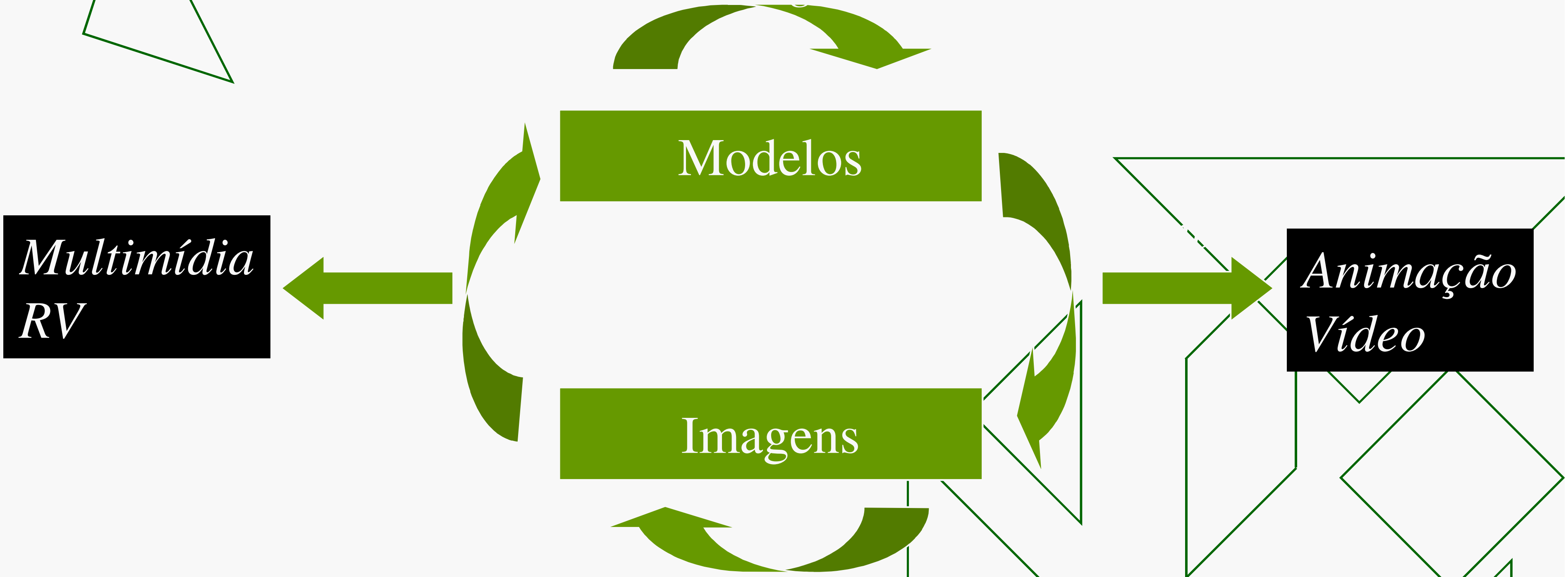
Revisão

Soraia Raupp Musse

Processamento Gráfico



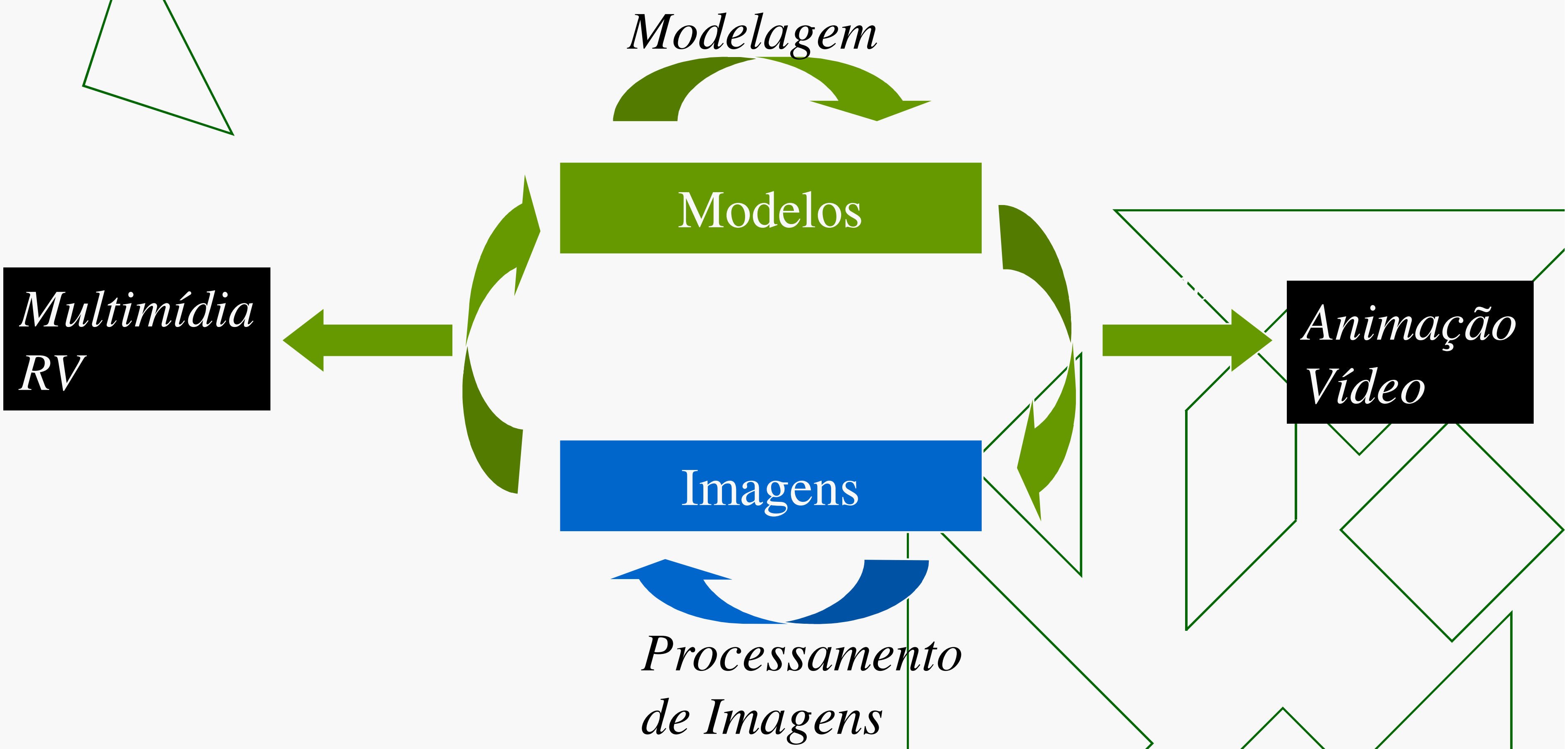
Modelos

Imagens

Multimídia
RV

Animação
Vídeo

Processamento de Imagens



Modelagem

Modelos

Multimídia
RV

Animação
Vídeo

Imagens

Processamento
de Imagens

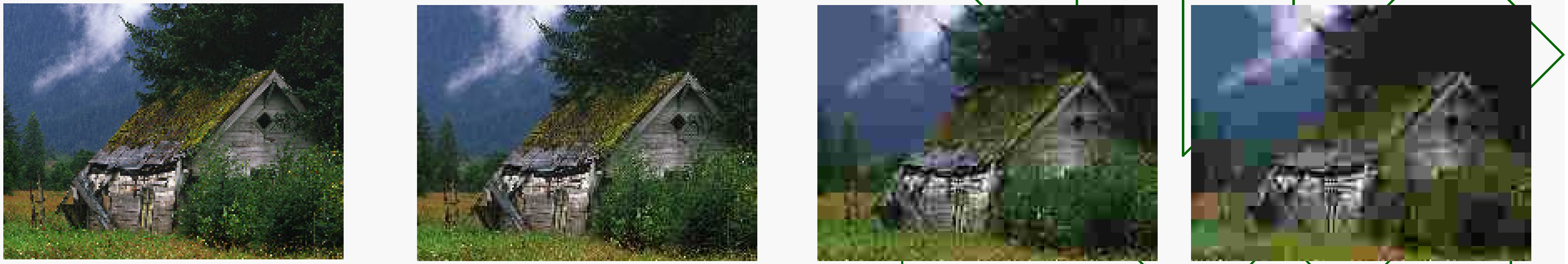
Compressão de Imagens

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



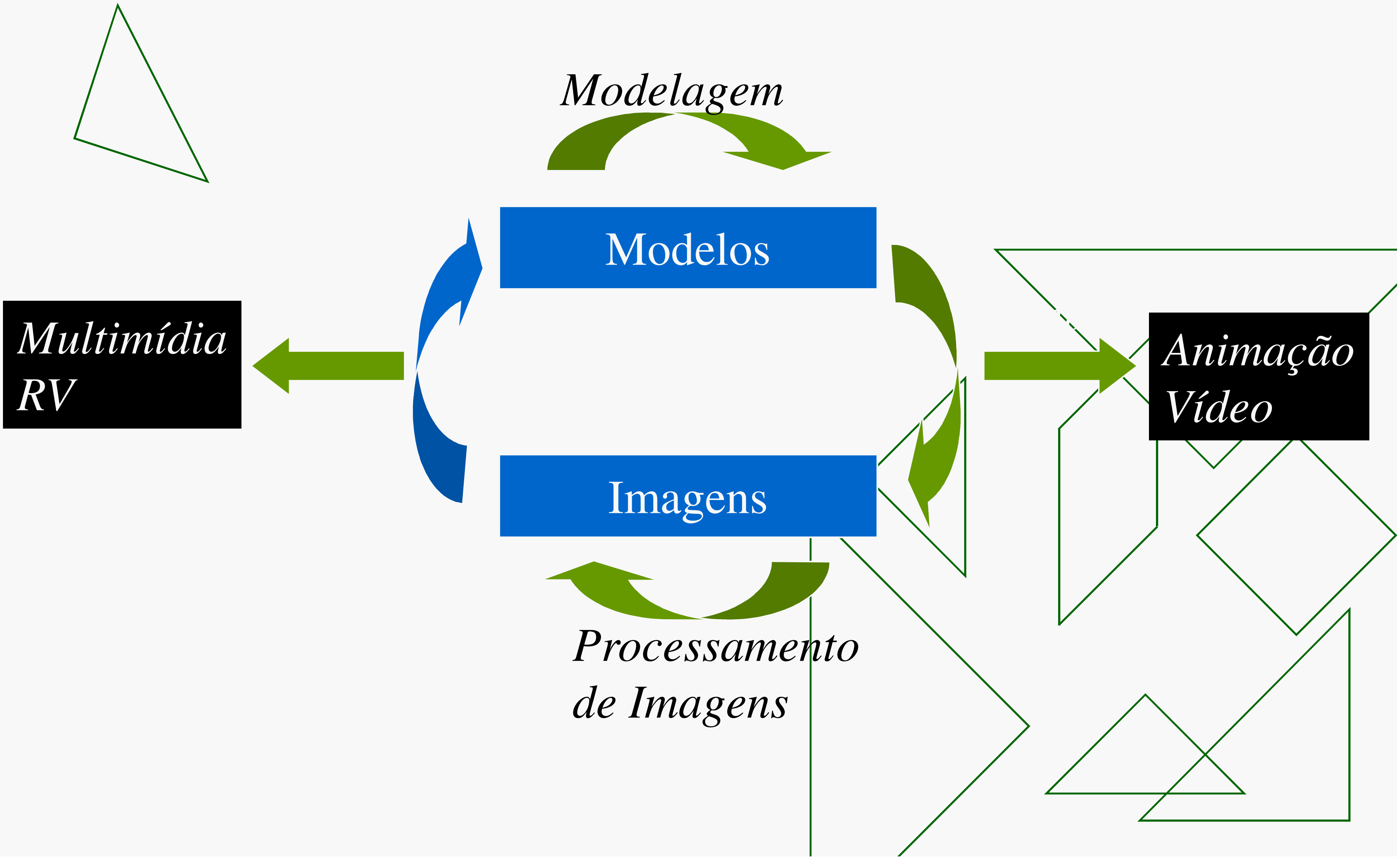
BMP (sem compressão) - 92 KB
(153 x 204 x 3 bytes)

JPEG - 6 KB

JPEG - 3 KB

JPEG - 2 KB

Visão Computacional



Modelagem

Modelos

Multimídia
RV

Animação
Vídeo

Imagens

Processamento
de Imagens

Image denoising and enhancement

Rectangle detection

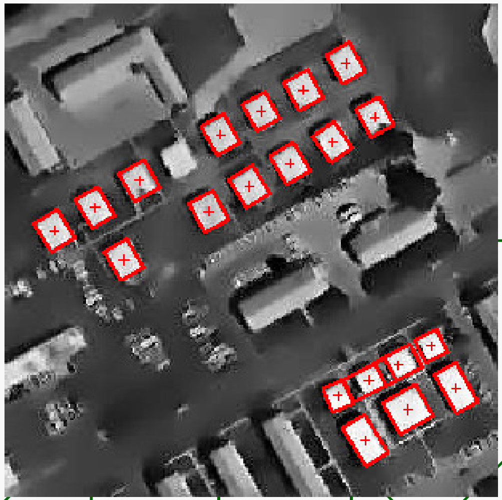
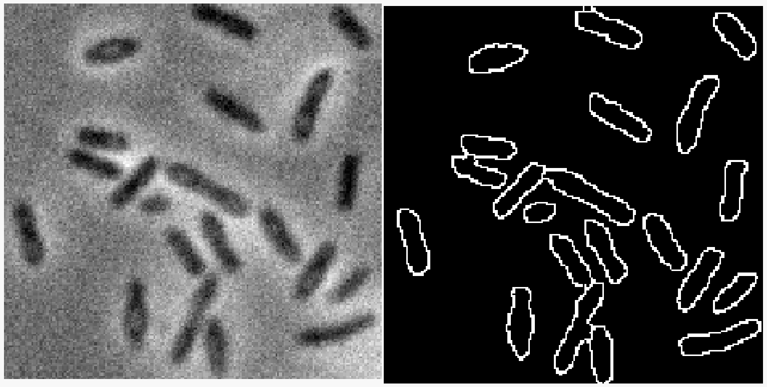
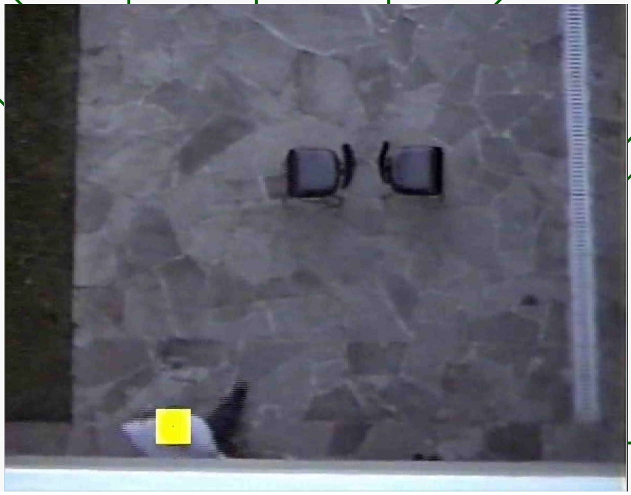


Image Segmentation



People tracking



O que é Computação Gráfica?

