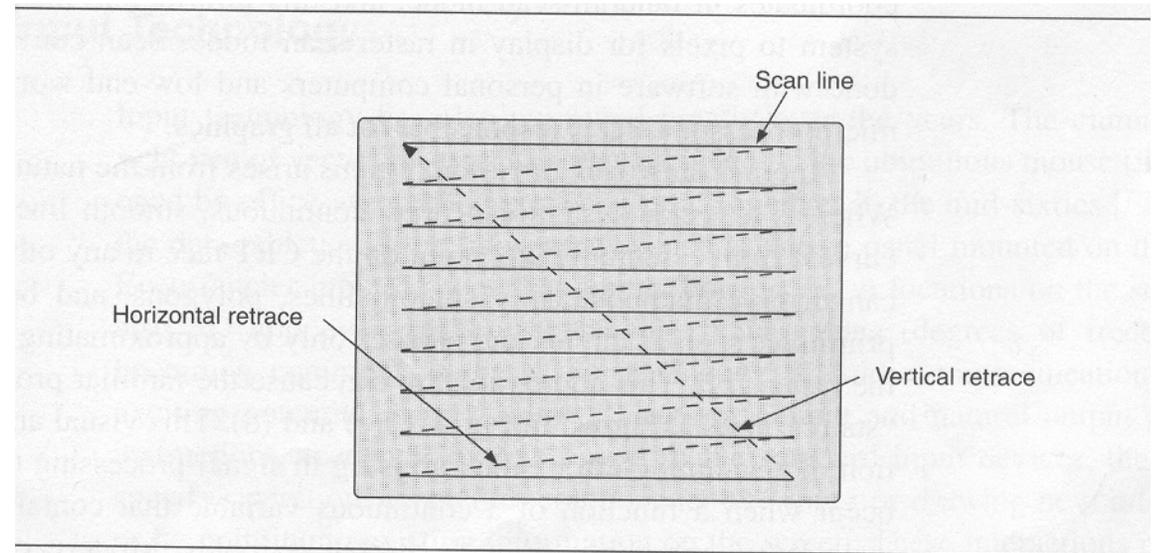
- ▣ Anos 60) Vector systems  
Processador de display (I/O) conectado na CPU  
Um display buffer memory
- 



# Tecnologia de output

- ▣ (Anos 70) Raster systems

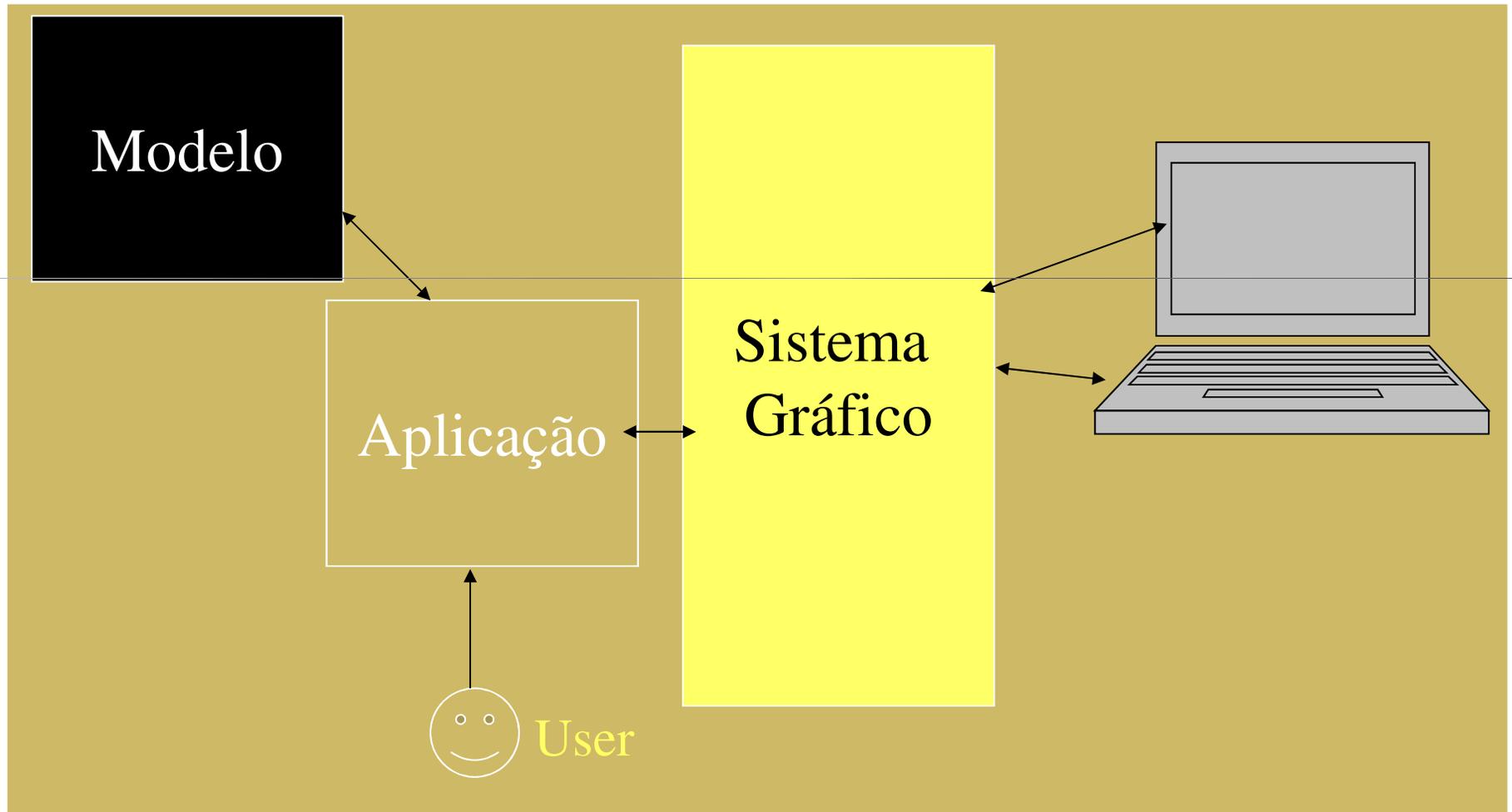
Tecnologia baseada em TV (tecnologia raster onde linhas são traçadas horizontalmente)



# Tecnologia de input

- ❑ 1968 Mouse substituiu o light pen (vector systems)
  - ❑ Tecnologias de captura 2D e 3D
  - ❑ Audio, forcefeedbacks devices, etc
-

# Esquema Conceitual de CG



# DISPOSITIVOS



# Dispositivos



- ❑ Entrada
- ❑ Saída
- ❑ Entrada/Saída (exemplo: *force feedback*)
- ❑ *Olfactory* e *Taste* (em pesquisa)
  - Sensação de olfato e gosto
  - Ainda não se conhece meios efetivos para induzir artificialmente estas sensações

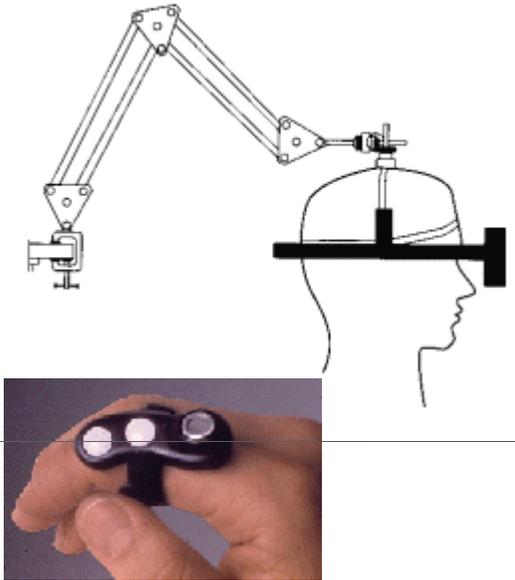
# Escaneamento 3D

- a) Scanner 3D a laser de mão ligado a um braço giratório
- b) Scanner 3D a laser giratório



# Dispositivos

- Tracking devices
  - Rastreadores Mecânicos
  - Rastreadores Ultrassônicos



# Motion Capture

- ❑ O processo de captura tem duas fases. Na primeira fase as posições espaciais dos sensores são capturadas e usadas para deduzir as posições das juntas do esqueleto do ator. Esta fase é observada e dirigida em tempo-real pelo diretor. Na segunda fase o esqueleto é revestido de volume e processos de rendering são aplicados.

