
PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
DEPARTAMENTO DE COMPUTAÇÃO
COMPUTAÇÃO GRÁFICA CMP 1170 – 2019/1
PROF. MSC. GUSTAVO VINHAL

Aula 12

Realismo Visual e Iluminação

Realismo Virtual

- Criar imagens é o objetivo final da Computação Gráfica.
- Realismo virtual são conjuntos de técnicas de tratamento computacional aplicadas a objetos sintéticos com o objetivo de deixá-los mais próximos da realidade.
- Duas etapas:
 - Estáticas: cenas sem movimentos
 - Dinâmicas: animação

Realismo Virtual

- Aplicações:
 - Modelagem de apartamentos;



Realismo Virtual

- Aplicações:
 - Filmes;



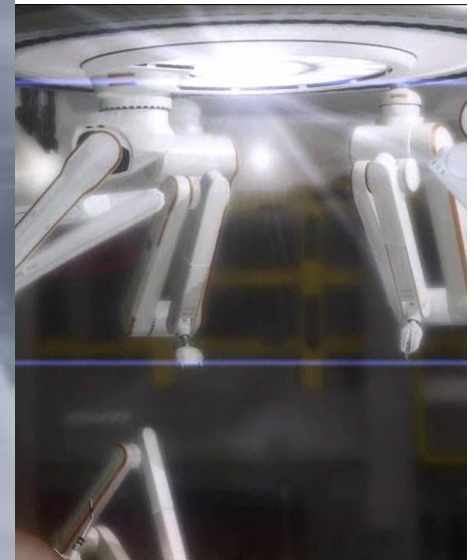
Realismo Virtual

- Aplicações:
 - Propagandas;



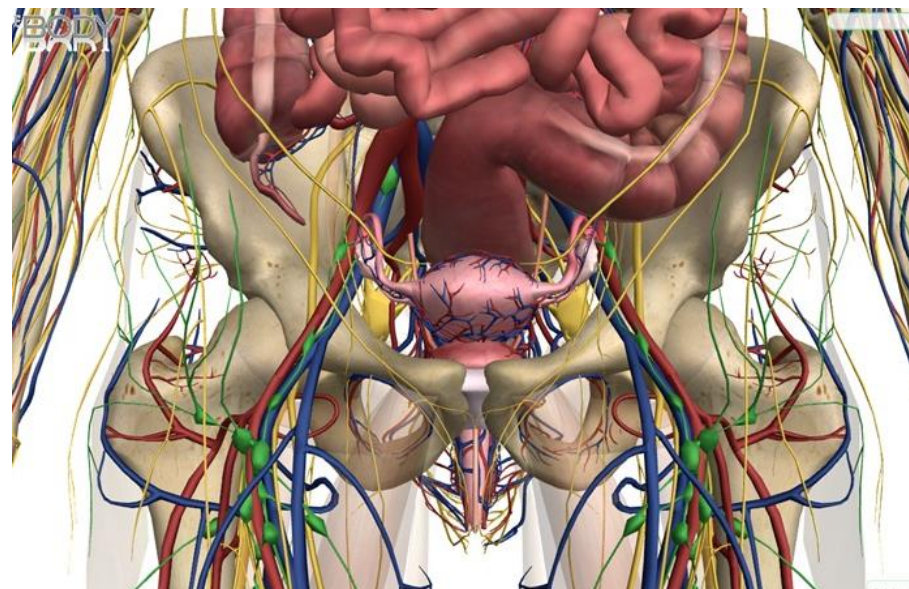
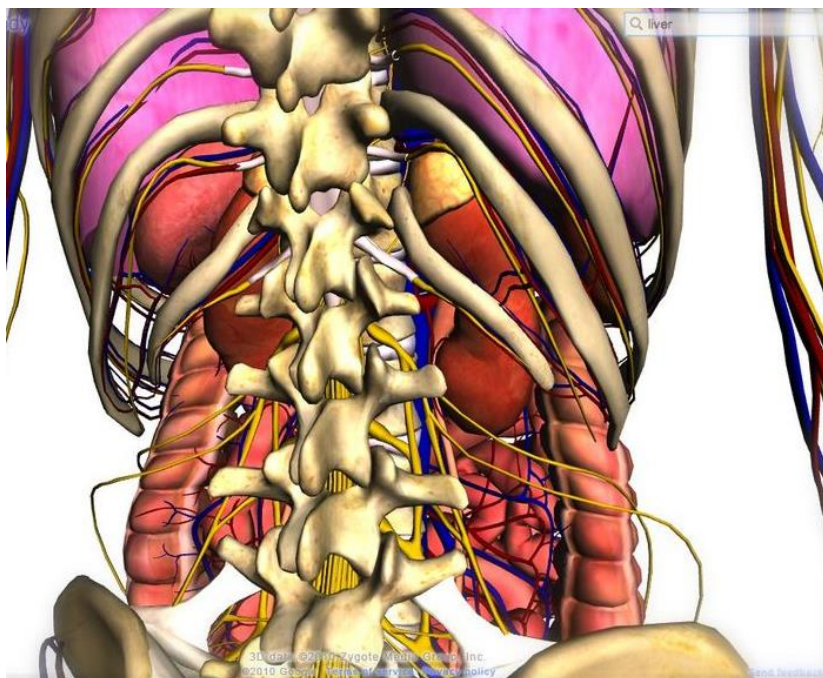
Realismo Virtual