Dados

IMAGEM

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

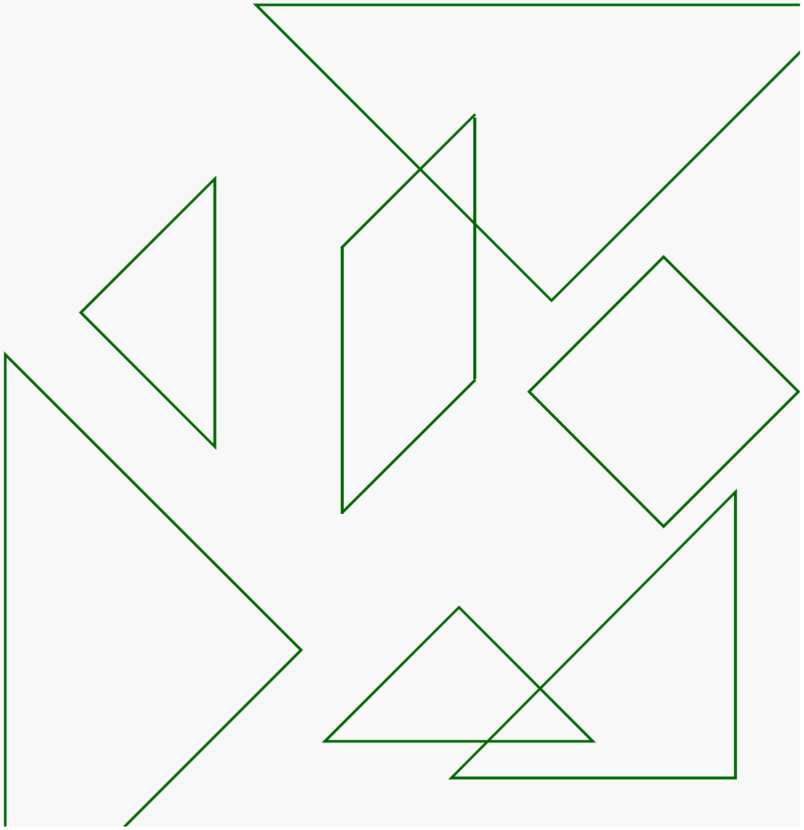
Temas da Computação Gráfica

Forma
Modelagem Geométrica

Aparência
Renderização

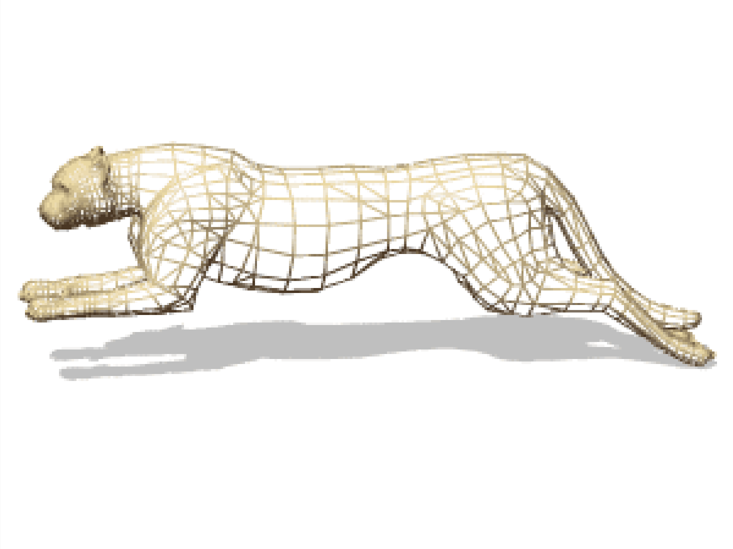
Ação
Animação

Interfaces
RV



Modelagem Geométrica

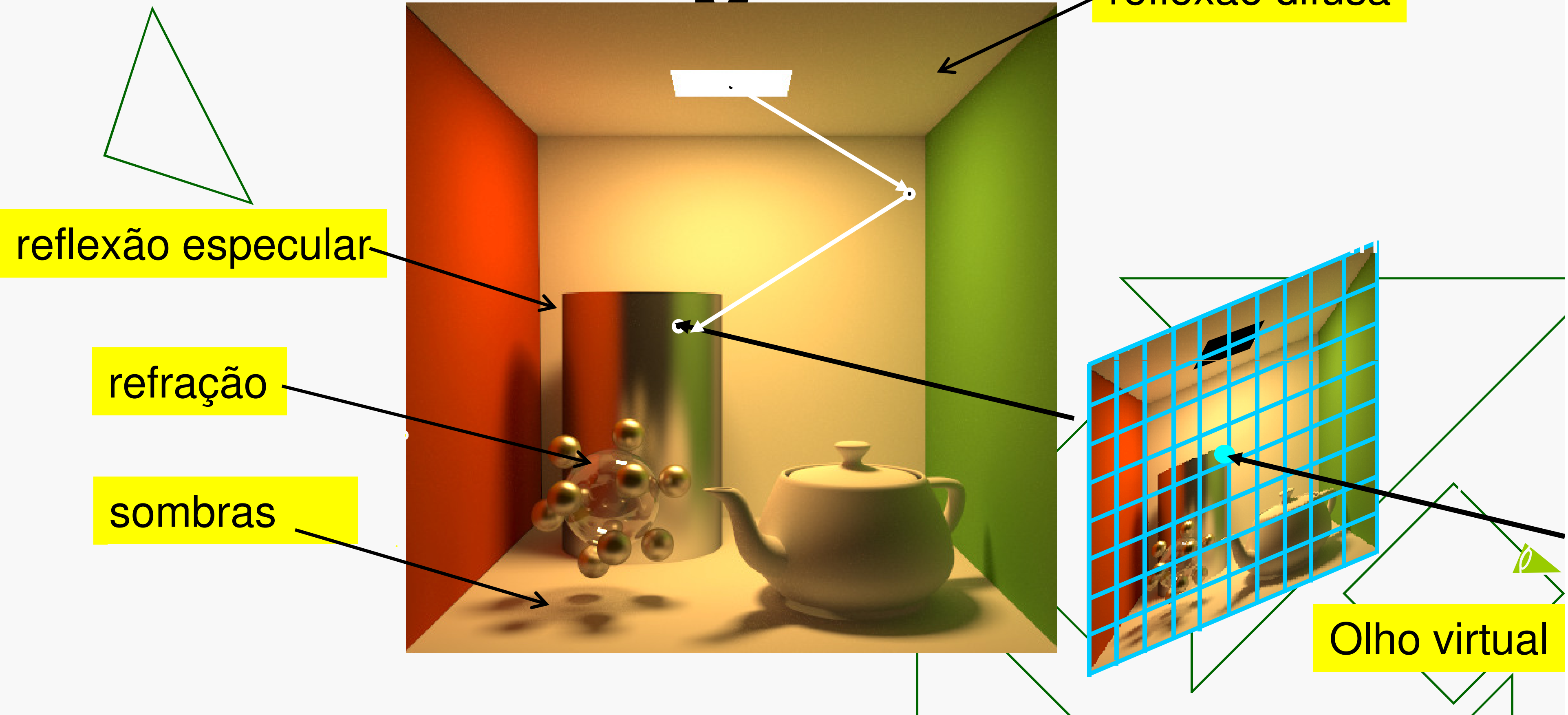
- Como criar/projetar/representar objetos
- Como representar coisas e ambientes complexos (um bicho de pelúcia é complexo?)



Renderização

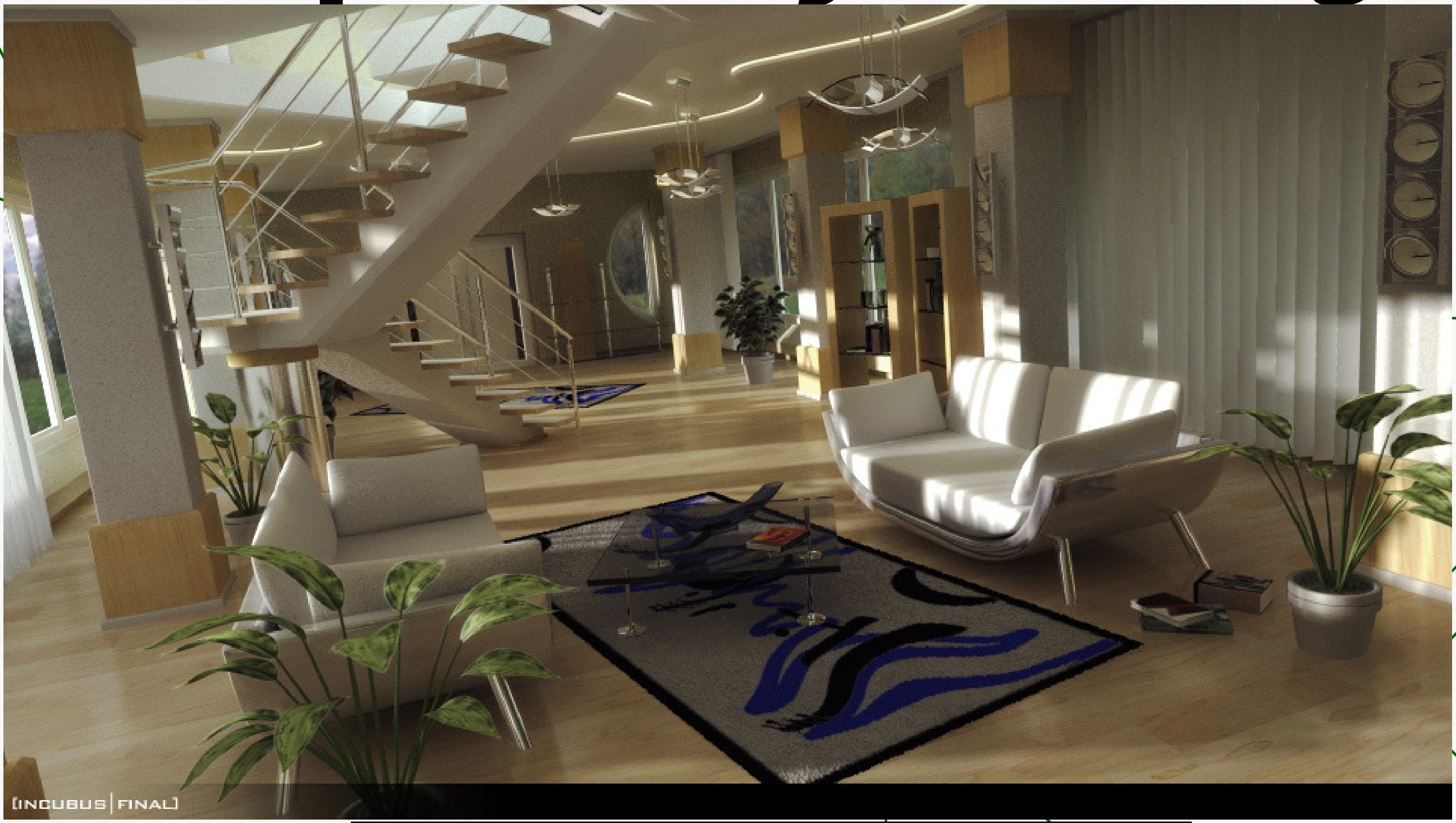
- Uma imagem é uma distribuição de energia luminosa num meio bidimensional (o plano do filme fotográfico, por exemplo)
- Dados uma descrição do ambiente 3D e uma câmera virtual, calcular esta energia em pontos discretos (tirar a fotografia)
- Resolver equações de transporte de energia luminosa através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

Rendering


reflexão difusa
reflexão especular
refração
sombras
Olho virtual

Principais fenômenos que podem acontecer na interação entre luz e objetos

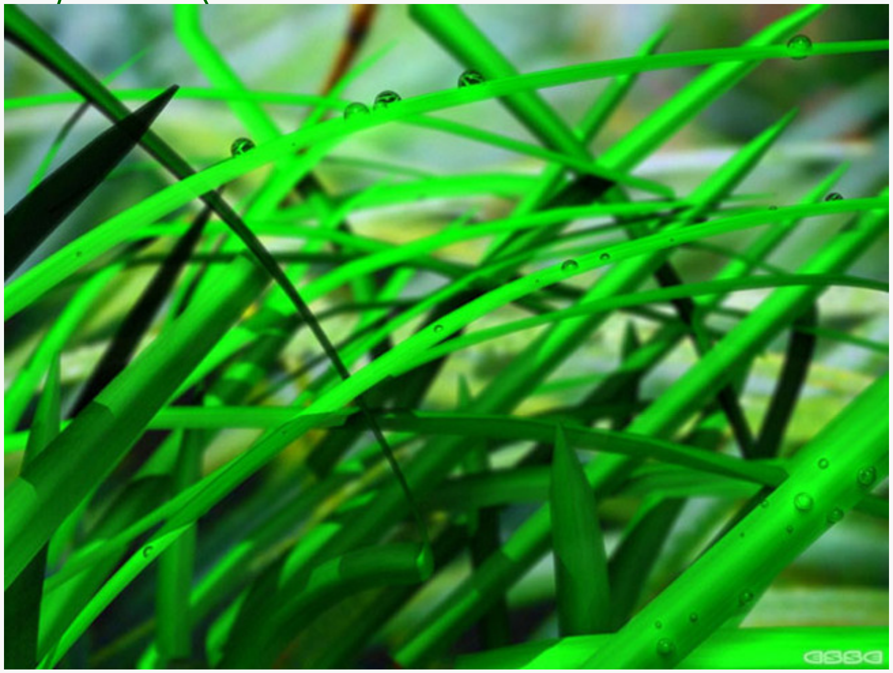
Exemplo – *Ray Tracing*

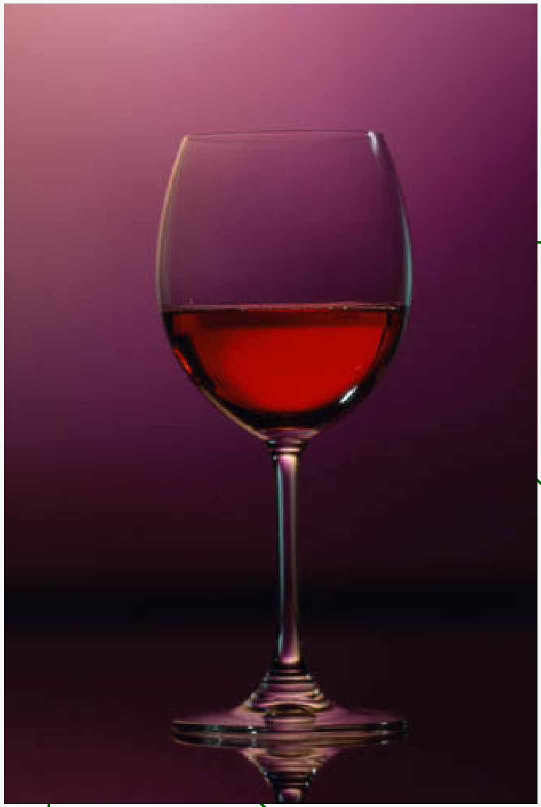


Kirschner, Andre
RENDERER USED: 3d studio max
RENDER TIME: approx 6 hours 30 minutes

HARDWARE USED: AMD1600+, ti4200

Real ou Computação Gráfica?