# MOCAP



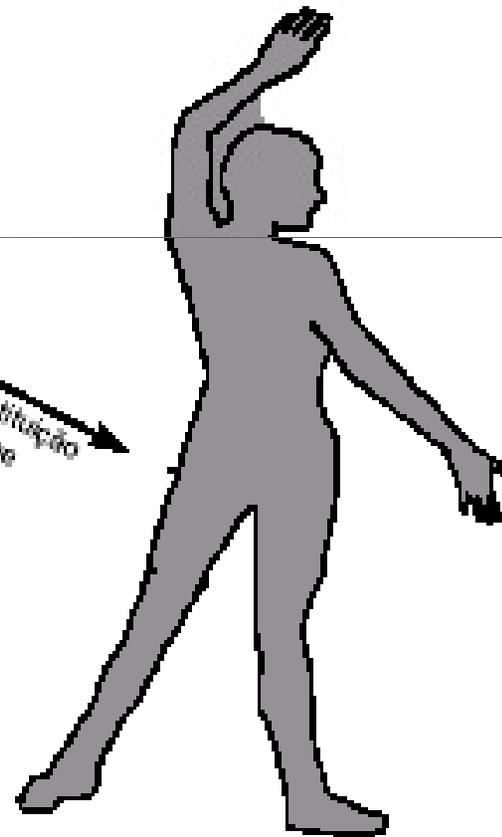
Captura

*Dedução do Esqueleto*



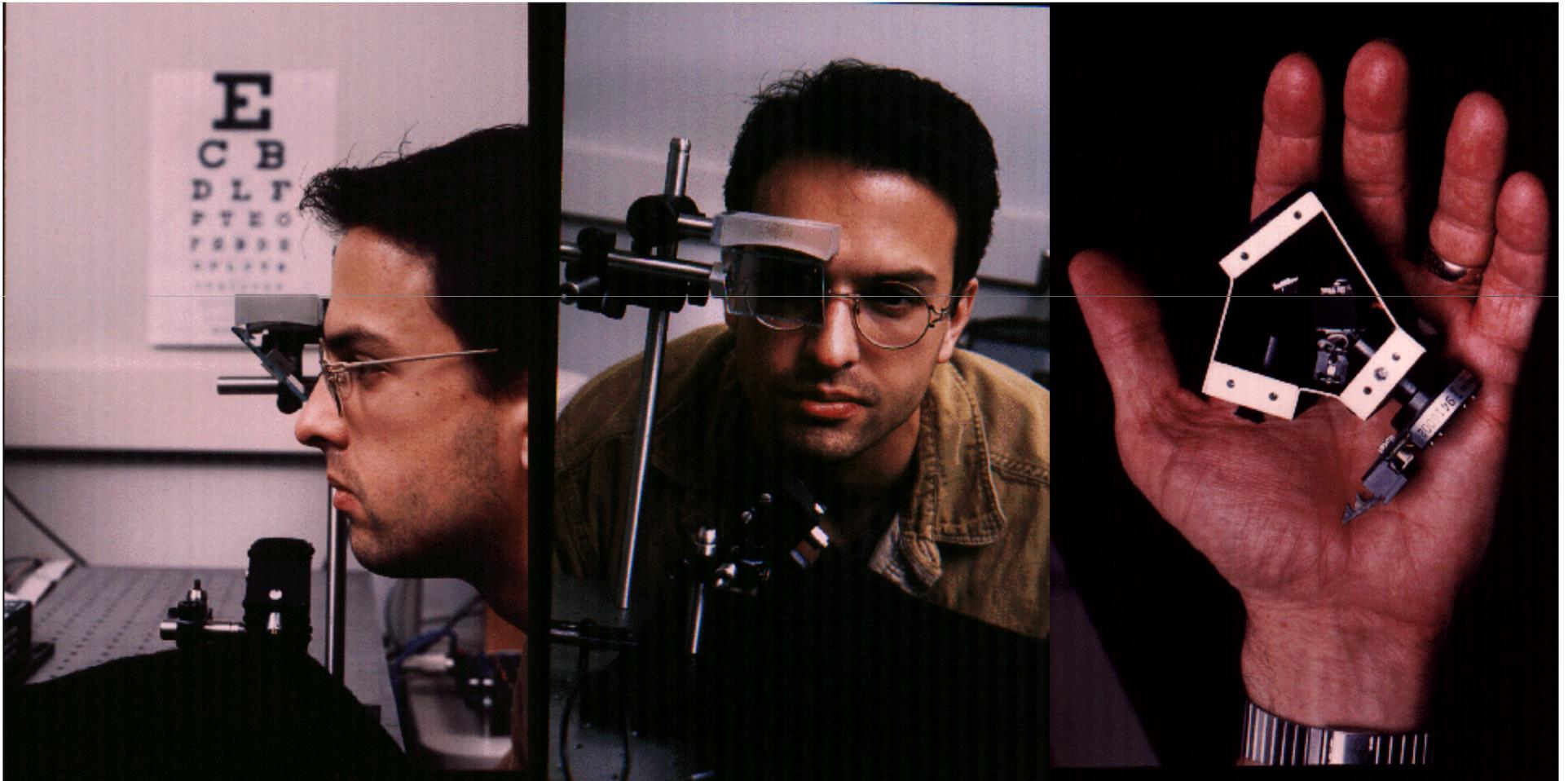
Esqueleto

*Reconstituição do Volume*



Personagem Virtual

# Dispositivos de RV



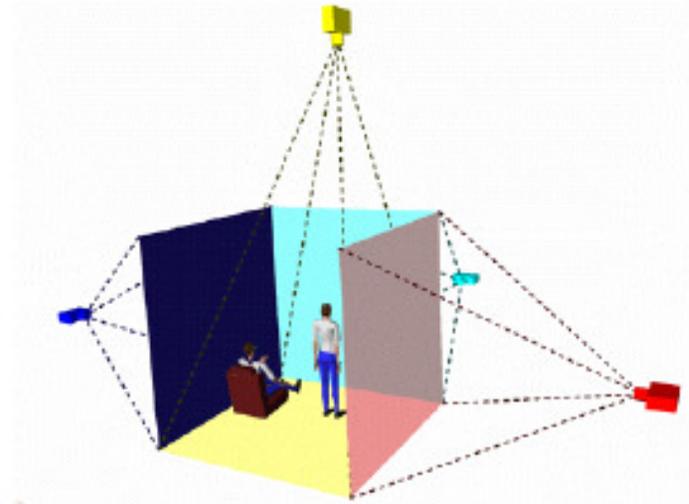
# *Head Mounted Displays – HMD*

- ❑ Duas pequenas telas (CRT x LCD)
  - ❑ Imagens mono ou estéreo
  - ❑ **Pode** incluir fone de ouvido
  - ❑ **Pode** vir com um rastreador de posição para capturar o movimento da cabeça
- 



# CAVE

- ▣ Sala com projeções nas paredes (a projeção vem do lado de fora)



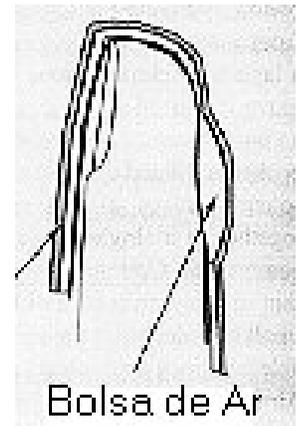
# Dispositivos de E/S: Geradores de Sensação de Tato e de Força

- Haptic devices:

  - touch feedback (tato: textura, geometria da superfície, temperatura)

---

  - force feedback (peso, maciez ou dureza da superfície)



# MATEMÁTICA PARA CG

---

# Nesta aula nós vimos..

- ▣ Introdução
  - ▣ Revisão Matemática
    - Vetores
    - Matrizes
-

# **Algoritmos de rasterização para primitivas 2D**

---

**Objetivo:**

**Aproximar primitivas  
matemáticas descritas  
através de vértices por  
meio de um conjunto de  
pixels de determinada cor**

