



UFC

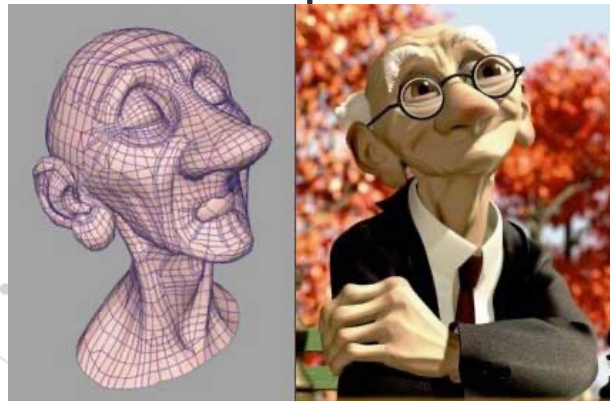
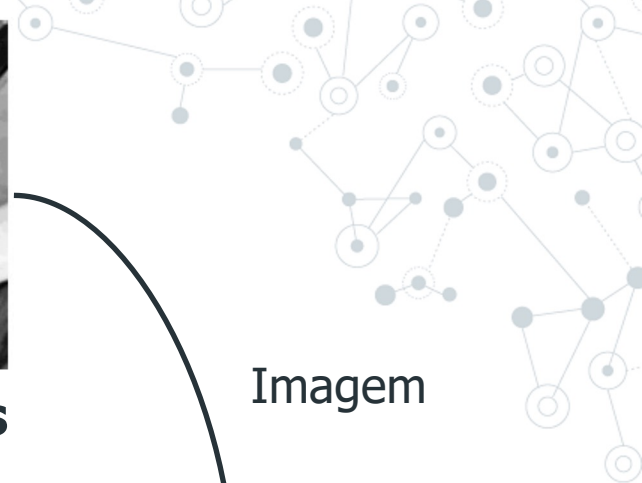
Introdução

Prof. Iális Cavalcante

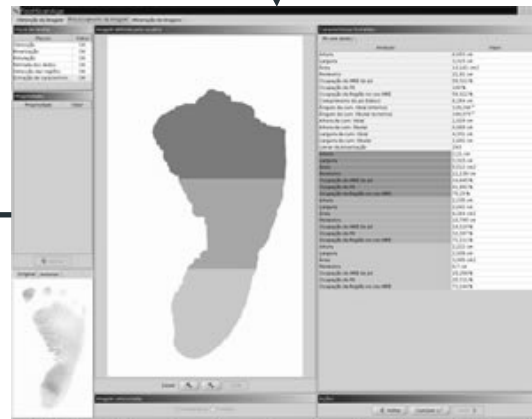
Imagem

Processamento de Imagens

Imagem



Dado



Computação Gráfica

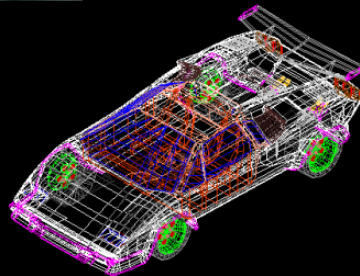
Visão Computacional

O que você acha que é Computação Gráfica?

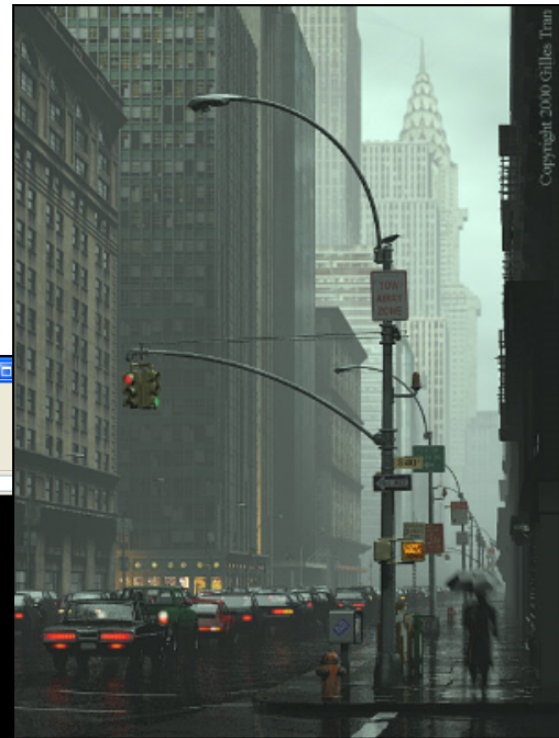


Layer	0
Linetype	CONTINUO
Linetype scale	1
Font Height	1.76979526
Font Name	Arial

Favorites	
Name	Path
Columba.dwg	D:\MyDoc...
Ford Cobra...	D:\MyDoc...
Lamborghini d...	D:\MyDoc...