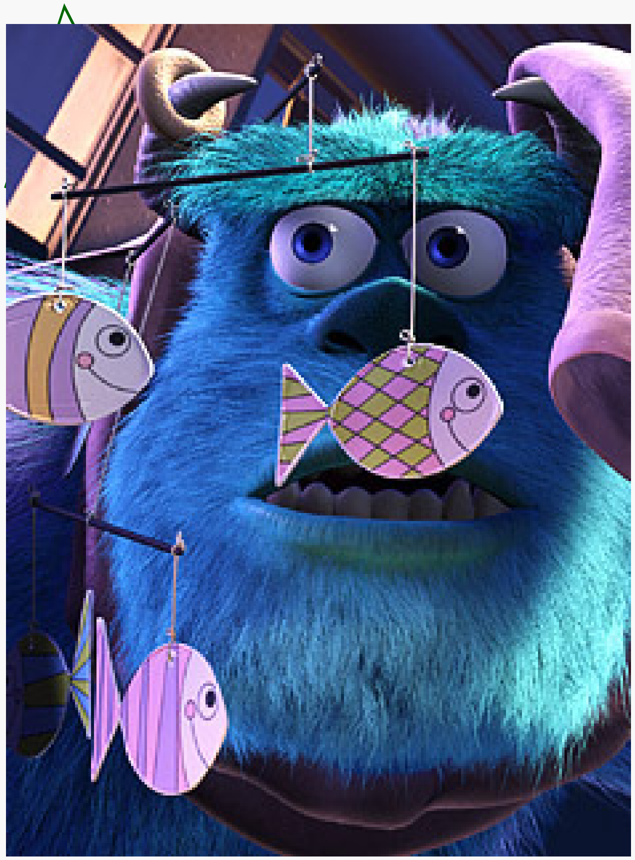


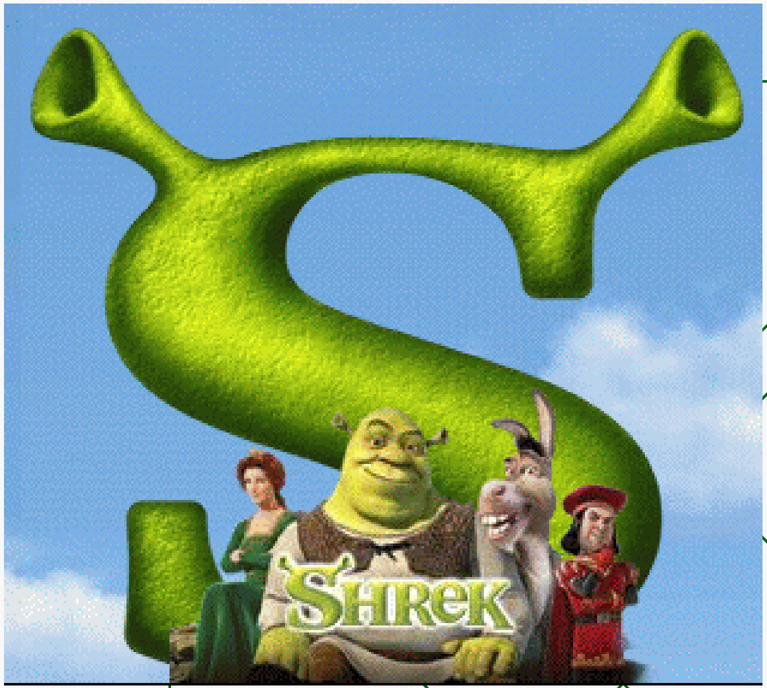
http://www.fakeorreal.com

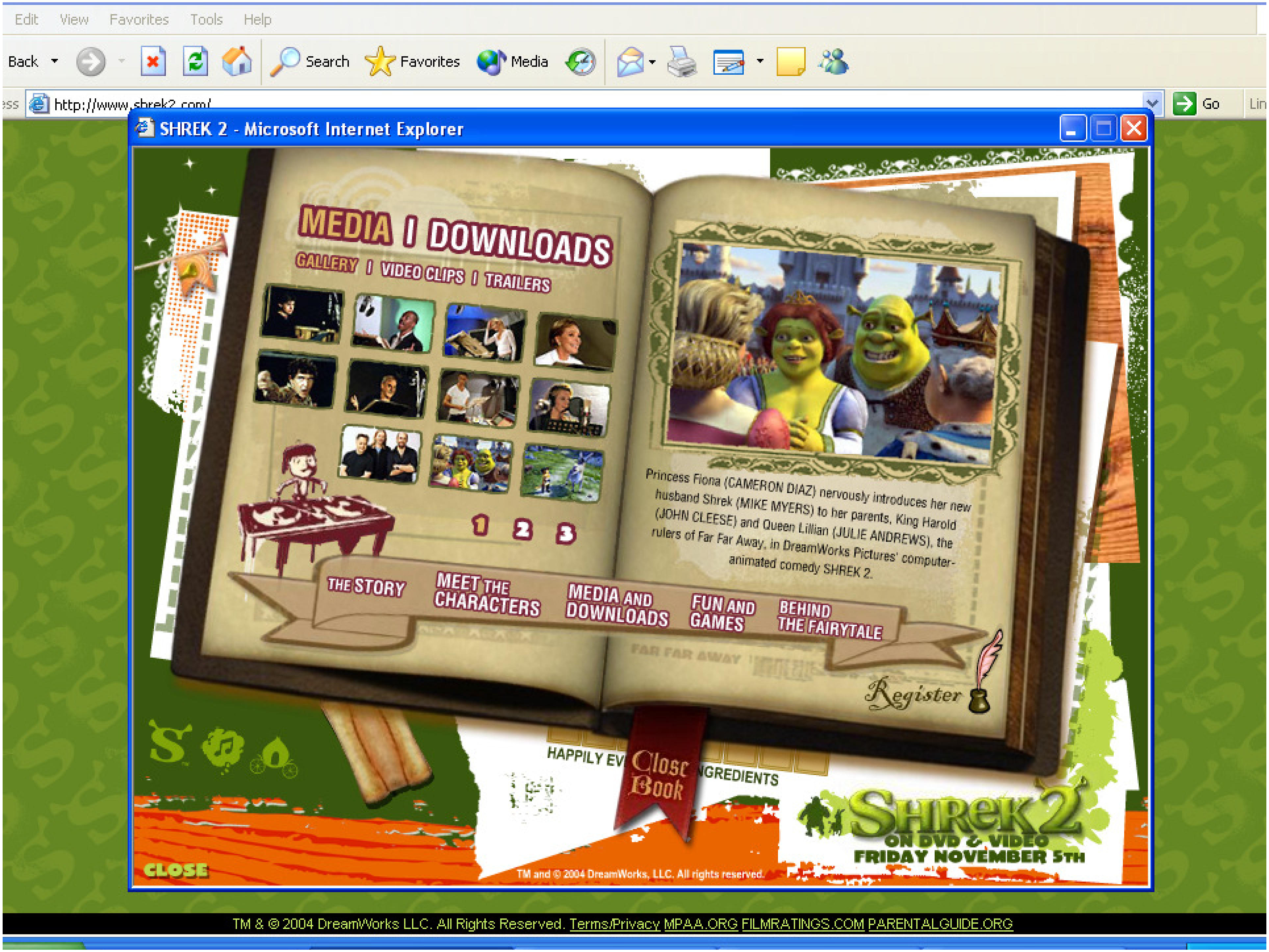
Animação

- 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 a Animação?

Exemplos
Monstros, Shrek






Edit View Favorites Tools Help
Back
Search
Favorites
Media
http://www.shrek2.com/
Go
SHREK 2 - Microsoft Internet Explorer
MEDIA | DOWNLOADS
GALLERY | VIDEO CLIPS | TRAILERS
1 2 3
THE STORY
MEET THE CHARACTERS
MEDIA AND DOWNLOADS
FUN AND GAMES
BEHIND THE FAIRYTALE
FAR FAR AWAY
Register
HAPPILY EV
INGREDIENTS
Close Book
SHREK 2
ON DVD & VIDEO
FRIDAY NOVEMBER 5TH
CLOSE

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.

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

TM & © 2004 DreamWorks LLC. All Rights Reserved. Terms/Privacy MPAA.ORG FILMRATINGS.COM PARENTALGUIDE.ORG

Matemática para CG

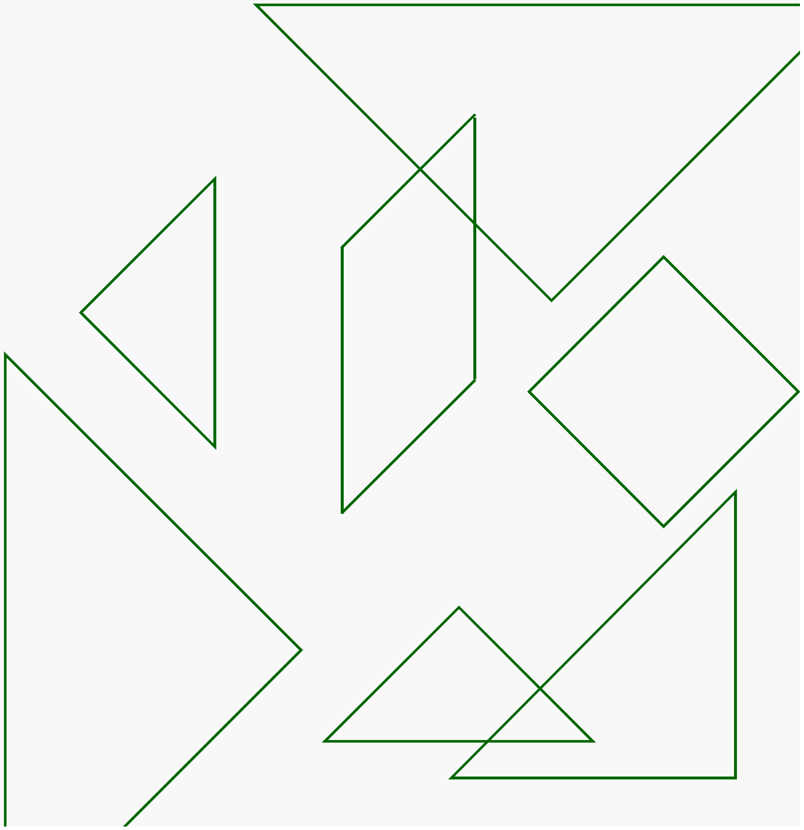
Nesta aula nós vimos..

- Introdução
- Revisão Matemática
  - Vetores
  - Matrizes

Processamento de Imagens

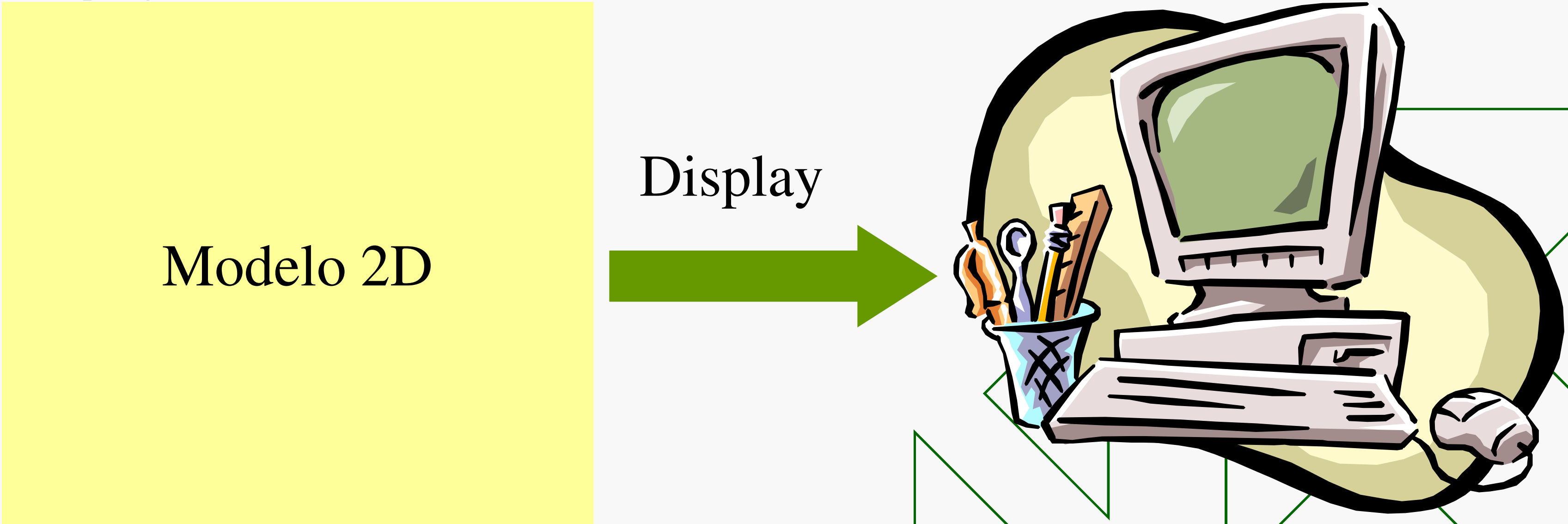
- Cor
- Percepção
- Filtros
- Img colorida/B&W
- Segmentação/detecção/tracking
- Aplicações

Algoritmos de rasterização para primitivas 2D



Objetivo:

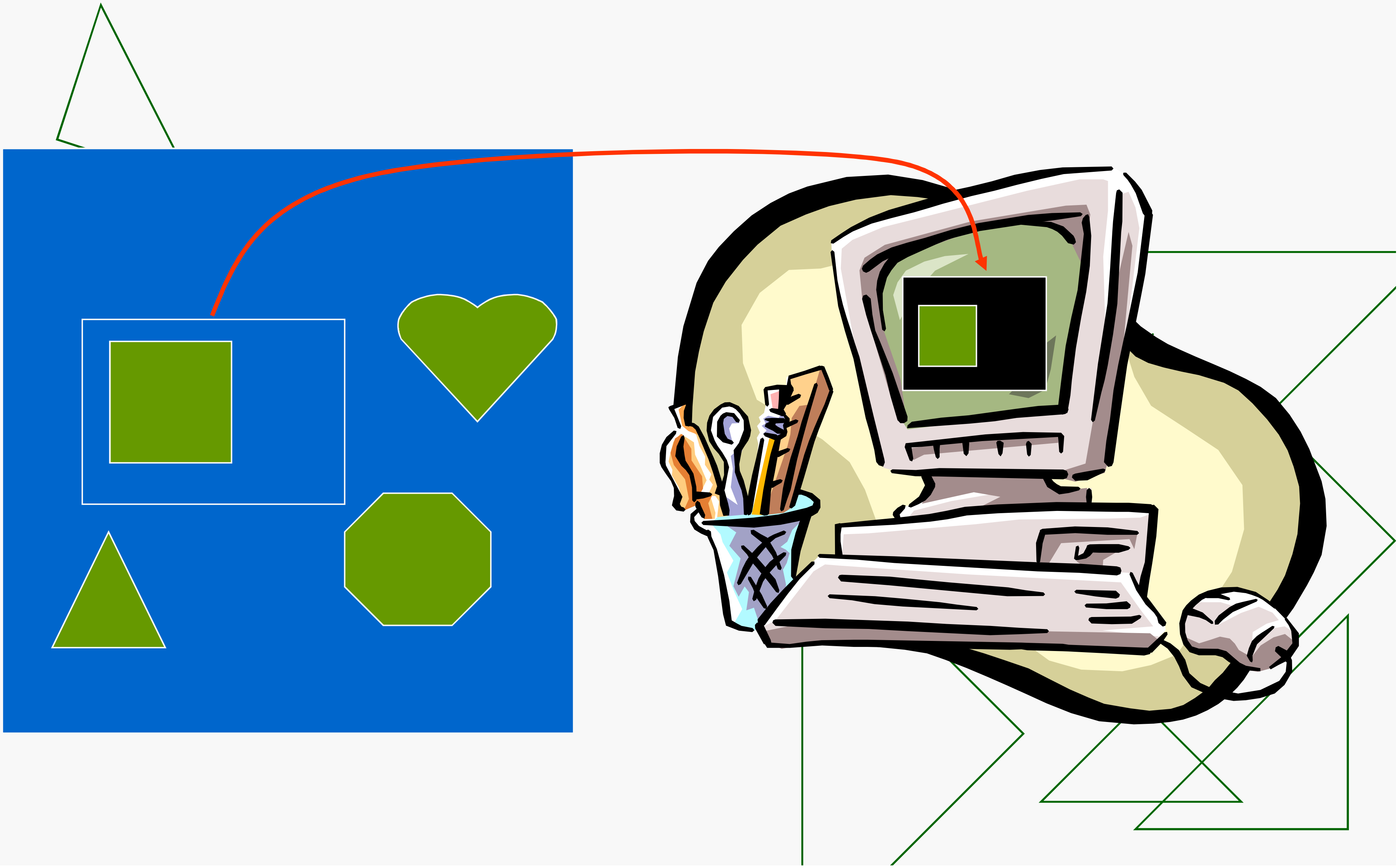
Aproximar primitivas matemáticas descritas através de vértices por meio de um conjunto de pixels de determinada cor



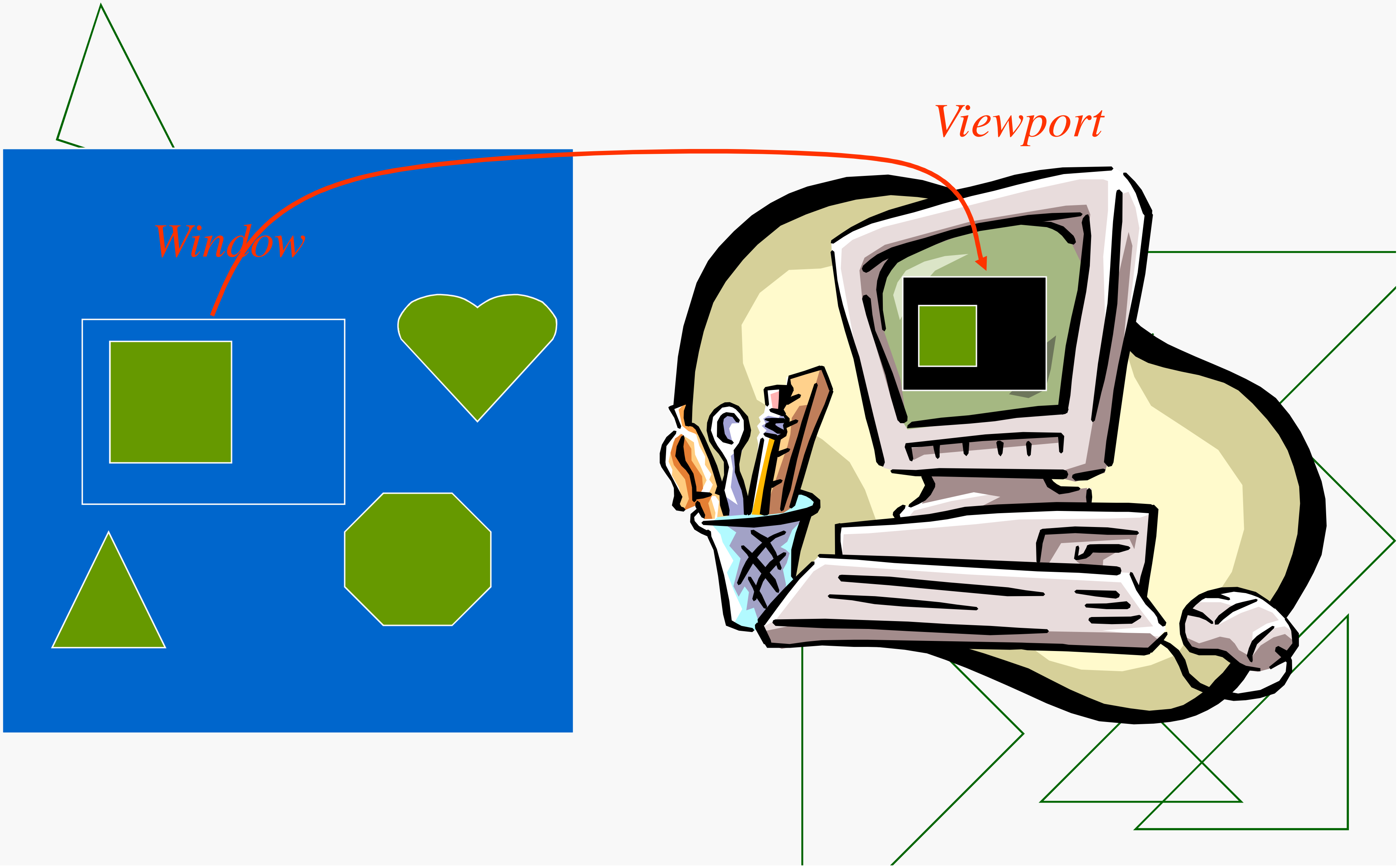
Modelo 2D

Display

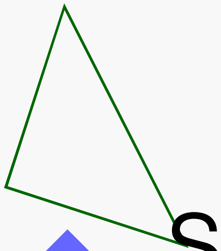
Pipeline de visualização 2D



Pipeline de visualização 2D


Window
Viewport

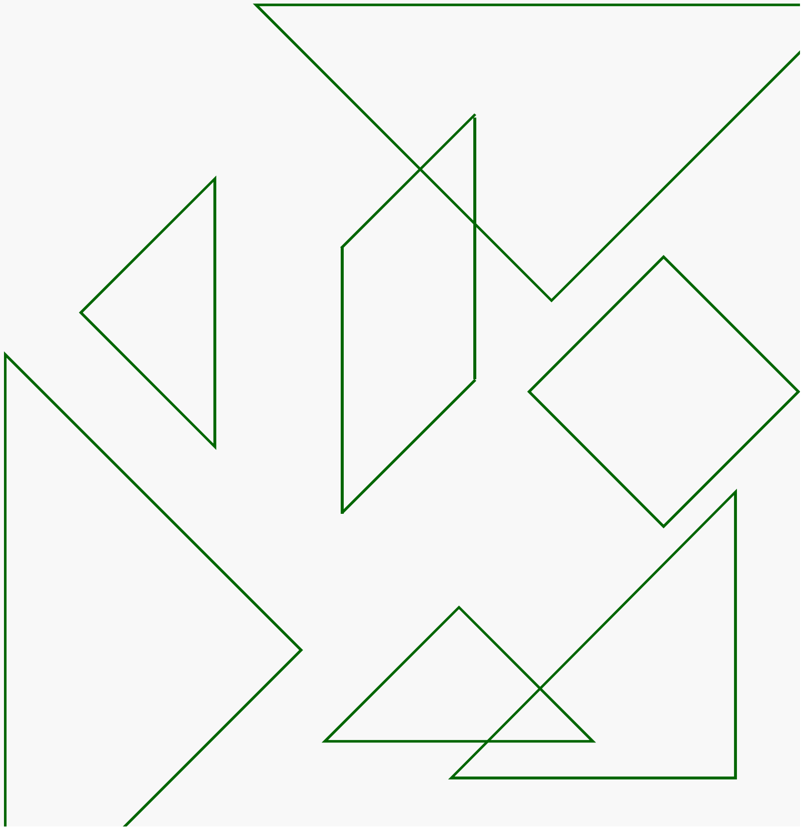
SR´s 2D



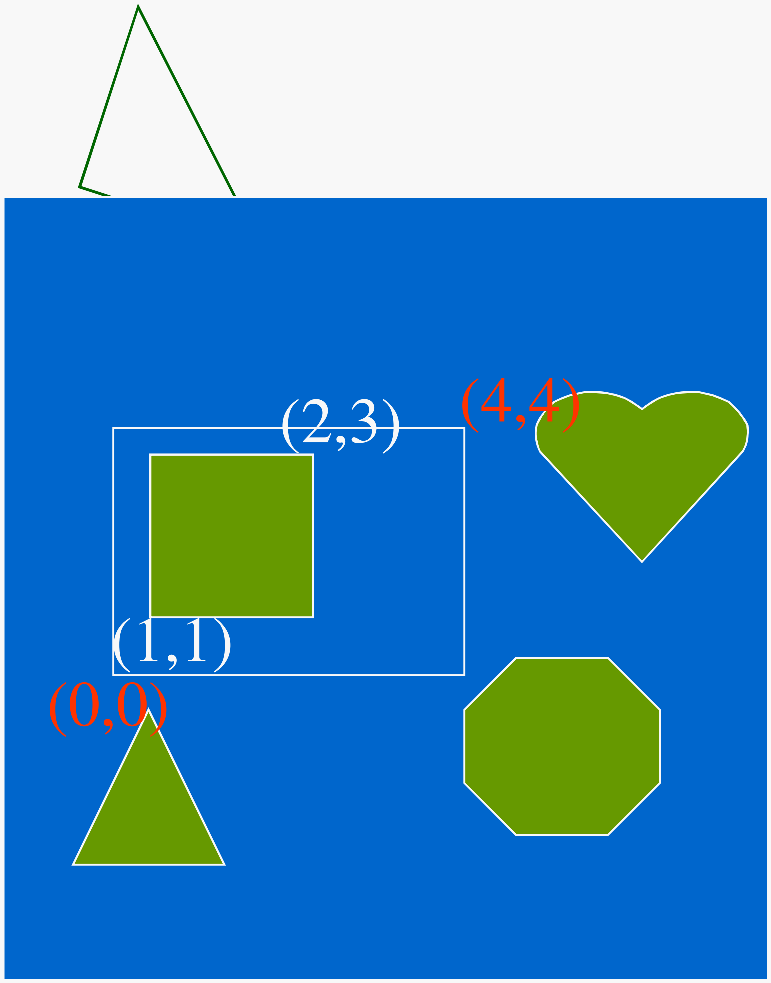
- SRO
- SRU
- SRW

(recorte 2D)

- SRV
- SRD



Pipeline de visualização 2D



Window_{vi} SRU = (1,7)
Window_{vf} SRU = (5,11)

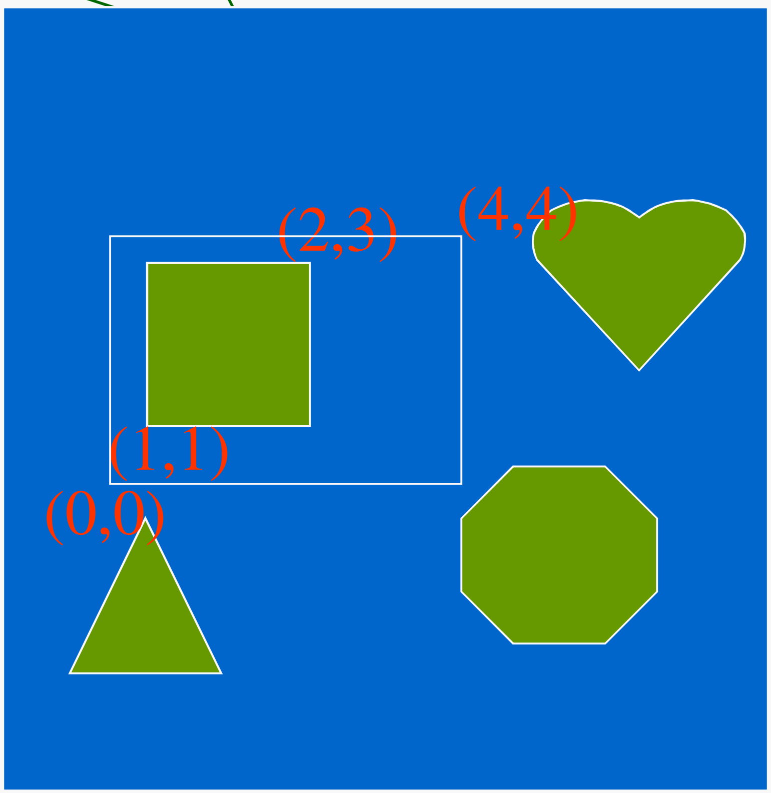
$T_{SRU\text{-}SRW} = 0 - \text{Window}_{vi}\text{SRW}$