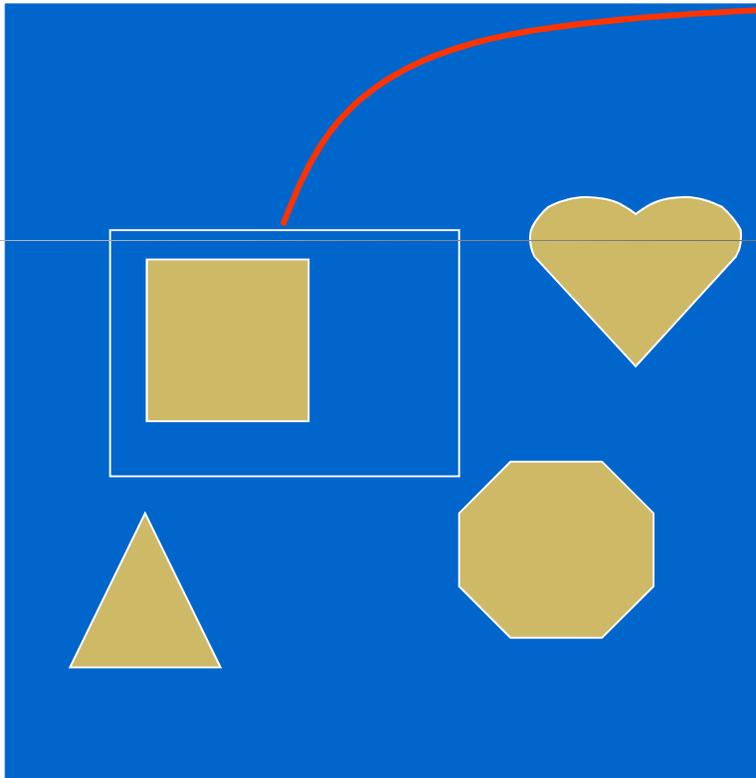
---

Modelo 2D

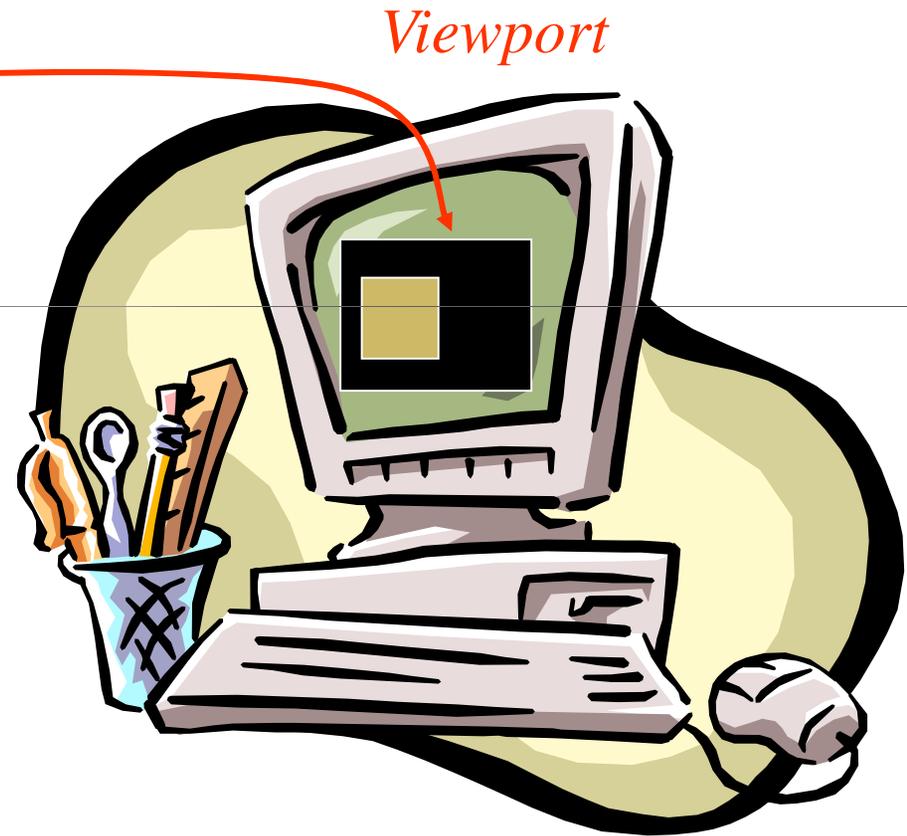
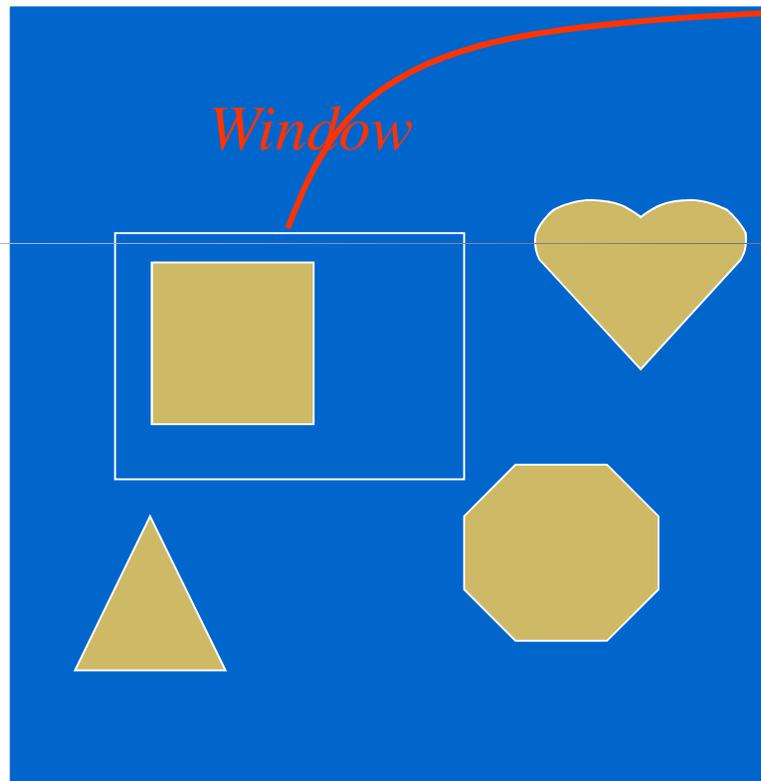
Display



# Pipeline de visualização 2D



# Pipeline de visualização 2D



# SR 's 2D

- ▣ SRO

- ▣ SRU

- ▣ SRW

(recorte 2D)