- Aplicações:
 - Jogos;



Realismo Virtual

- Aplicações:
 - Hospitais.



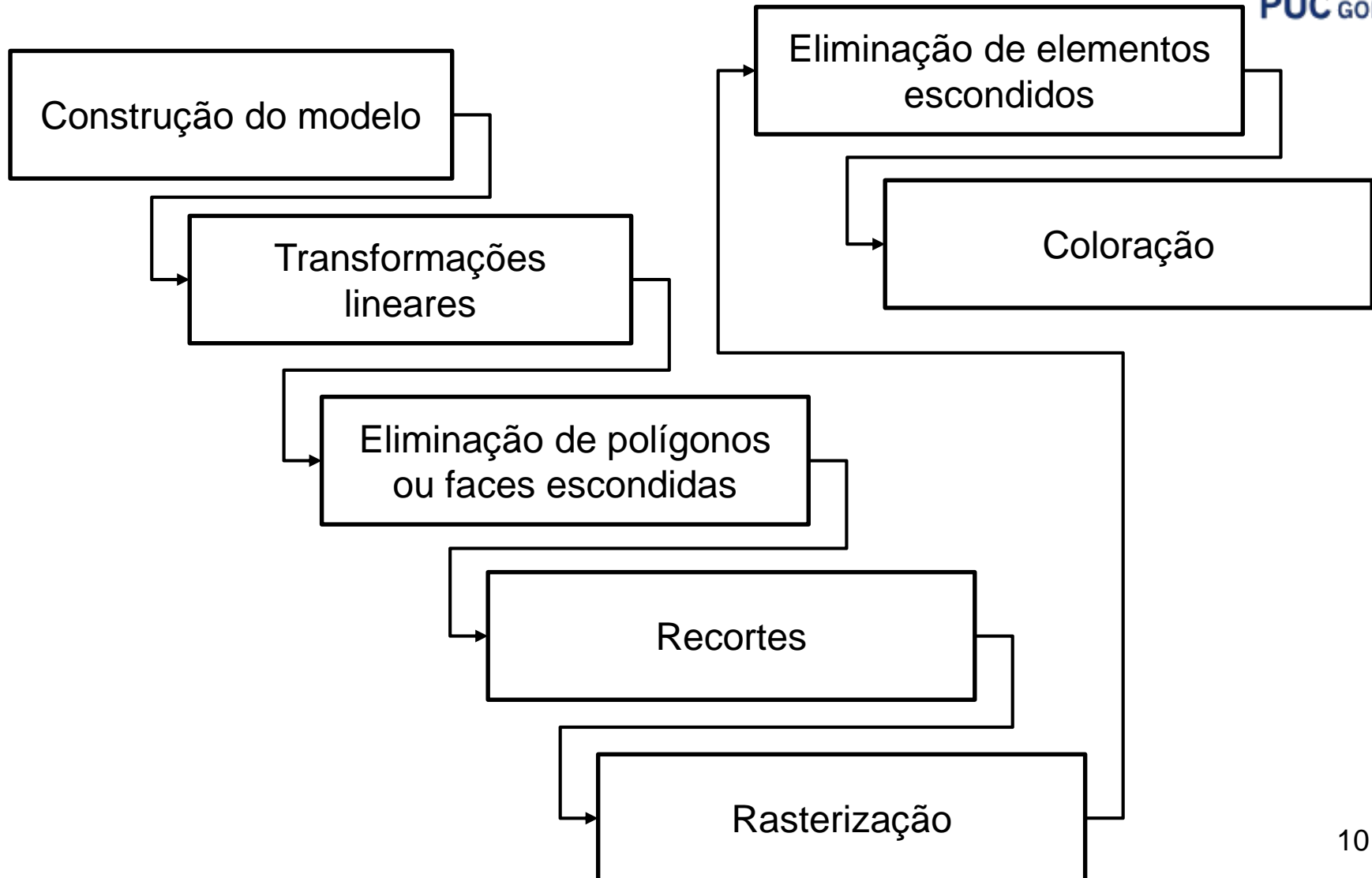
Realismo Virtual

- *Rendering*;
- Rasterização;
- Iluminação;
- Texturas;
- Hiper-Realismo

Rendering

- Computação gráfica = síntese de imagens \neq mostrar um objeto (gerado por um dispositivo) na tela.
- Sintetizar imagens é criar imagens considerando:
 - Suas informações;
 - Geometria da cena;
 - Informações sobre cores e texturas;
 - Iluminação ambiente;
 - Posição de observação da cena.
- *Rendering* é sintetizar um objeto ou cena até eles possuírem uma aparência de algo real.

Rendering

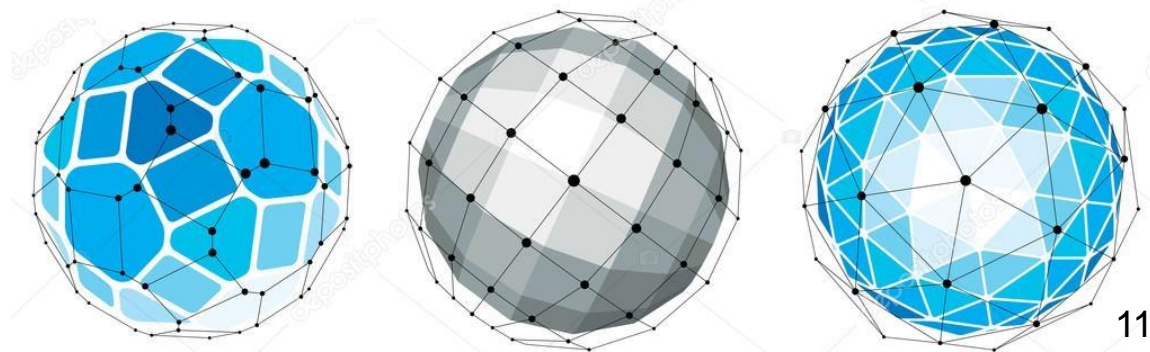
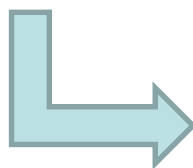


Rendering

Construção do modelo:

- Utiliza alguma técnica de modelagem para geração do objeto com base nas suas informações.