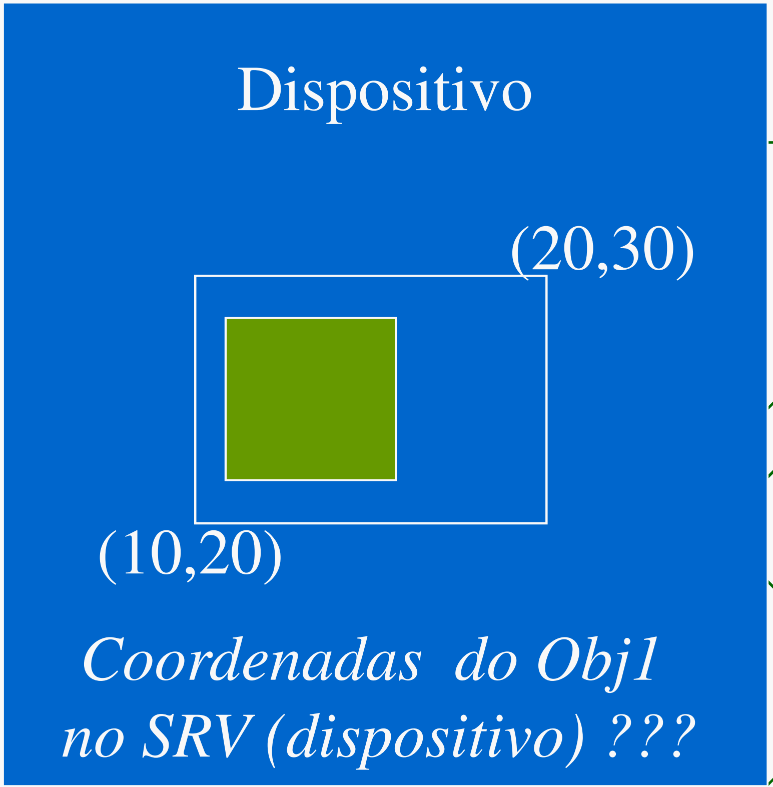
$T_{SRU\text{-}SRW}x = -1$
$T_{SRU\text{-}SRW}y = -7$

Window_{vi} SRW = (0,0)
Window_{vf} SRW = (4,4)

Obj1vi Window = (1,1)
Obj1vf Window = (2,3)

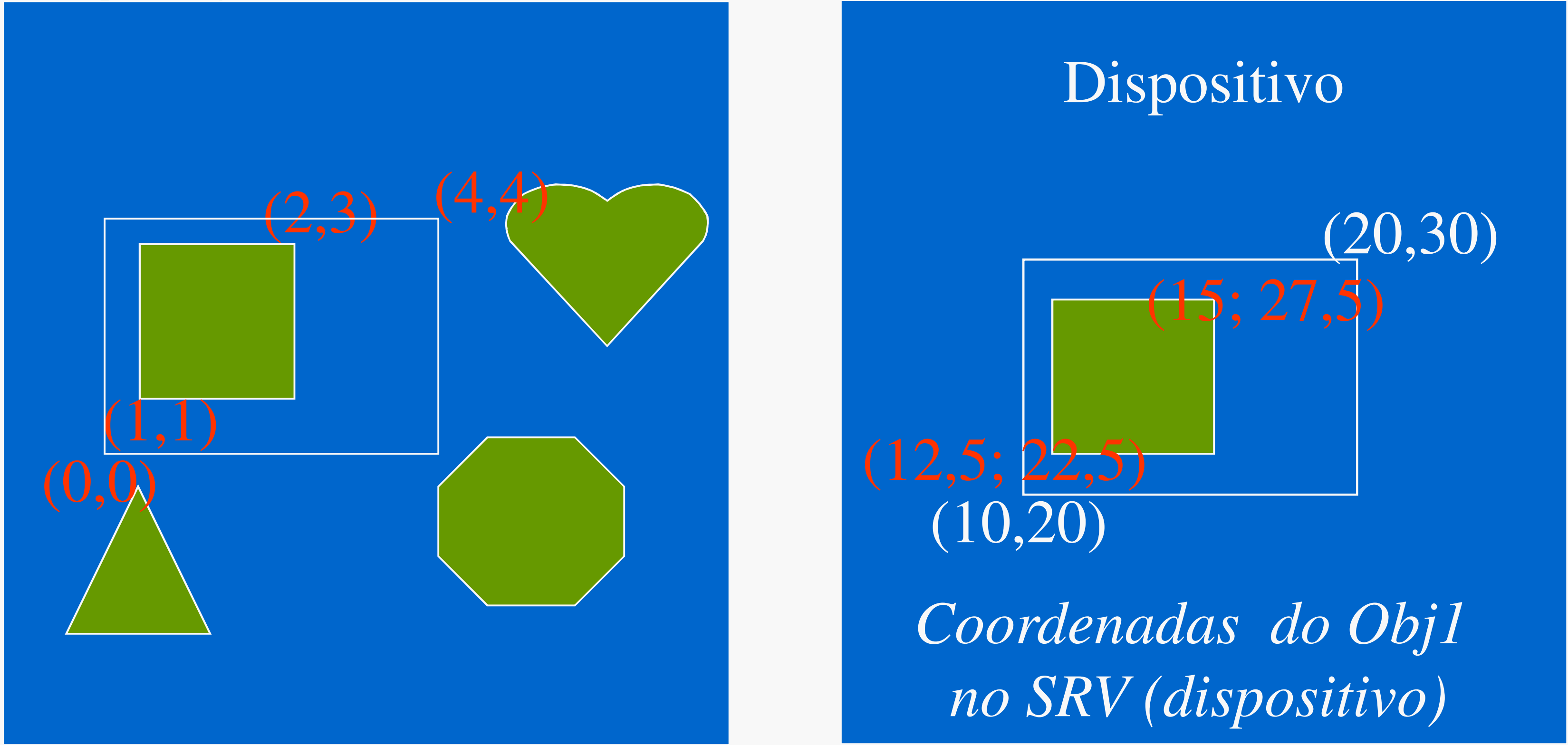
Pipeline de visualização 2D




Dispositivo
(20,30)
(10,20)

Coordenadas do Obj1 no SRV (dispositivo) ???

Mapeamento:

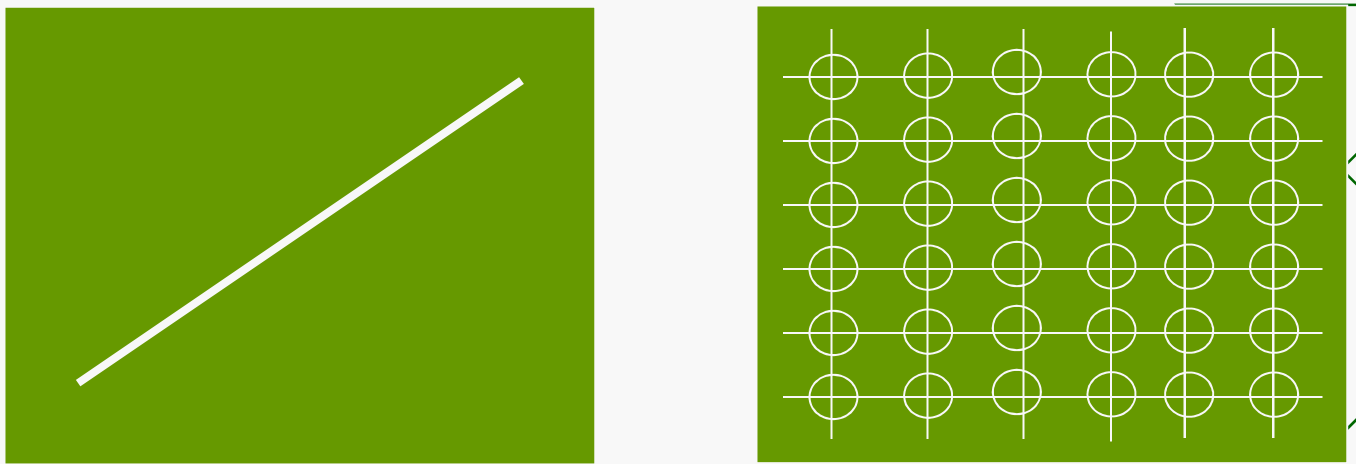

(4,4)
(2,3)
(1,1)
(0,0)
(20,30)
(15; 27,5)
(12,5; 22,5)
(10,20)

Dispositivo

Coordenadas do Obj1 no SRV (dispositivo)

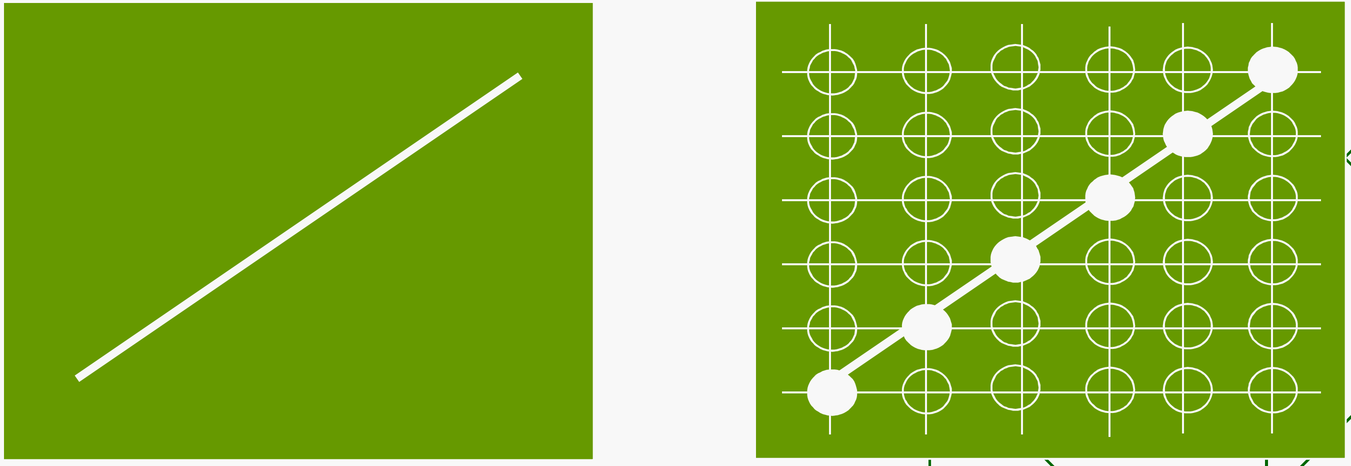
Algoritmos de rasterização

- Algoritmos de varredura



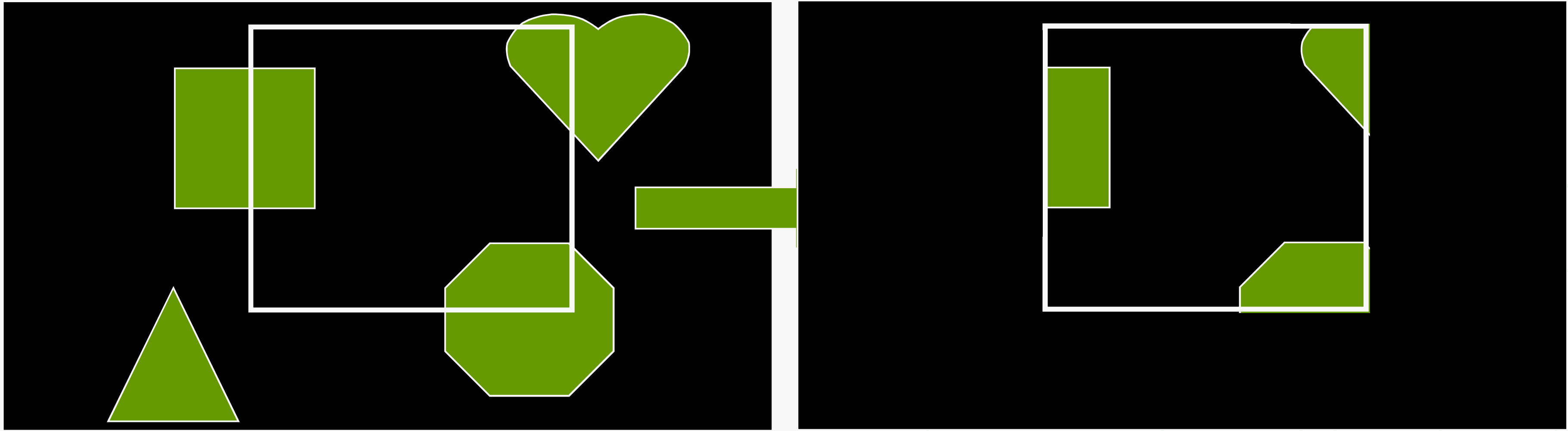
Algoritmos de rasterização

- Algoritmos de varredura



Algoritmos de rasterização

- Algoritmos de varredura
- Algoritmos de recorte



Algoritmos de varredura

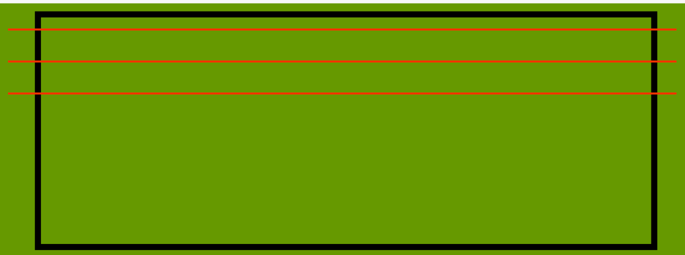
- Bresenham – Algoritmo do ponto médio
  - Atrativo porque usa somente operações aritméticas (não usa round ou floor)
  - É incremental

Algoritmos de varredura

- Algoritmos para preenchimento. Dois problemas:
  - Decisão de qual pixel preencher
  - Decisão de qual valor preencher

Algoritmos de varredura

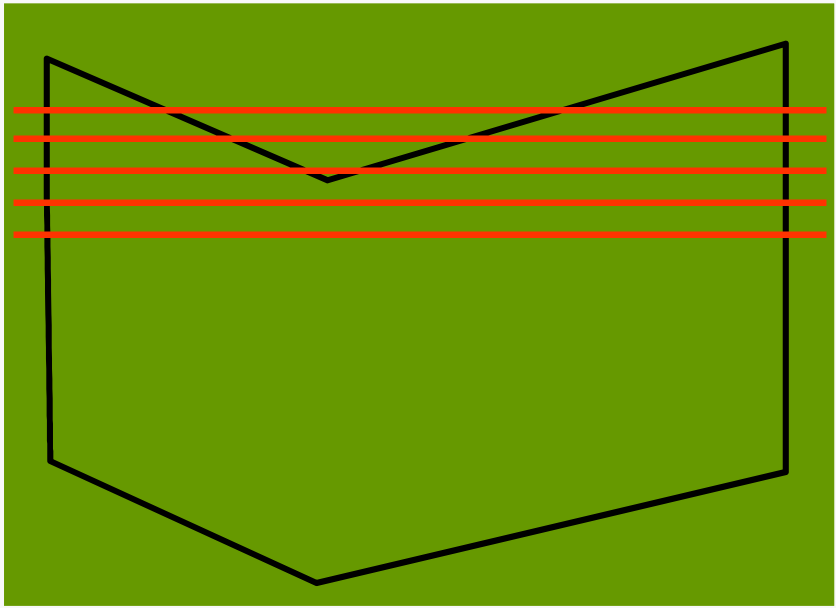
- Algoritmos para preenchimento: Retângulo