---

- ▣ SRV

- ▣ SRD

# Pipeline de visualização 2D

$$\text{Window}_{vi} \text{SRU} = (1,7)$$

$$\text{Window}_{vf} \text{SRU} = (5,11)$$

$$T_{\text{SRU-SRW}=0} - \text{Window}_{vi} \text{SRW}$$

$$T_{\text{SRU-SRW}x} = -1$$

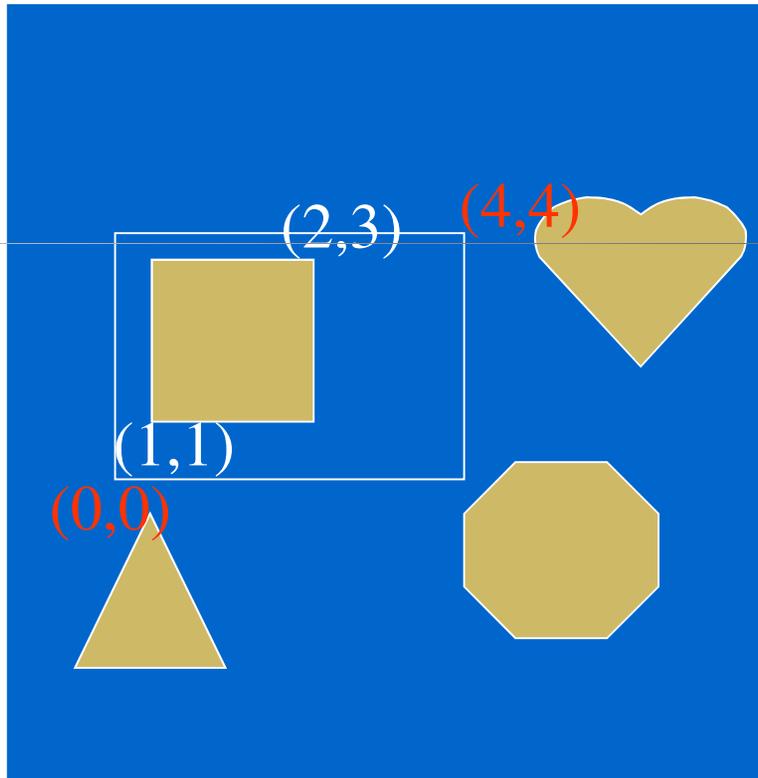
$$T_{\text{SRU-SRW}y} = -7$$

$$\text{Window}_{vi} \text{SRW} = (0,0)$$

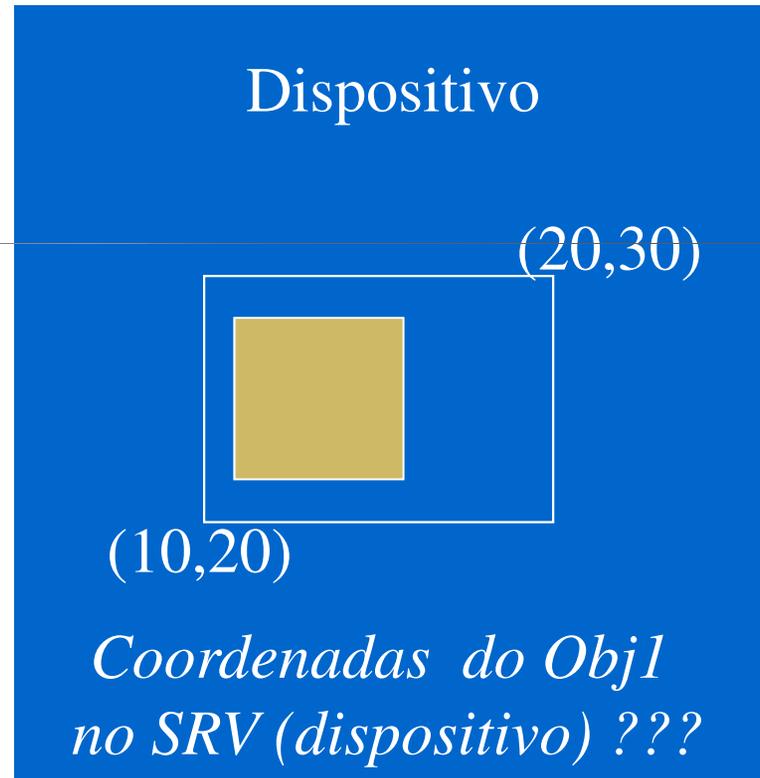
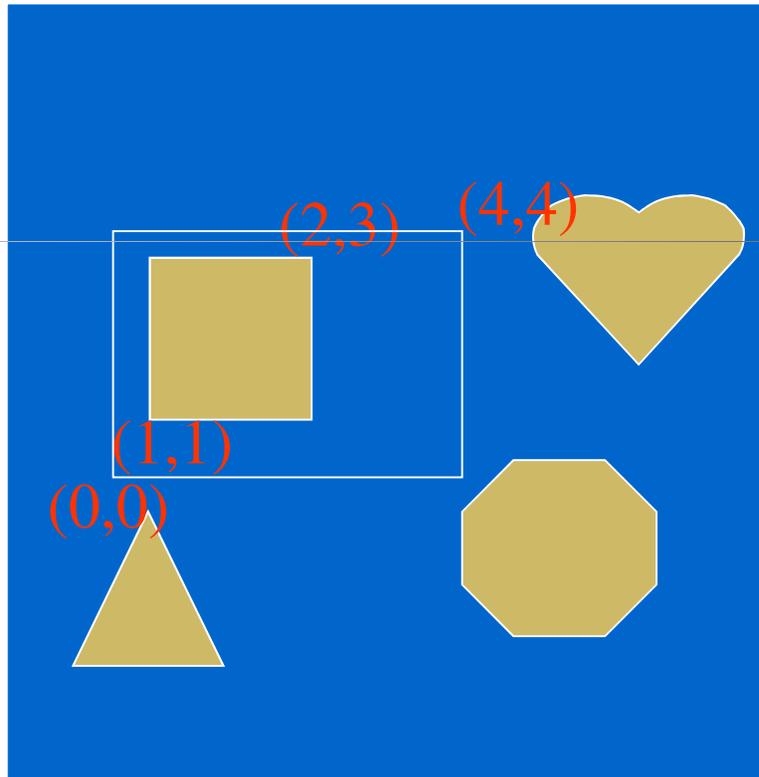
$$\text{Window}_{vf} \text{SRW} = (4,4)$$

$$\text{Obj1}_{vi} \text{Window} = (1,1)$$

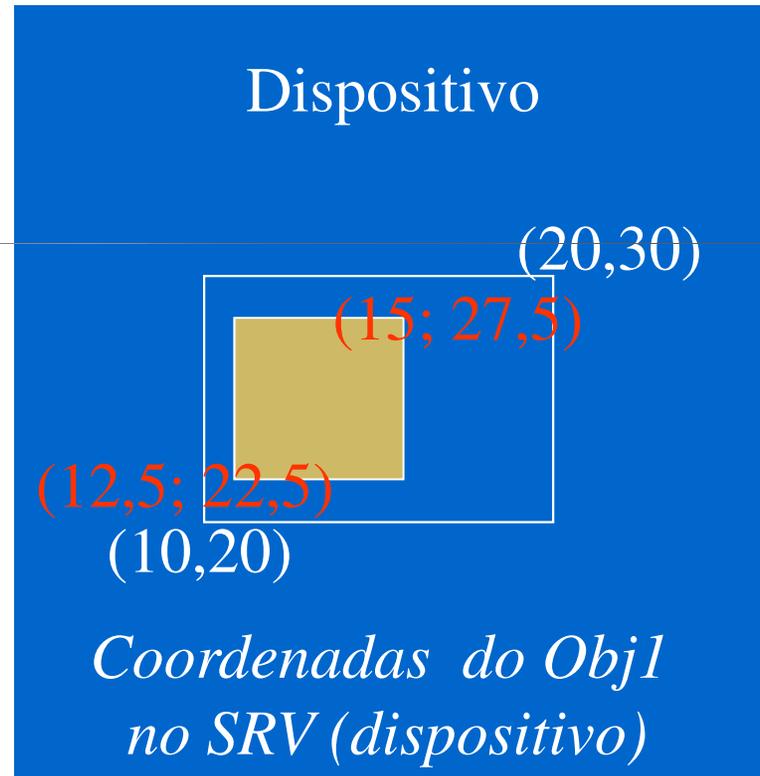
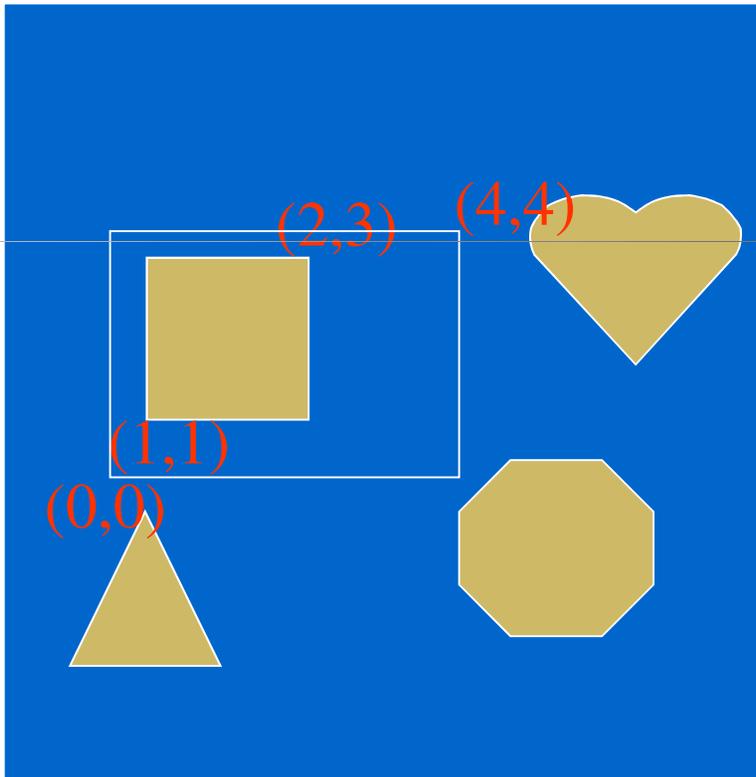
$$\text{Obj1}_{vf} \text{Window} = (2,3)$$