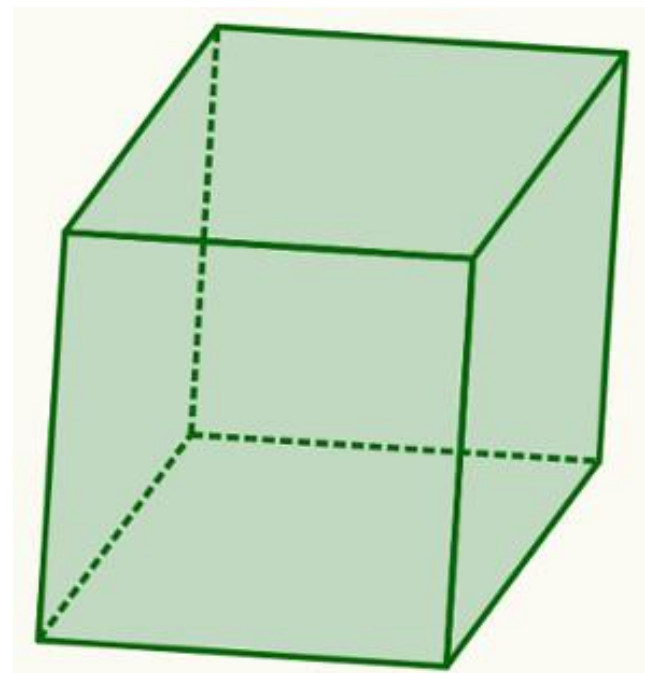
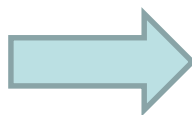
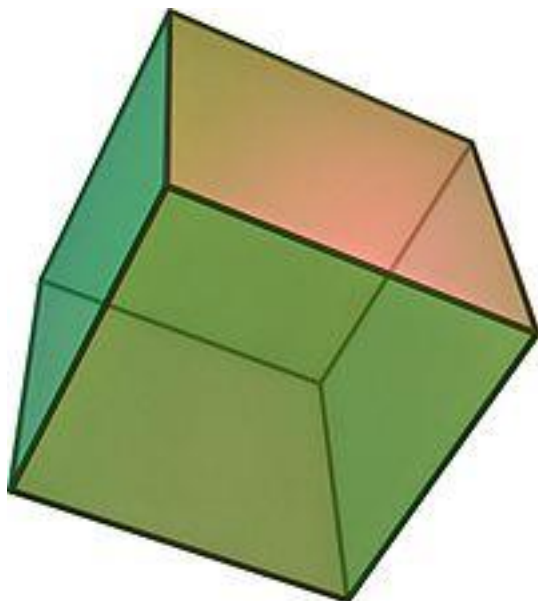
		VÉRTICES							
		A	B	C	D	E	F	G	H
VÉRTICES	A	(6,6,6)	(-2,6,6)		(6,-2,6)	(6,6,-2)			
	B	(2,6,6)	(-6,6,6)	(-6,-2,6)			(-6,6,-2)		
	C		(-6,2,6)	(-6,-6,6)	(2,-6,6)			(-6,-6,-2)	
	D	(6,2,6)		(-2,-6,6)	(6,-6,6)				(6,-6,-2)
	E	(6,6,2)				(6,6,-6)	(-2,6,-6)		(6,-2,-6)