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

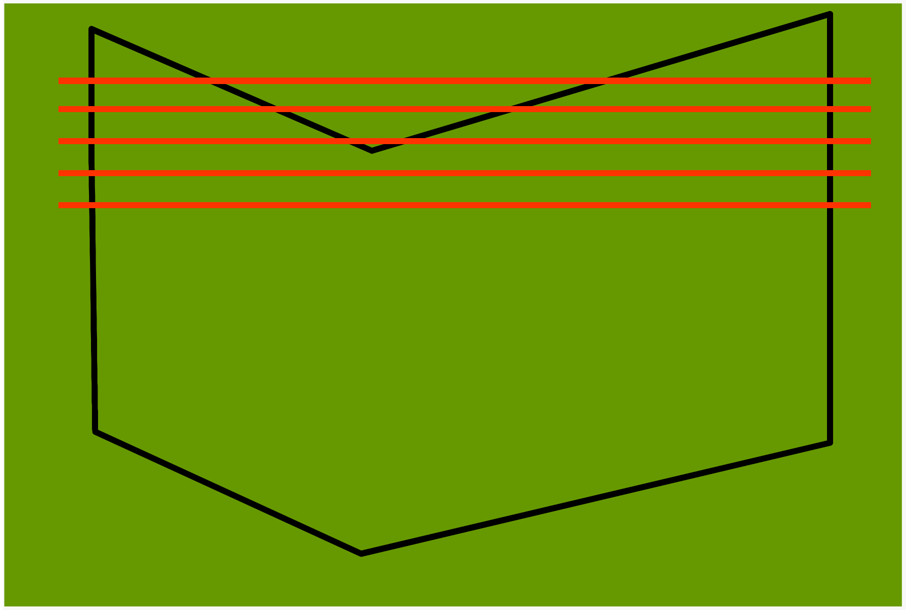
Algoritmos de varredura

- Algoritmos para preenchimento: Polígonos convexos ou não



Algoritmos de varredura

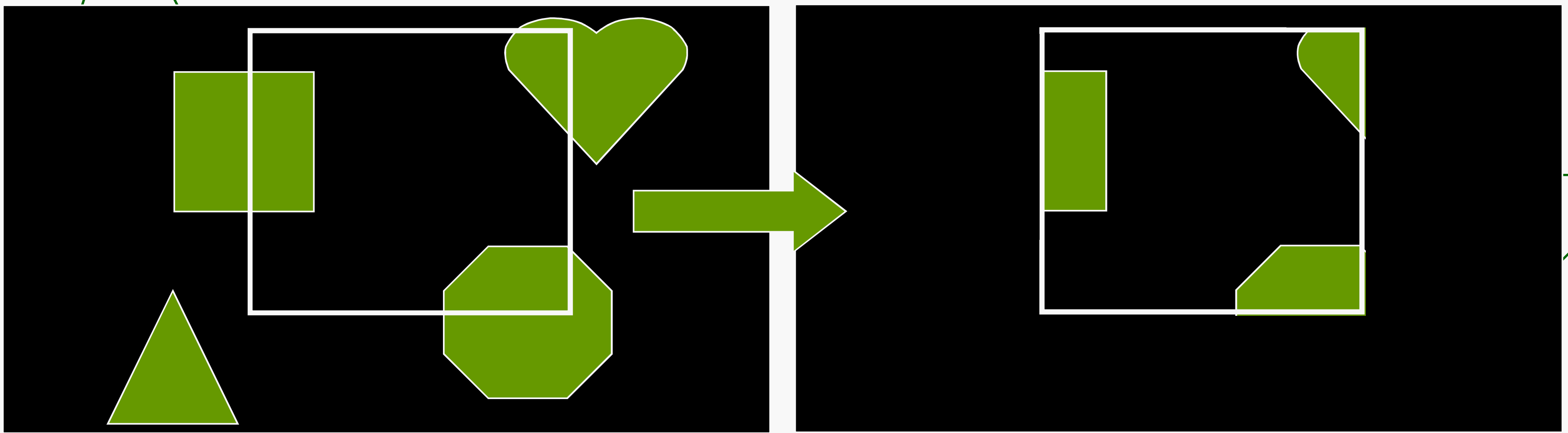
- Algoritmos para preenchimento: Polígonos convexos ou não



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)

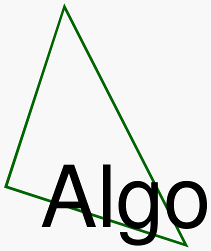
Algoritmos de recorte



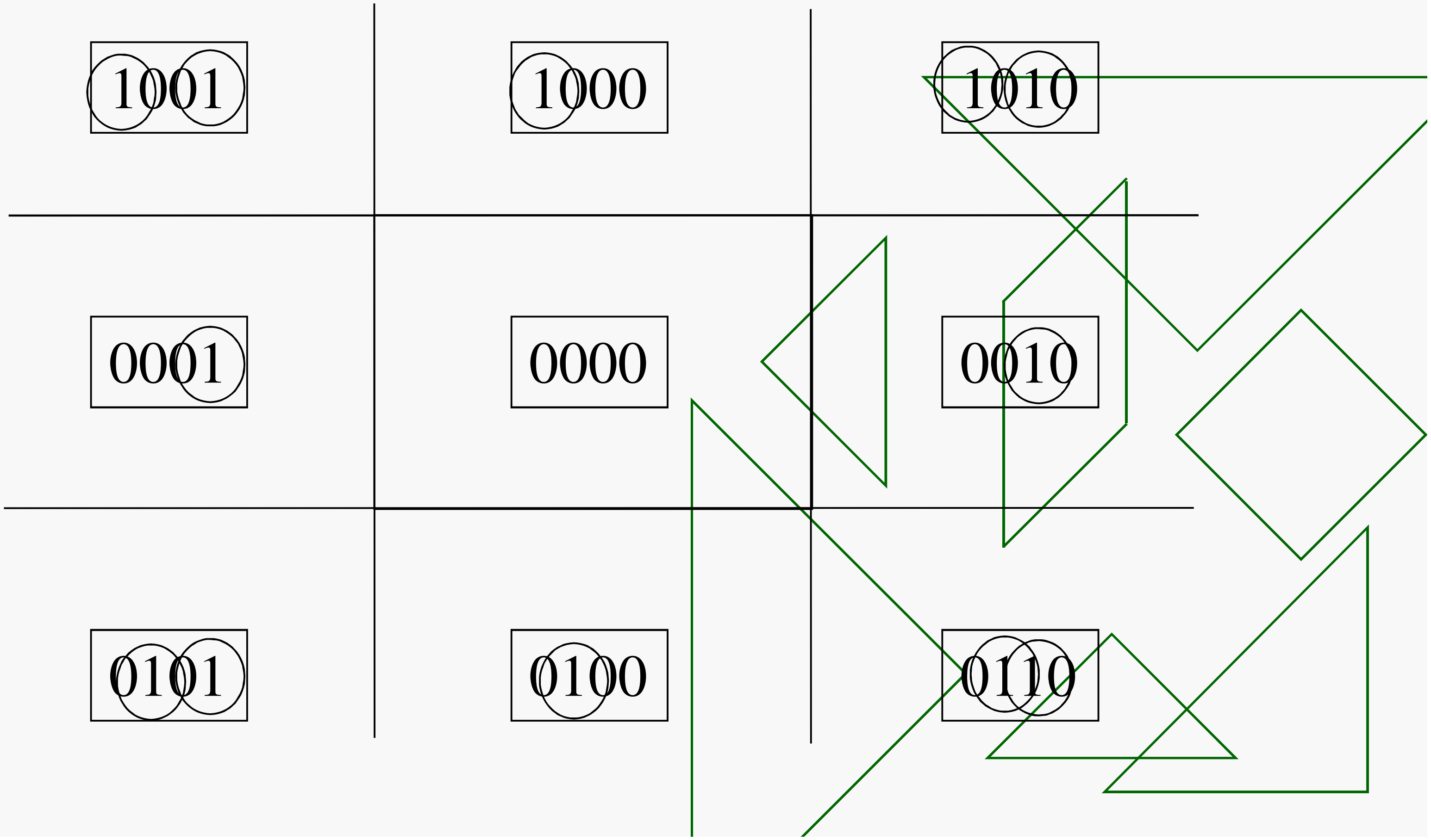
Objetivo: otimização das estruturas a serem desenhadas no dispositivo

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

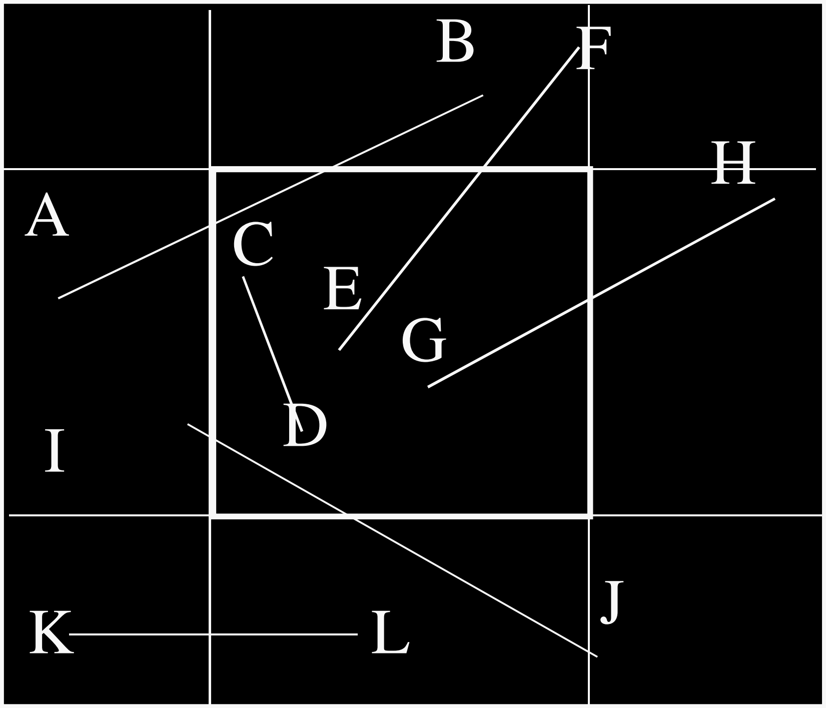
Algoritmos de recorte

Algoritmo de Cohen-Sutherland

If y > yf ➔ seta primeiro bit em 1
If y < yi ➔ seta segundo bit em 1
If x > xf ➔ seta terceiro bit em 1
If x < xi ➔ seta quarto bit em 1

Algoritmos de recorte

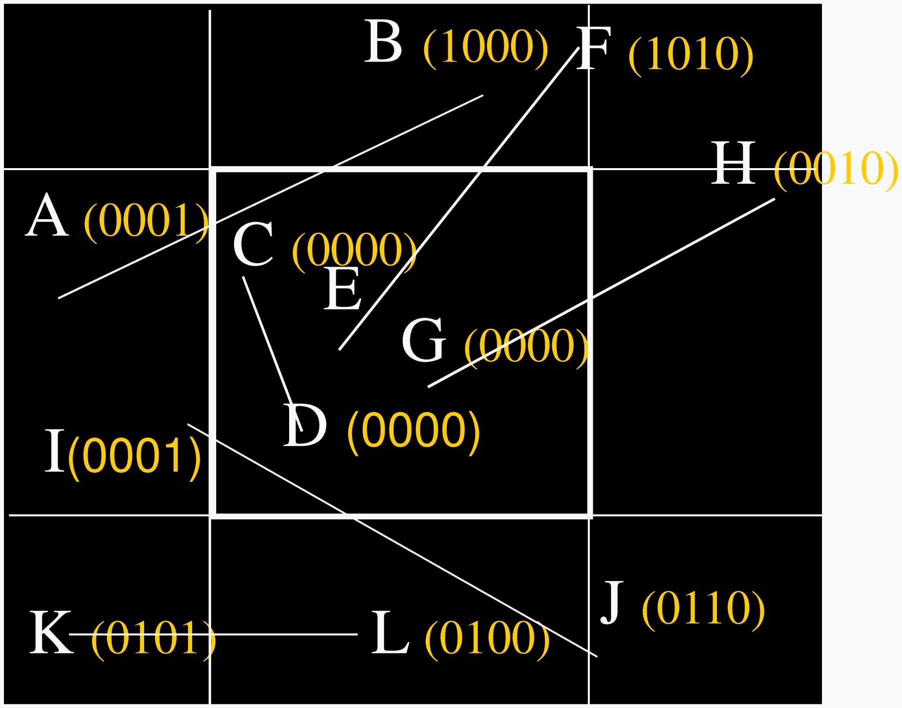
Algoritmo de Cohen-Sutherland



- 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

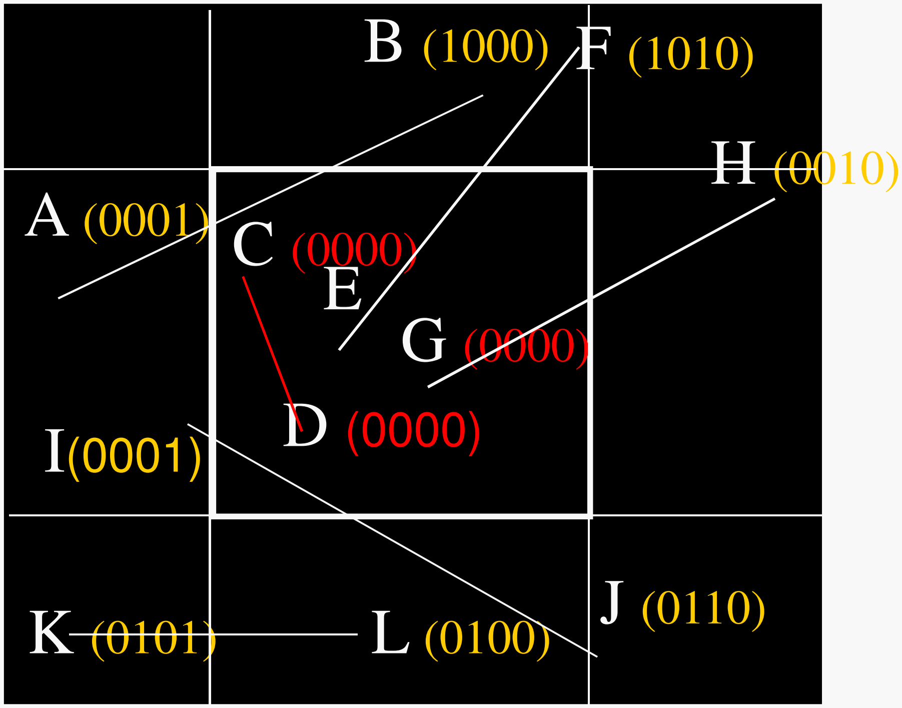
Algoritmo de Cohen-Sutherland


B (1000)
F (1010)
H (0010)
A (0001)
C (0000)
E
G (0000)
D (0000)
I(0001)
J (0110)
K (0101)
L (0100)

- Bit codes dos pontos?