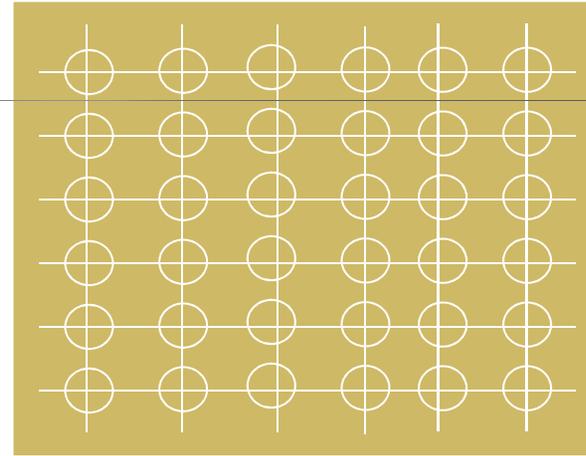
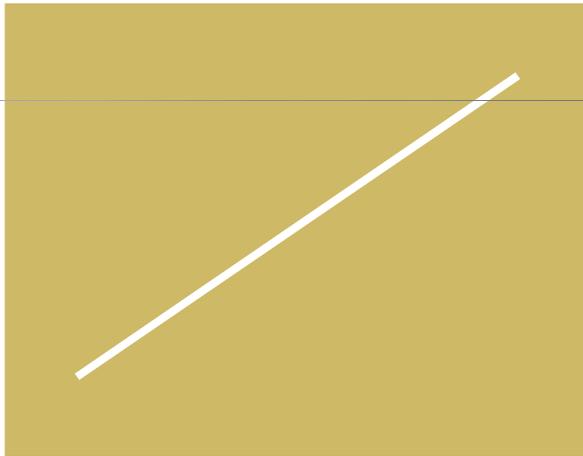
# Pipeline de visualização 2D



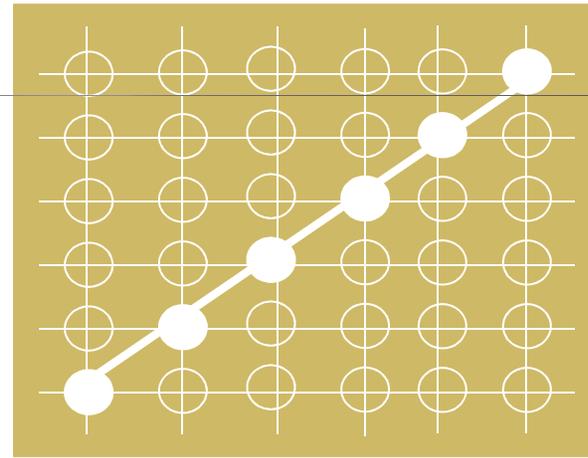
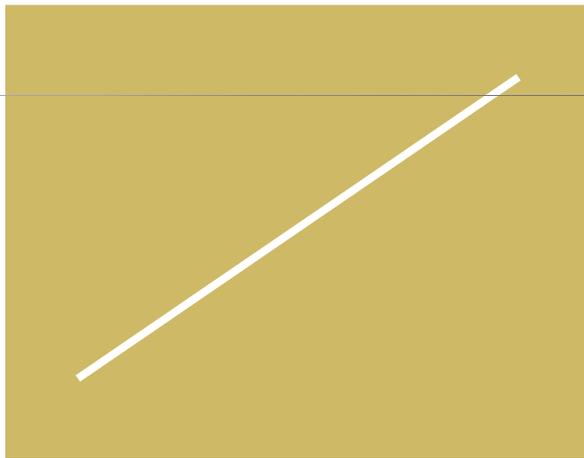
# Mapeamento:



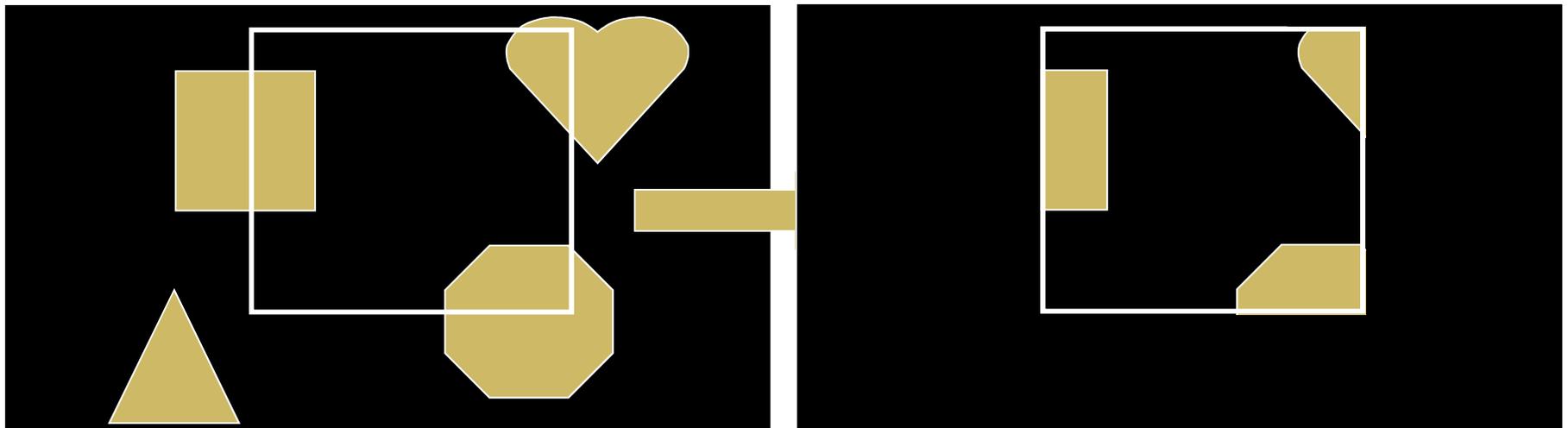
# Algoritmos de rasterização



# Algoritmos de rasterização



# Algoritmos de rasterização



# Algoritmos de varredura

- ▣ Bresenham – Algoritmo do ponto médio
    - Atrativo porque usa somente operações aritméticas (não usa round ou floor)
    - É incremental
-

# Algoritmos de varredura

- ▣ Algoritmos para preenchimento. Dois problemas:

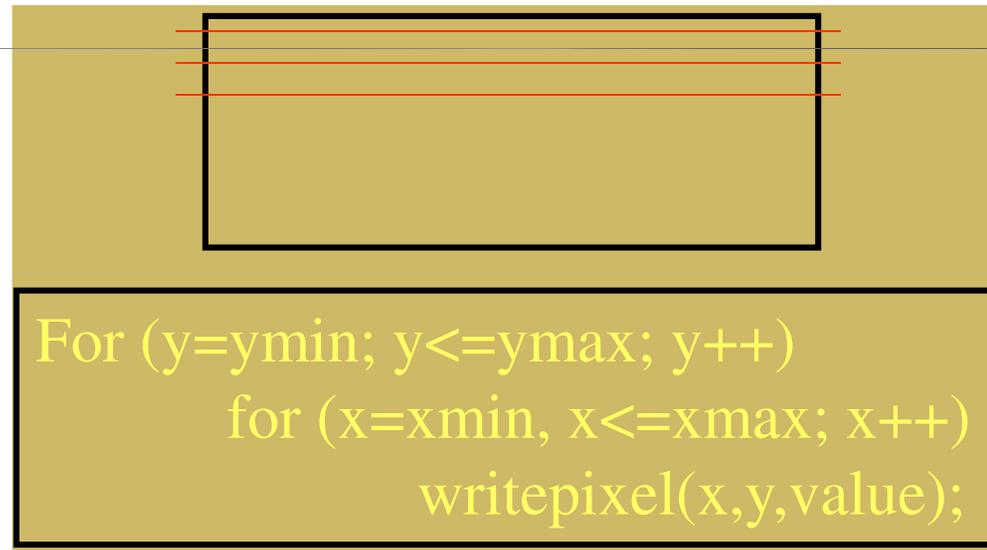
Decisão de qual pixel preencher

Decisão de qual valor preencher

---

# Algoritmos de varredura

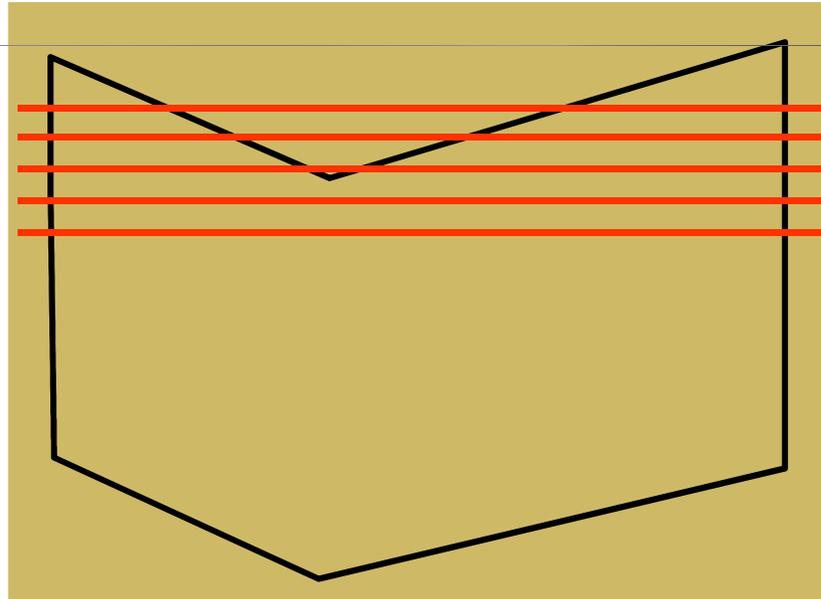
- Algoritmos para preenchimento: Retângulo



```
For (y=ymin; y<=ymax; y++)  
    for (x=xmin, x<=xmax; x++)  
        writepixel(x,y,value);
```