	F		(-6,6,2)			(2,6,-6)	(-6,6,-6)	(-6,-2,-6)	
	G			(-6,-6,2)			(-6,2,-6)	(-6,-6,-6)	(2,-6,-6)
	H				(6,-6,2)	(6,2,-6)		(-2,-6,-6)	(6,-6,-6)



Rendering

Transformações lineares:

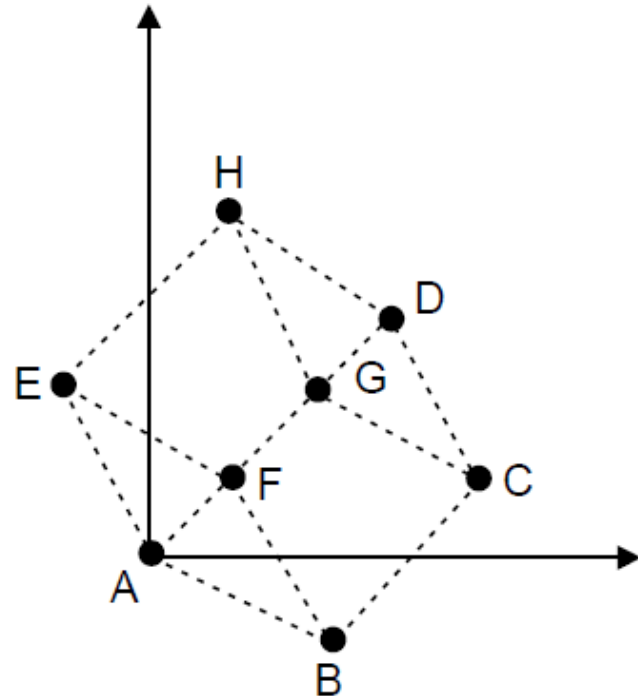
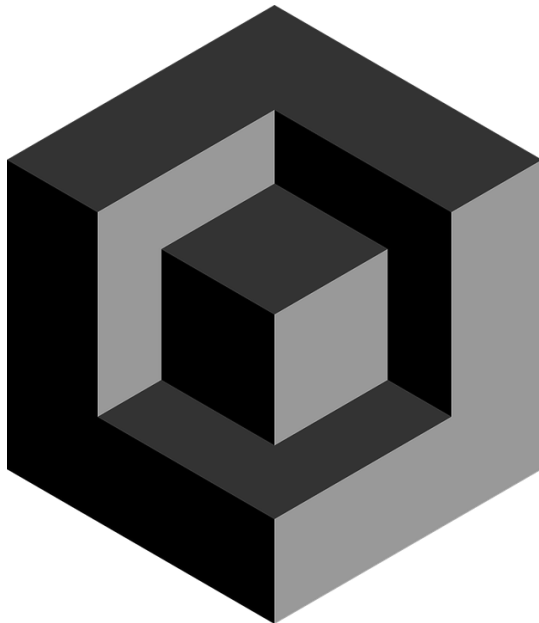
- Aplica transformações lineares nos modelos de modo que ele tenha aparência tridimensional;
- Consiste na utilização de projeções e perspectivas adequadas.