LAMB([130.0741; -194.5167; 0] 225.751; 146.9765; 114.6424; -0.4216



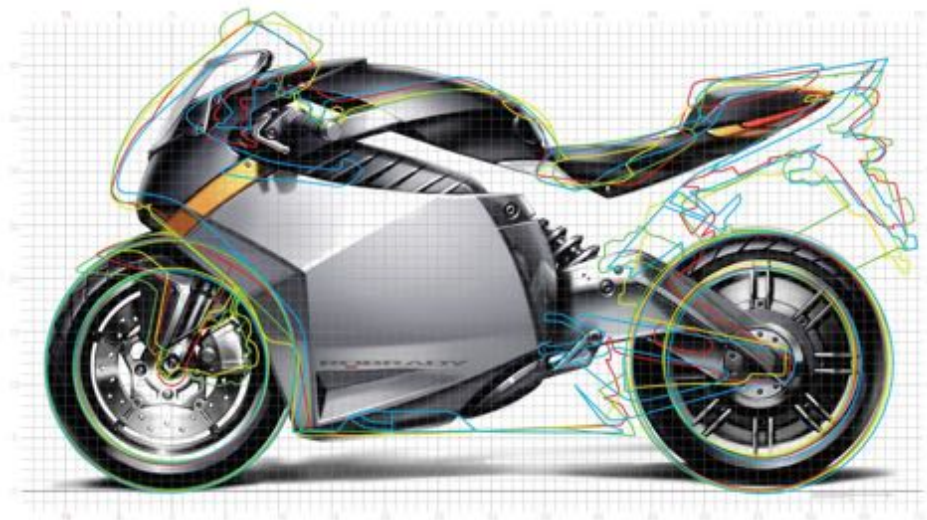
Copyright 2000 Gilles Tarr

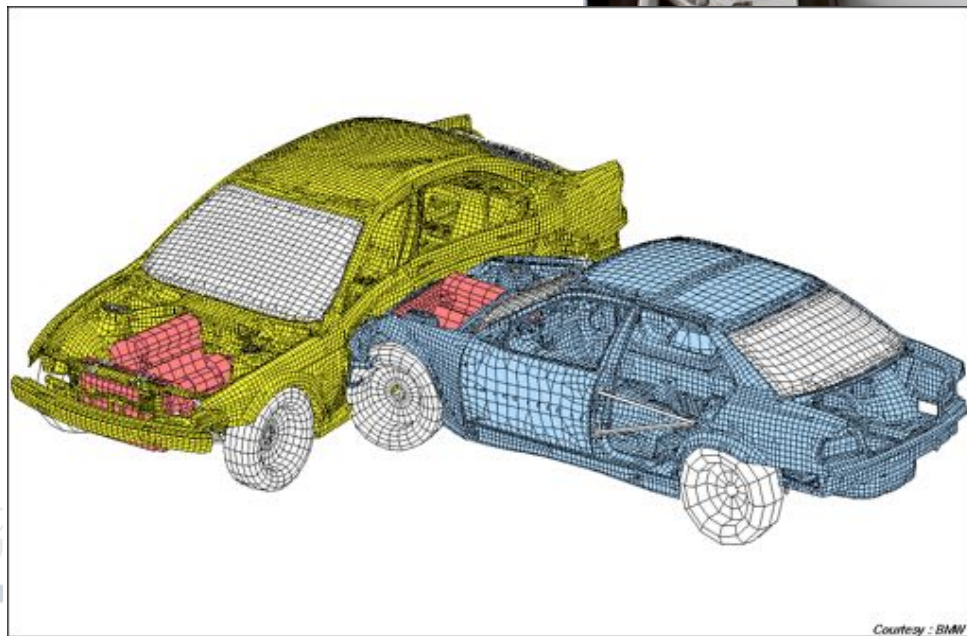
O que define a Computação Gráfica?