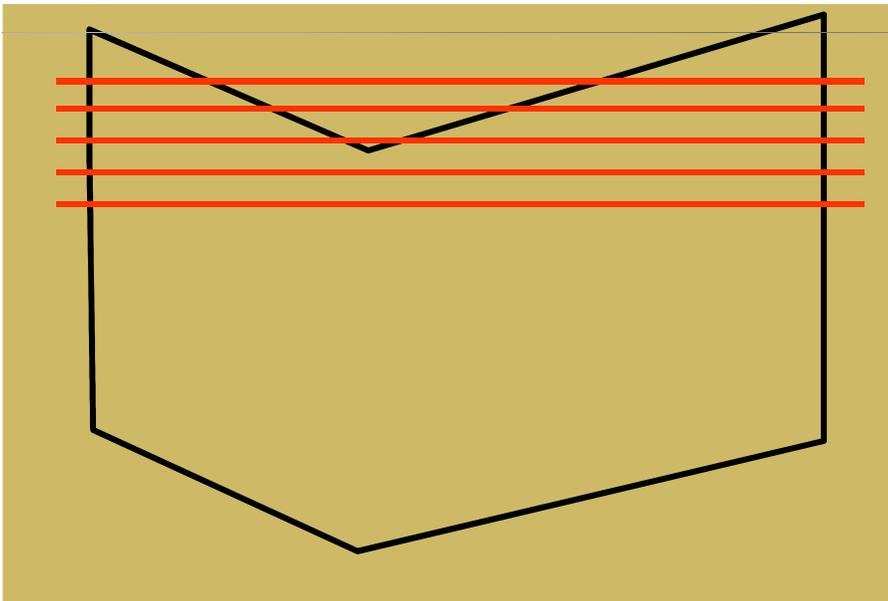
# Algoritmos de varredura

- Algoritmos para preenchimento: Polígonos convexos ou não



# Algoritmos de varredura

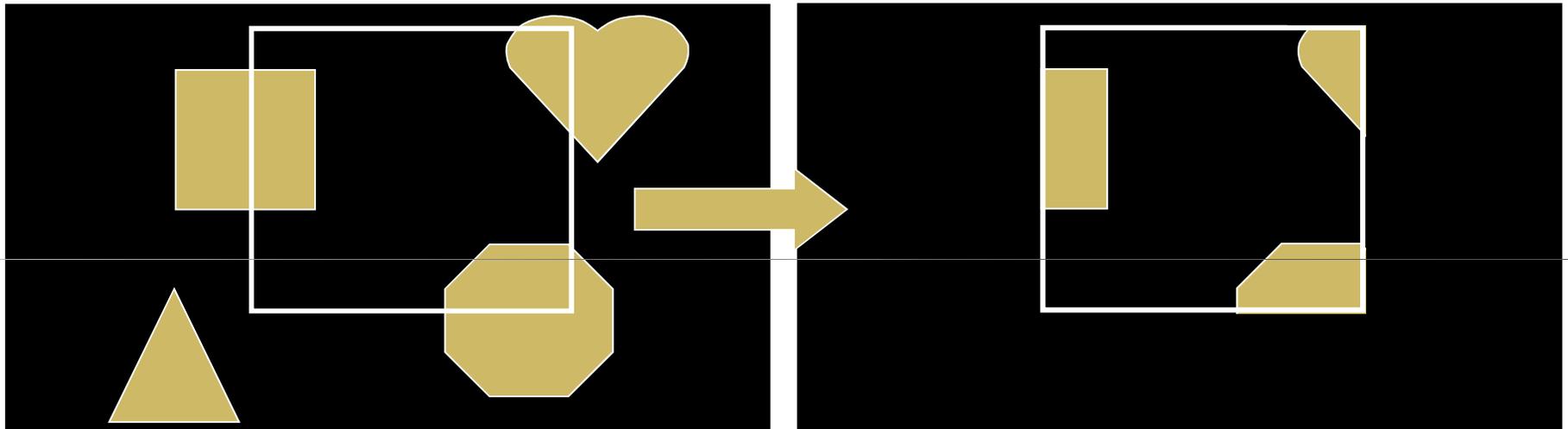
- Algoritmos para preenchimento: Polígonos convexos ou não



Passos:

- 1) Encontrar as intersecções da scan line com as arestas do polígono
- 2) Ordenar as intersecções
- 3) Preencher os pixels entre 2 intersecções (regra de paridade que inicia em par, muda quando encontra uma intersecção e escreve quando é impar)

# Algoritmos de recorte



Objetivo: otimização das estruturas a serem desenhadas no dispositivo

Trata o recorte contra as linhas da janela de seleção (retângulo)