Rendering

Eliminação de polígonos ou faces escondidas:

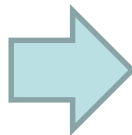
- Devido a posição do objeto em relação ao observador, algumas faces estarão ocultas;
- Essa etapa remove as faces.



Rendering

Recortes (*clipping*) :

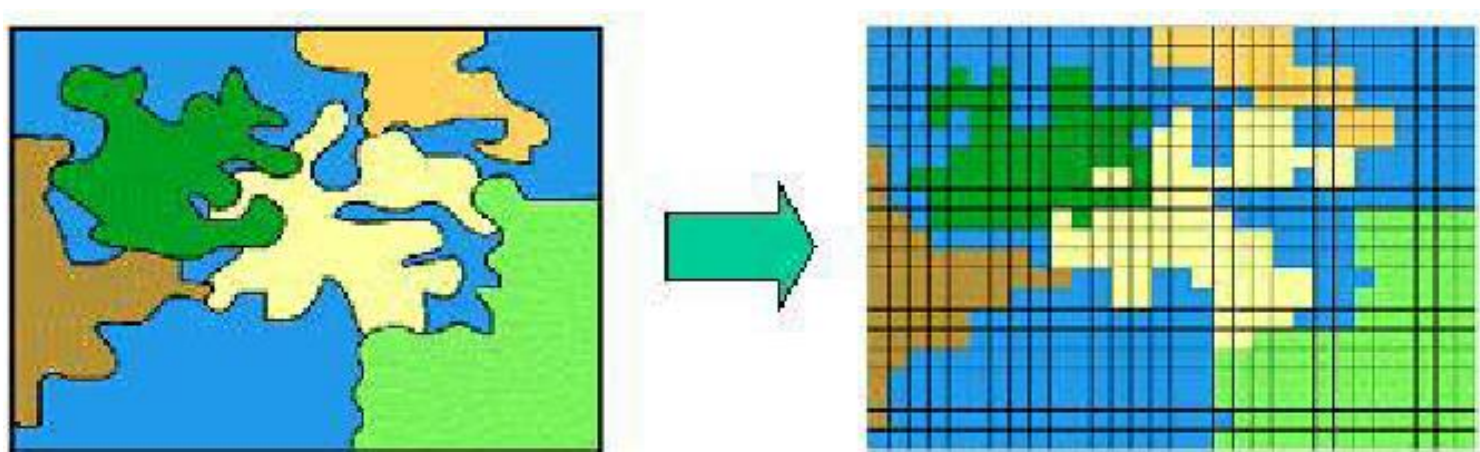
- Partes das cenas que não serão mostradas são desconsideradas.



Rendering

Rasterização:

- Conversão da representação tridimensional para uma bidimensional (3D para 2D);
- Representação em forma de pixels.