Algoritmos de recorte

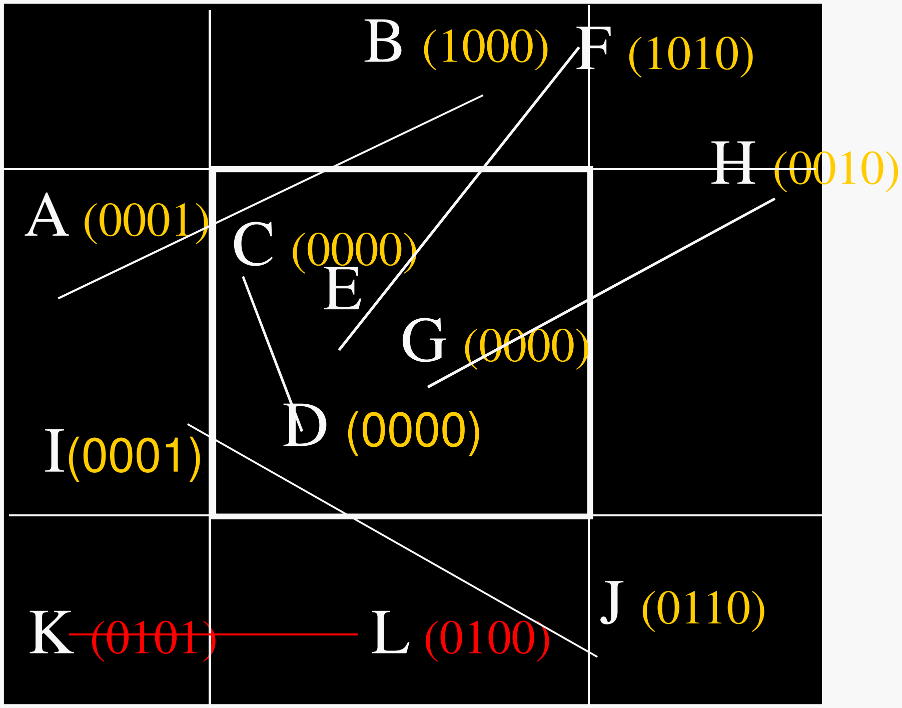
Algoritmo de Cohen-Sutherland


B (1000)
F (1010)
H (0010)
A (0001)
C (0000)
E
G (0000)
D (0000)
I(0001)
J (0110)
K (0101)
L (0100)

- 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

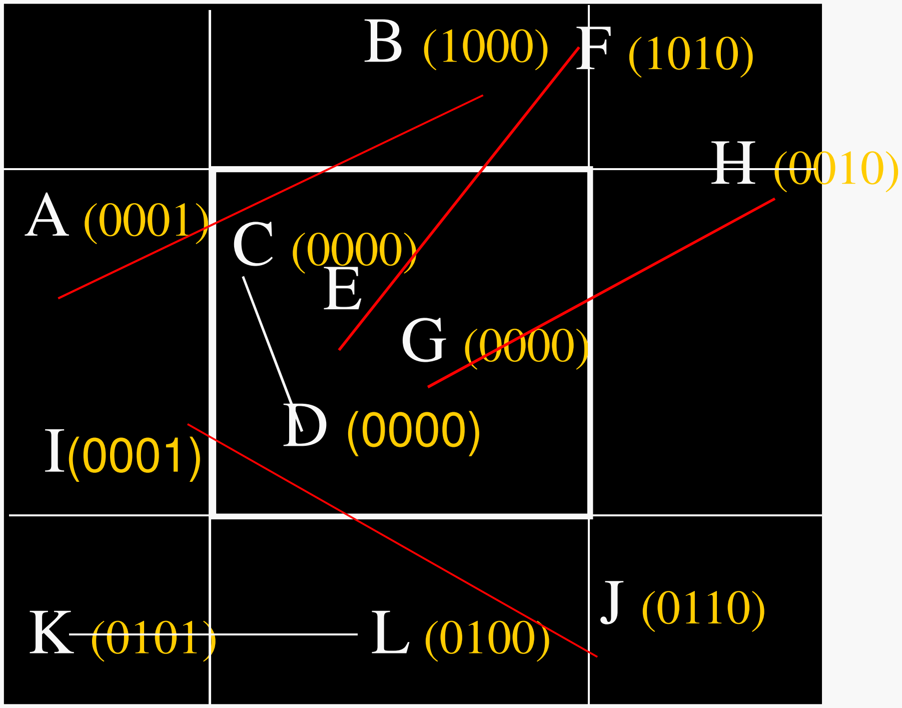
Algoritmo de Cohen-Sutherland


B (1000)
F (1010)
H (0010)
A (0001)
C (0000)
E
G (0000)
D (0000)
I(0001)
J (0110)
K (0101)
L (0100)

- 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 != 0

Algoritmos de recorte

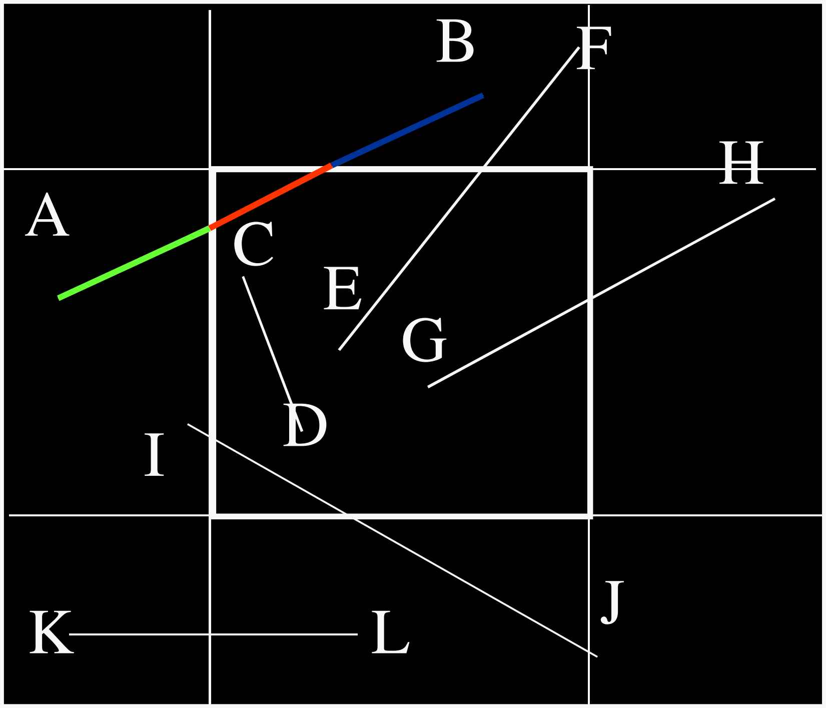
Algoritmo de Cohen-Sutherland


B (1000)
F (1010)
H (0010)
A (0001)
C (0000)
E
G (0000)
D (0000)
I(0001)
J (0110)
K (0101)
L (0100)

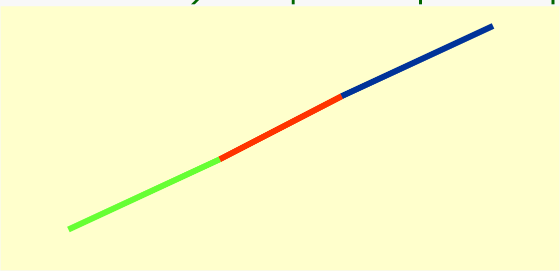
- Como devem ser os bit codes dos pontos trivialmente aceitos? 0000
- Como devem ser os bit codes das arestas trivialmente recusadas? AND entre end points deve ser != 0
- O que fazer com os que não são nem aceitos nem recusados?

Algoritmos de recorte

Algoritmo de Cohen-Sutherland

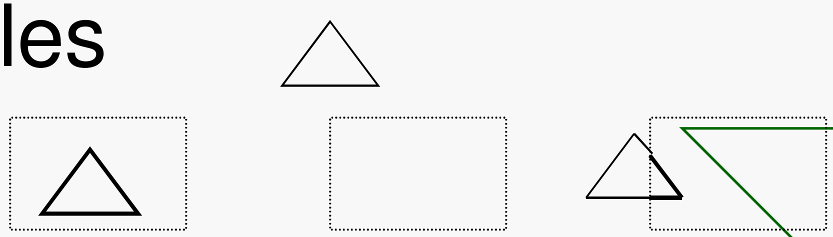

B
F
H
A
C
E
G
D
I
J
K
L

- Calcular segmentos através das intersecções

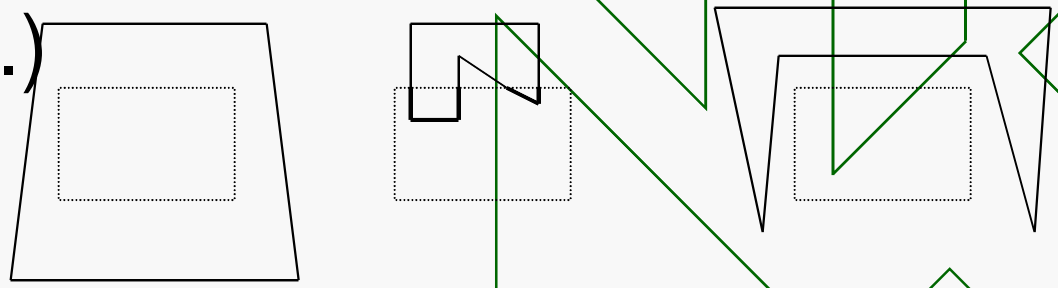


Recorte de Polígono contra Retângulo

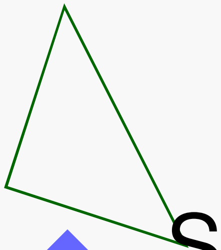
- Casos Simples



- Casos Complicados (pode mudar topologia...)



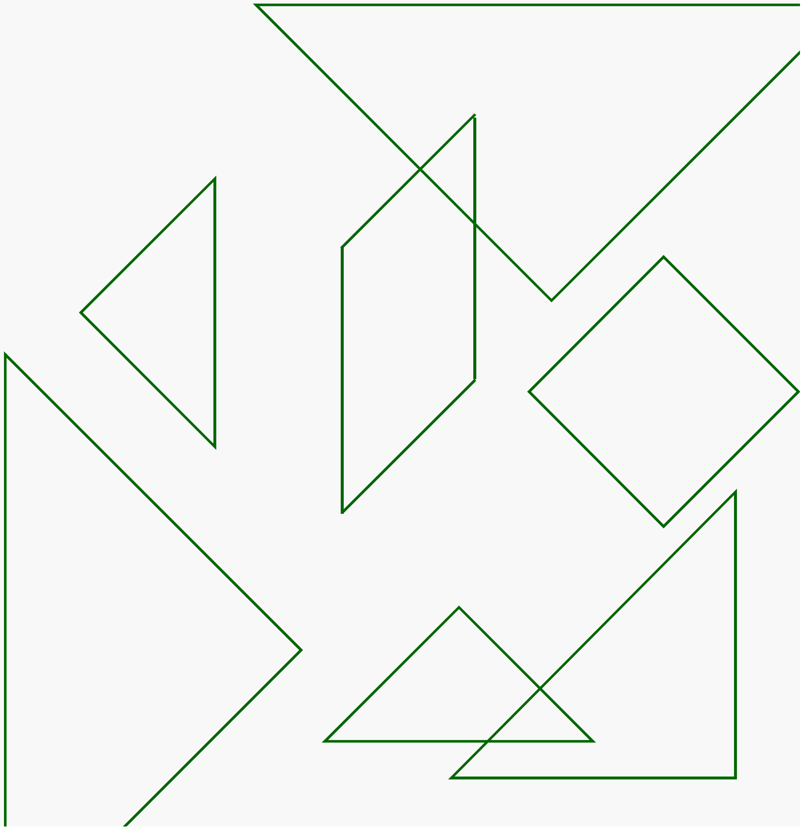
SR´s 2D



- SRO
- SRU
- SRW

(recorte 2D)