# Algoritmos de recorte

## Algoritmo de Cohen-Sutherland

1001

1000

1010

0001

0000

0010

0101

0100

0110

# Algoritmos de recorte

## Algoritmo de Cohen-Sutherland

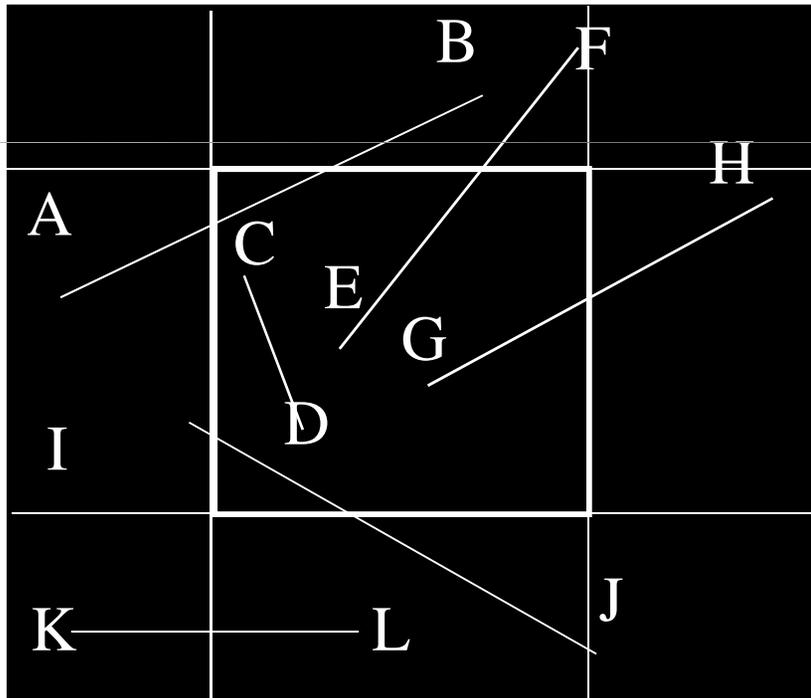
If  $y > y_f$  → seta primeiro bit em 1

If  $y < y_i$  → seta segundo bit em 1

If  $x > x_f$  → seta terceiro bit em 1

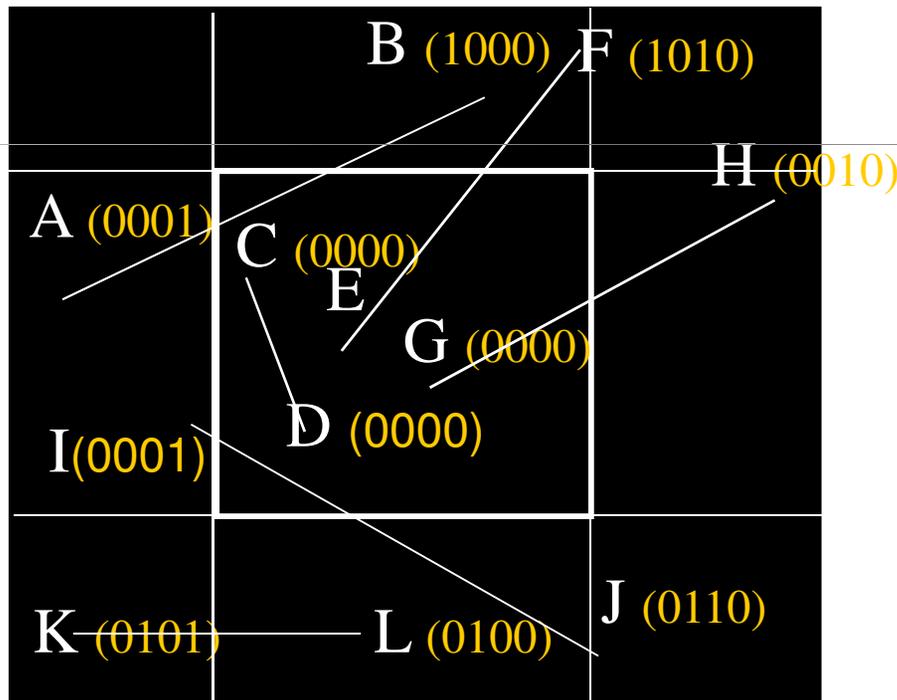
If  $x < x_i$  → seta quarto bit em 1

# Algoritmos de recorte



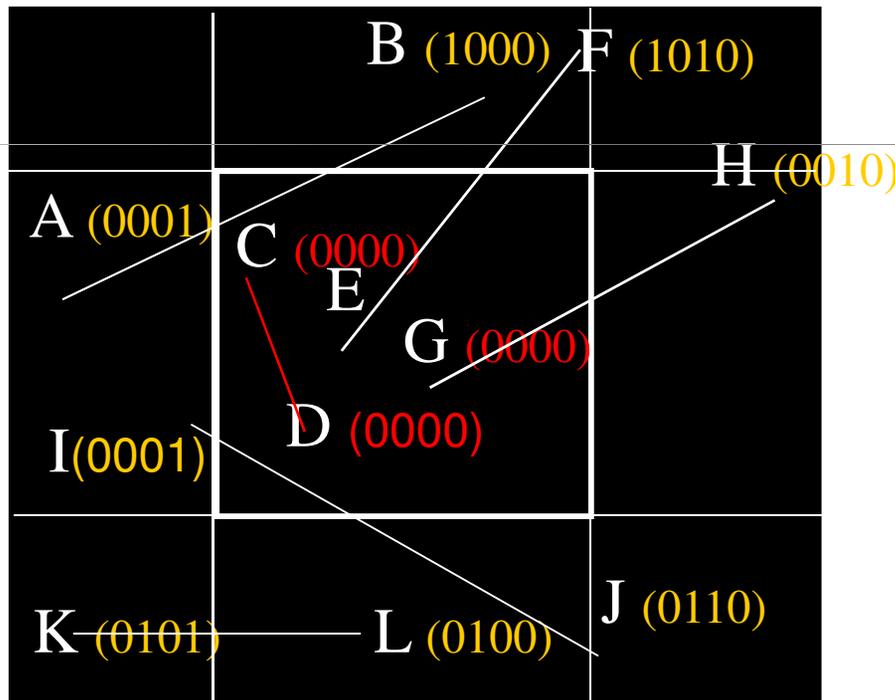
- Bit codes dos pontos?
- Como devem ser os bit codes dos pontos trivialmente aceitos?
- Como devem ser os bit codes das arestas trivialmente recusadas?
- O que fazer com os que não são nem aceitos nem recusados?

# Algoritmos de recorte



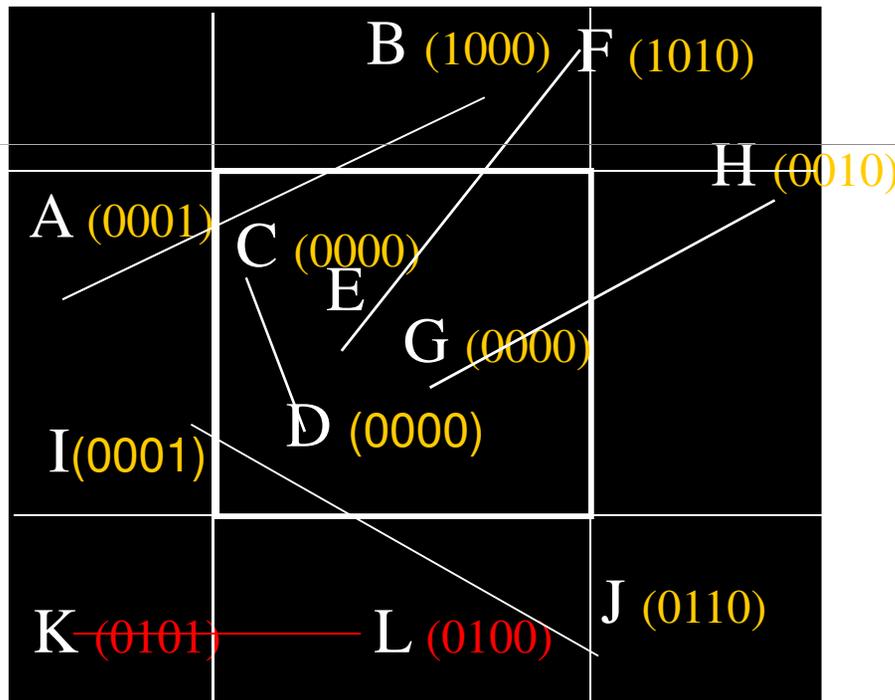
- Bit codes dos pontos?

# Algoritmos de recorte



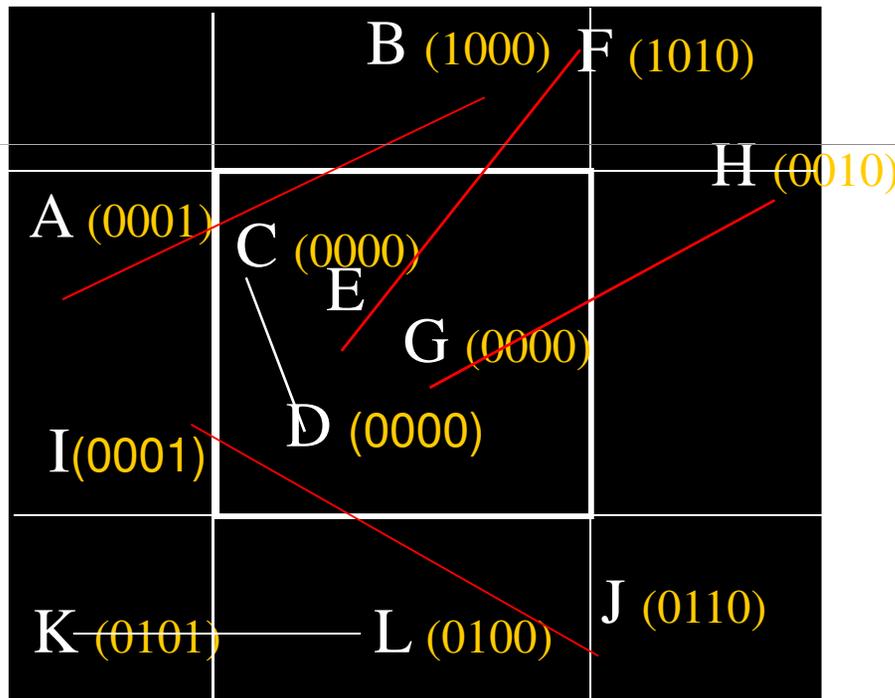
- Bit codes dos pontos?
- Como devem ser os bit codes dos pontos trivialmente aceitos?  
0000
  - Arestas formadas por pontos trivialmente aceitos são trivialmente aceitas

# Algoritmos de recorte



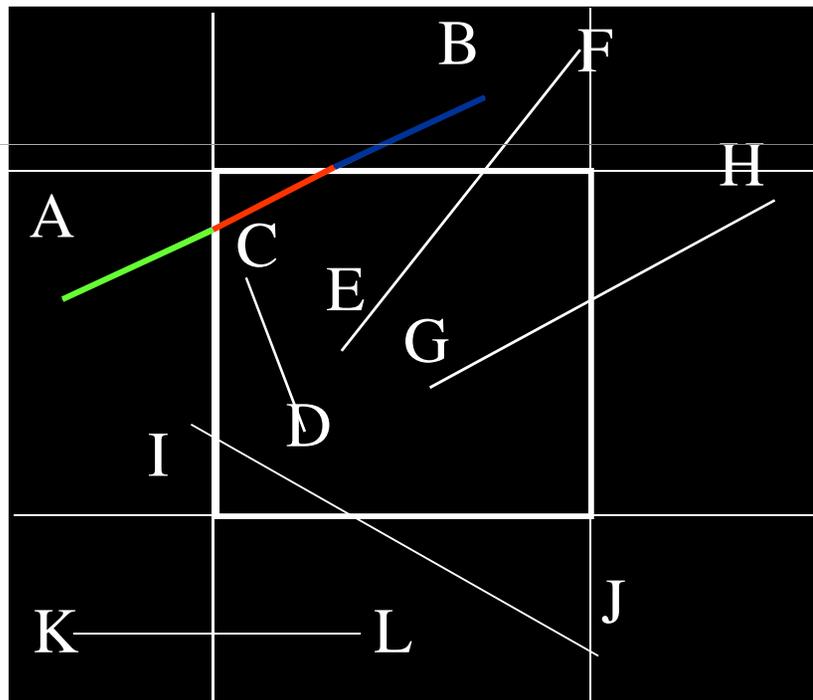
- Como devem ser os bit codes dos pontos trivialmente aceitos? 0000
- Como devem ser os bit codes das arestas trivialmente recusadas? AND entre end points deve ser  $\neq 0$