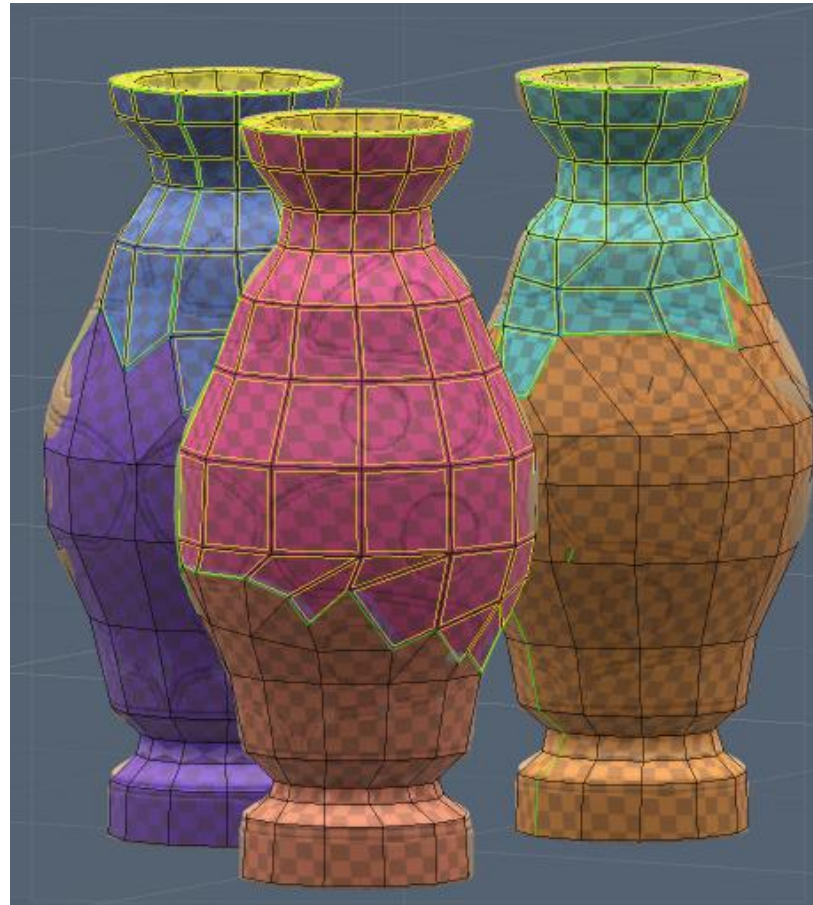


Rendering

Eliminação de elementos escondidos:

- Eliminação de partes de um objetos ocasionados pela obstrução visual de outro objeto;
- As vezes esta etapa é realizada junto com a terceira etapa:

tratamento de partes escondidas.



Rendering

Coloração:

- Colorir cada pixels individualmente;
- Deve-se levar em consideração:
 - Luzes presentes na cena (suas intensidades e direções);
 - Superfície: transparência, brilho, reflexão e textura;
 - Influência de um objeto em outro e sobre a cena (sombras).



Rendering

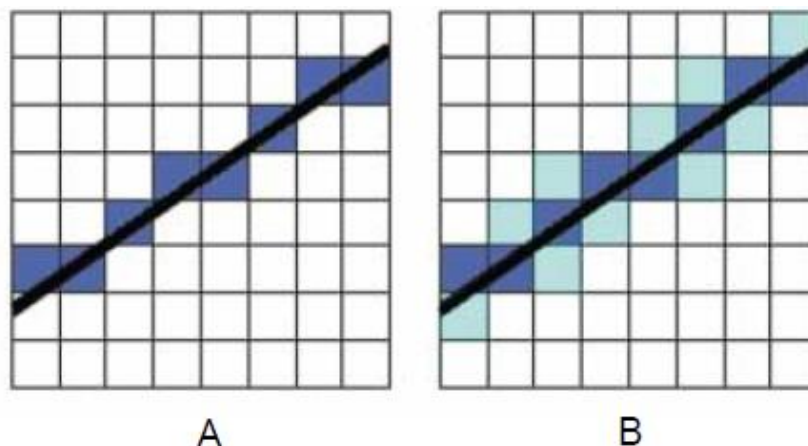
Acabamentos Não-fotográficos

- Renders realísticos tentam fazer uma imagem sintética o mais próximo de uma fotografia;
- Em contrapartida, os não realísticos (NPR – *non-photorealistic rendering*) são aqueles que constroem cenas que se distanciam das reais porém mantendo a essência da cena.