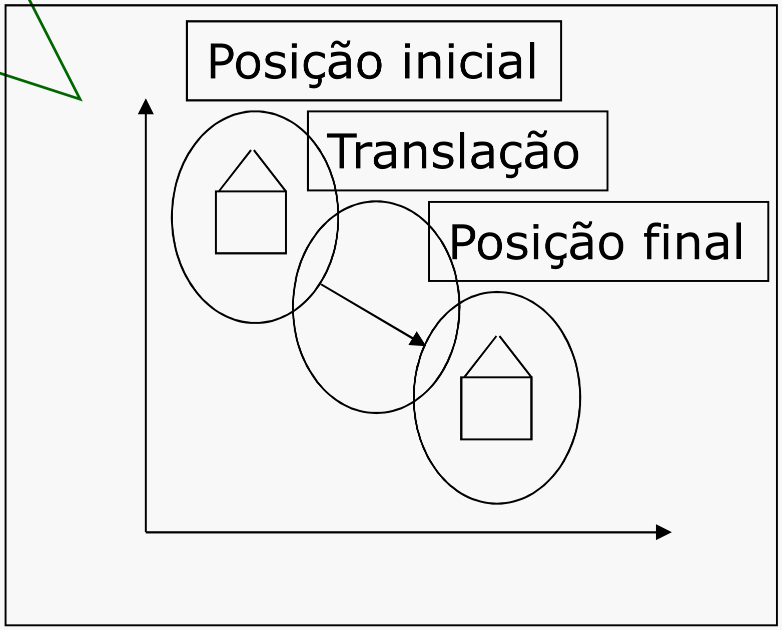
- SRV
- SRD



SR´s 3D

- SRO
- SRU
- SRC
  (recorte 3D)
- SRP
- SRV-SRD

Transformações 2D - Translação

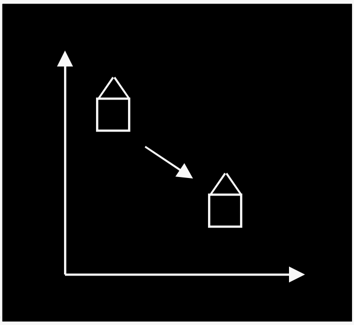


Posição inicial

Translação

Posição final

Transformações 2D - Translação



- Cada vértice é modificado

$$x' = x + t_x$$
$$y' = y + t_y$$

- Utiliza-se vetores para representar a transformação

$$\vec{t} = \begin{bmatrix} tx \\ ty \end{bmatrix}$$

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

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

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

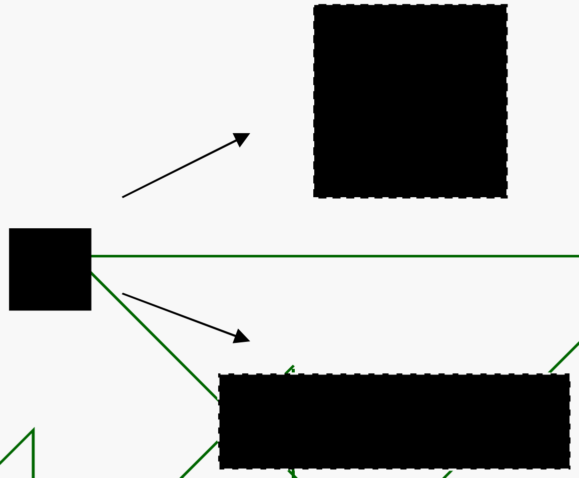
$$\vec{p}' = \vec{p} + \vec{t}$$

Transformações 2D - Escala

- Coordenadas são multiplicadas pelos fatores de escala

$$x' = x.s_x$$

$$y' = y.s_y$$



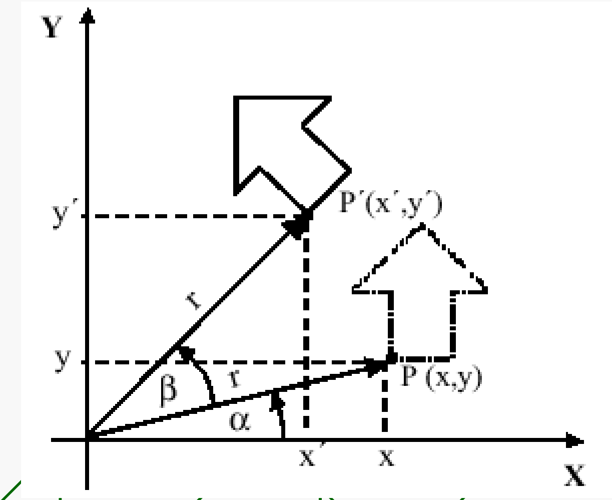
- Tipos de Escala
  - Uniforme:
    - sx = sy
  - Não-Uniforme
    - sx <> sy

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

- Escala é uma multiplicação de matrizes

Transformações 2D - Rotação

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


Y
y´
P´(x´,y´)
r
y
β
r
P (x,y)
α
x´
x
X

Coordenadas Homogêneas

- Introduzida em Matemática

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

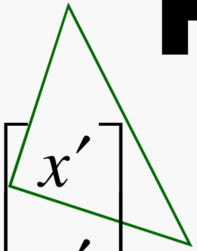
- Adiciona uma terceira coordenada *w*
- Um ponto 2D passa a ser um vetor com 3 coordenadas
- 2 pontos são iguais se e somente se:

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

- Homogeneizar: dividir por w
- Pontos homogeneizados:

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

Translação – Coord. Homogêneas

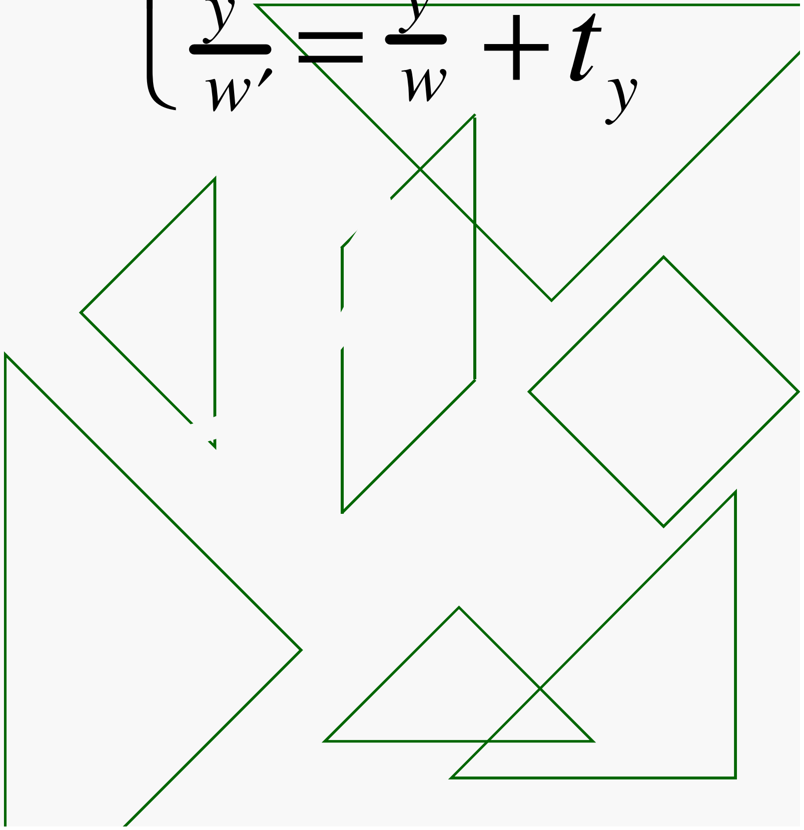


$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

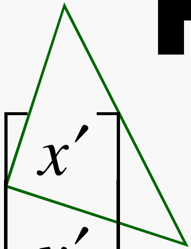
$$\vec{p}' = \vec{p} + \vec{t}$$

$$\begin{cases} \dfrac{x'}{w'} = \dfrac{x}{w} + t_x \\ \dfrac{y'}{w'} = \dfrac{y}{w} + t_y \end{cases}$$

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



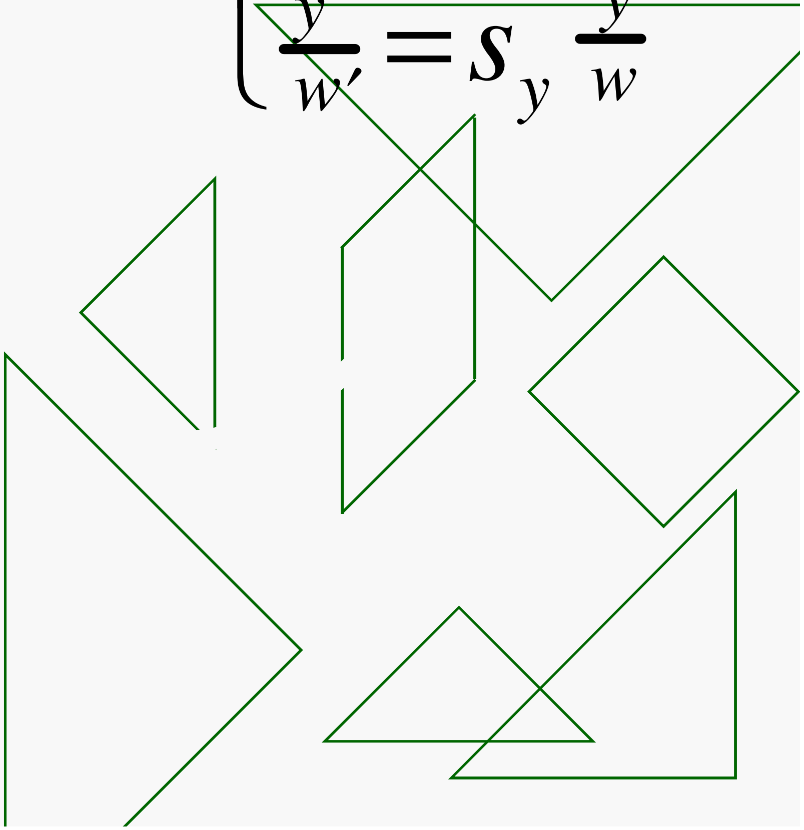
Escala – Coord. Homogêneas



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

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

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

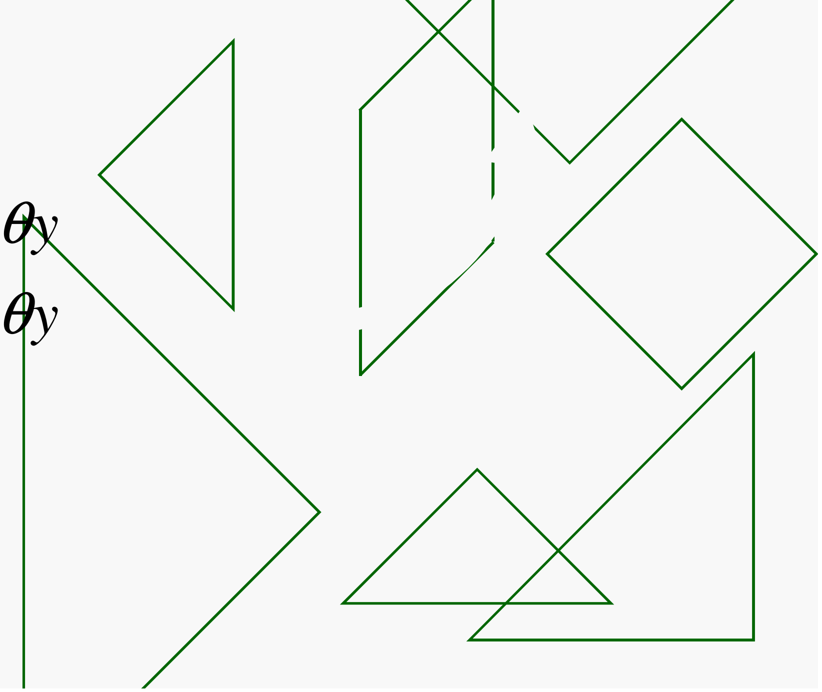


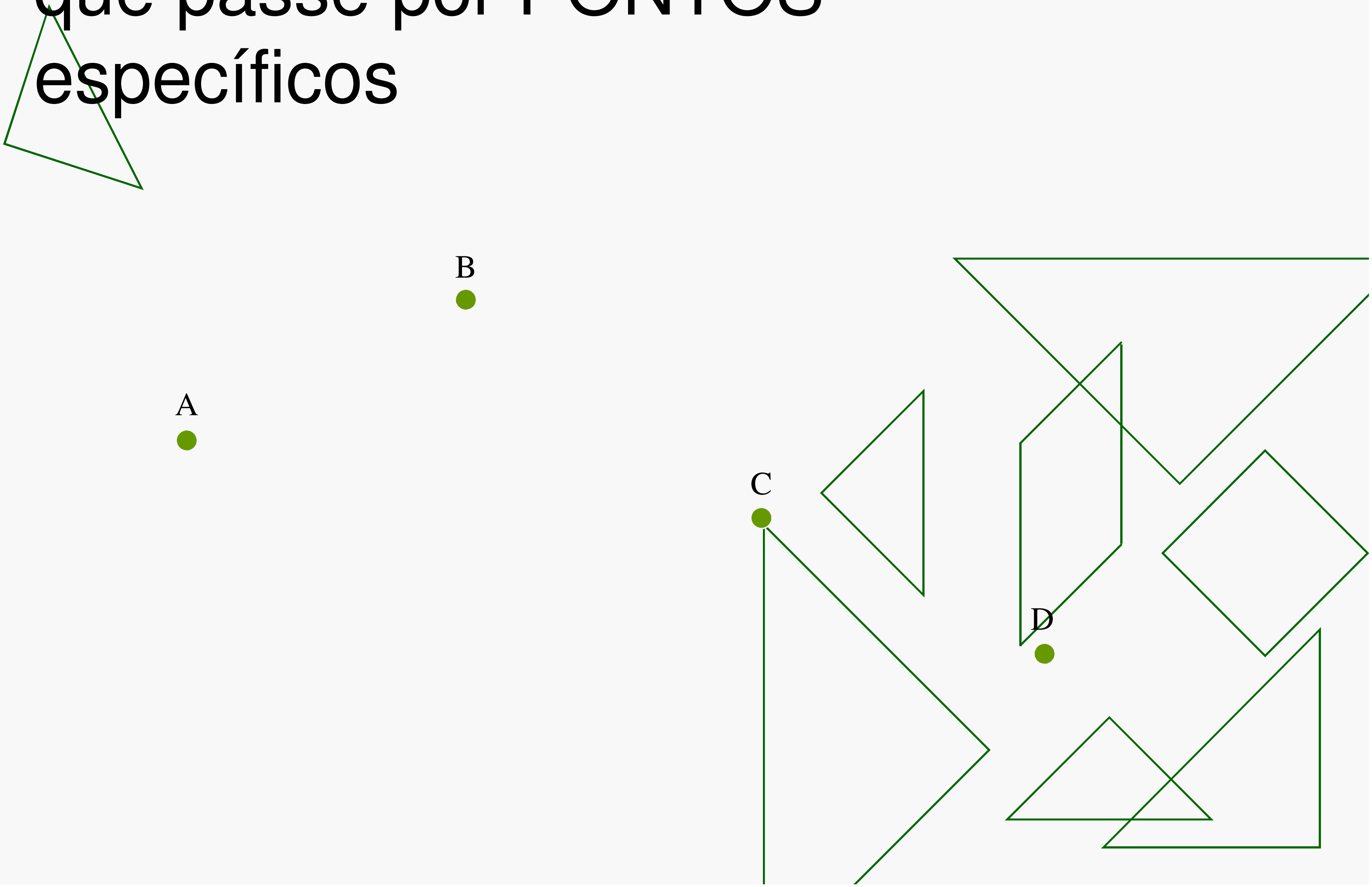
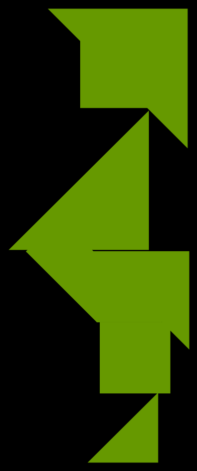
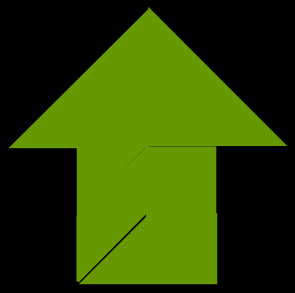
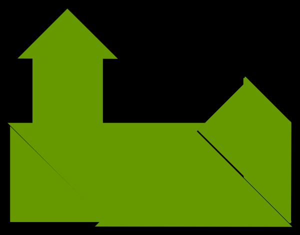
Rotação – Coord. Homogêneas

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

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

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



Problema: gerar uma curva SUAVE que passe por PONTOS específicos