# Algoritmos de recorte

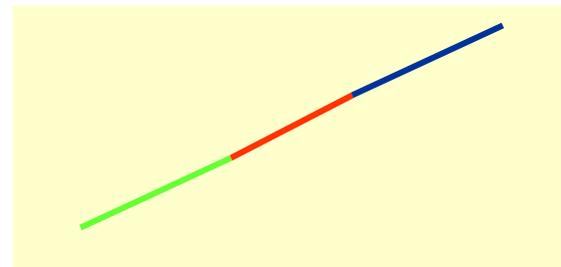


- Como devem ser os bit codes dos pontos trivialmente aceitos? 0000
- Como devem ser os bit codes das arestas trivialmente recusadas? AND entre end points deve ser  $\neq 0$
- O que fazer com os que não são nem aceitos nem recusados?

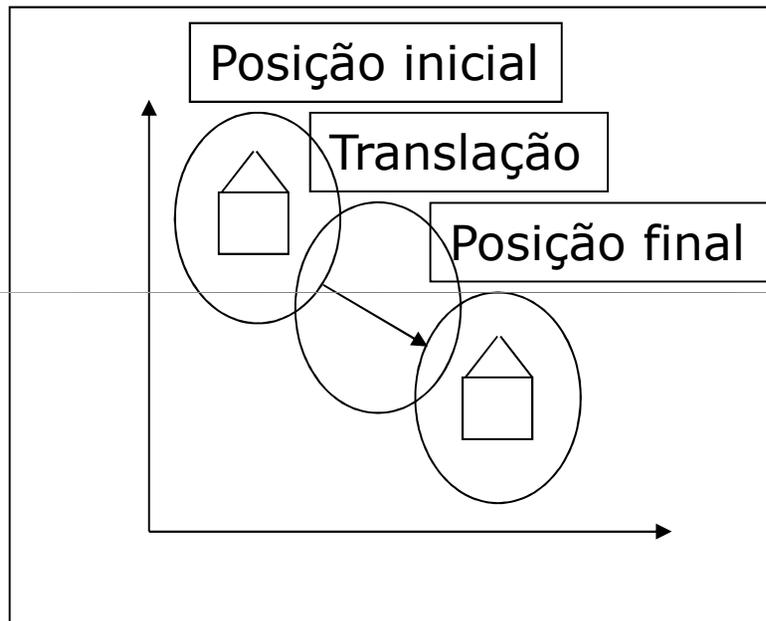
# Algoritmos de recorte



- Calcular segmentos através das intersecções



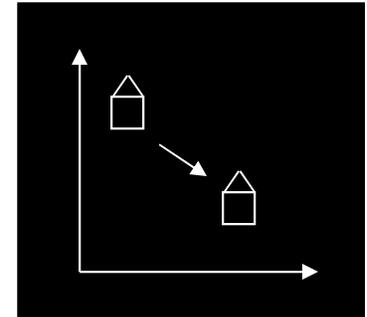
# Transformações 2D - Translação



# Transformações 2D - Translação

- ▣ Cada vértice é modificado

$$\begin{aligned}x' &= x + t_x \\ y' &= y + t_y\end{aligned}$$



- ▣ Utiliza-se vetores para representar a transformação

$$\vec{t} = \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

- ▣ Um ponto  $p(x,y)$  torna-se um vetor

- ▣ Assim, a translação torna-se uma mera soma de vetores

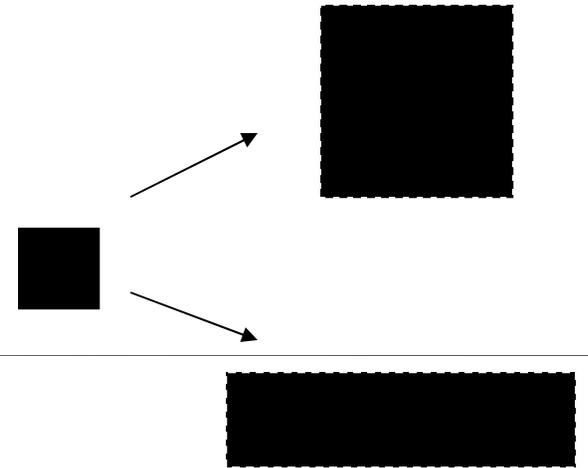
$$\vec{p} = \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\vec{p}' = \vec{p} + \vec{t}$$

# Transformações 2D - Escala

- Coordenadas são multiplicadas pelos fatores de escala

$$x' = x \cdot s_x$$
$$y' = y \cdot s_y$$



- Tipos de Escala

Uniforme:

$$s_x = s_y$$

Não-Uniforme

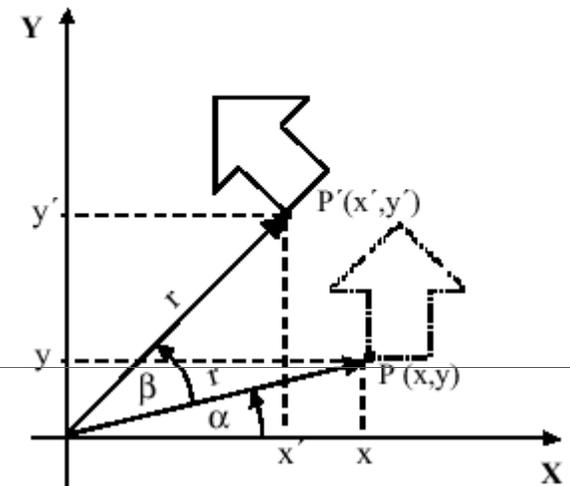
$$s_x \neq s_y$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x \cdot s_x + 0 \cdot y \\ 0 \cdot x + y \cdot s_y \end{bmatrix} = \begin{bmatrix} x \cdot s_x \\ y \cdot s_y \end{bmatrix}$$

- Escala é uma multiplicação de matrizes

# Transformações 2D - Rotação

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \beta & -\sin \beta \\ \sin \beta & \cos \beta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$



# Coordenadas Homogêneas

- Introduzida em Matemática
- Adiciona uma terceira coordenada  $w$
- Um ponto 2D passa a ser um vetor com 3 coordenadas

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

- 
- 2 pontos são iguais se e somente se:
  - Homogeneizar: dividir por  $w$
  - Pontos homogeneizados

$$\frac{x'}{w'} = \frac{x}{w} \text{ e } \frac{y'}{w'} = \frac{y}{w}$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

# Translação - Coord. Homogêneas

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix} \quad \vec{p}' = \vec{p} + \vec{t} \quad \begin{cases} \frac{x'}{w'} = \frac{x}{w} + t_x \\ \frac{y'}{w'} = \frac{y}{w} + t_y \end{cases}$$

---

$$\begin{cases} x' = x + wt_x \\ y' = y + wt_y \\ w' = w \end{cases}$$

# Escala - Coord. Homogêneas

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

$$\begin{cases} \frac{x'}{w'} = s_x \frac{x}{w} \\ \frac{y'}{w'} = s_y \frac{y}{w} \end{cases}$$

---

$$\begin{cases} x' = s_x x \\ y' = s_y y \\ w' = w \end{cases}$$

# Rotação - Coord. Homogêneas

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

$$\begin{cases} \frac{x'}{w'} = \cos \theta \frac{x}{w} - \sin \theta \frac{y}{w} \\ \frac{y'}{w'} = \sin \theta \frac{x}{w} + \cos \theta \frac{y}{w} \end{cases}$$

---

$$\begin{cases} x' = \cos \theta x - \sin \theta y \\ y' = \sin \theta x + \cos \theta y \\ w' = w \end{cases}$$

# Formas de Representação