Rasterização

- Conversão da representação tridimensional para uma bidimensional (3D para 2D);
- Representação em forma de pixels.
- Transformação de tempo contínuo para tempo discreto.



Rasterização

Rasterização de retas – Algoritmo de Bresenham para traçado de linhas

- O algoritmo de Bresenham considera como dados de entrada os dois pixels $(x_1, y_1)(x_2, y_2)$ da reta a ser rasterizada;
- Para cada ponto a ser traçado, o algoritmo verifica sua distância entre a posição do próximo pixel e a localização da reta no grid.

```
x = x1
y = y1
Δx = x2 - x1
Δy = y2 - y1
m = Δy/Δx
e = m - 1/2

for i = 1 to Δx do
    desenhaPonto(x,y)
    while e ≥ 0 do
        y = y + 1
        e = e - 1
    end while
    x = x + 1
    e = e + m
end for
```


Rasterização

Rasterização de polígonos

- Os polígonos são rasterizados, primeiramente, rasterizando todos os seus lados;

Se $\text{abs}(x_2 - x_1) \geq \text{abs}(y_2 - y_1)$ então

Tamanho = $\text{abs}(x_2 - x_1)$

else

Tamanho = $\text{abs}(y_2 - y_1)$

end if

$\Delta x = (x_2 - x_1) / \text{Tamanho}$

$\Delta y = (y_2 - y_1) / \text{Tamanho}$

$i = 1$

Enquanto $i \leq \text{Tamanho}$ faça

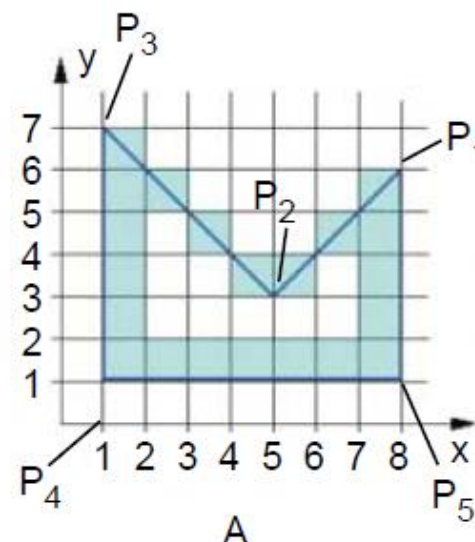
DesenhaPonto($\text{Arredonda}(x)$, $\text{Arredonda}(y)$) {Arredonda: valor arredondado de x e y }

$x = x + \Delta x$

$y = y + \Delta y$

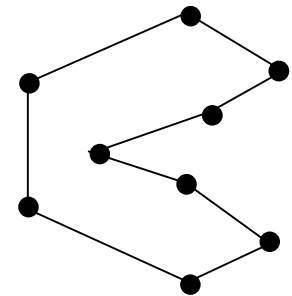
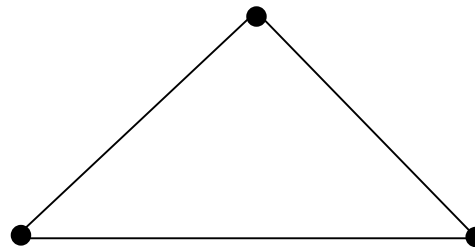
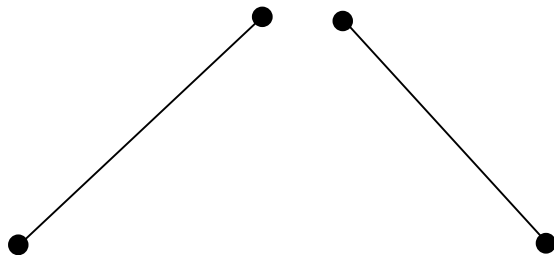
$i = i + 1$

Fim Enquanto



Exercícios

- Aplicando as técnicas de rasterização, rasterize as seguintes formas utilizando:
 - a. Uma matriz 50x50 com pixel de largura 1;
 - b. Uma matriz 100x100 com pixel de largura 0,5.



REFERÊNCIAS BIBLIOGRÁFICAS:

AZEVEDO, Eduardo; CONCI, Aura. **Computação gráfica: teoria e prática**. Rio de Janeiro: Campus, 2003.