- ◎ Técnicas para a geração, exibição, manipulação e interpretação de modelos de objetos e imagens a partir do computador
- ◎ Modelos e imagens criada a partir de dados do mundo real -> converter dados em imagens
- ◎ Usado em diversas área e com diferentes aplicações

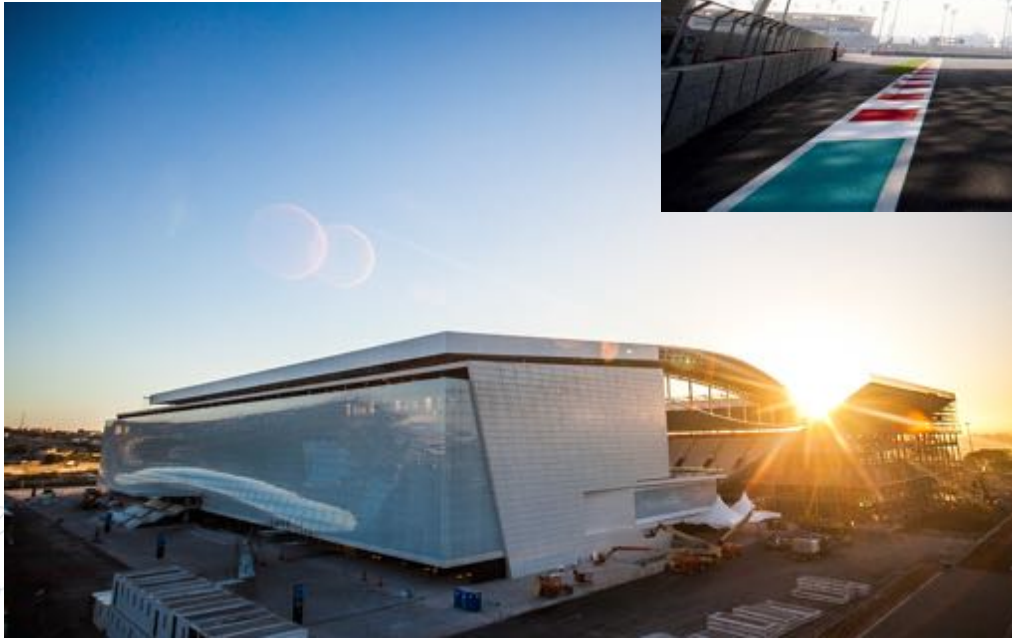


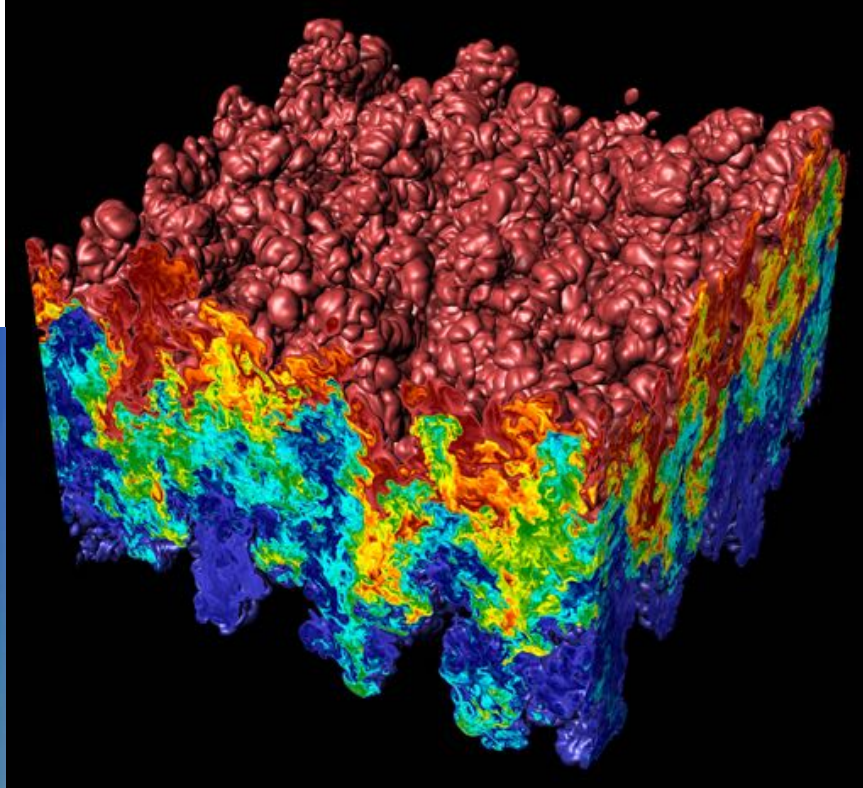
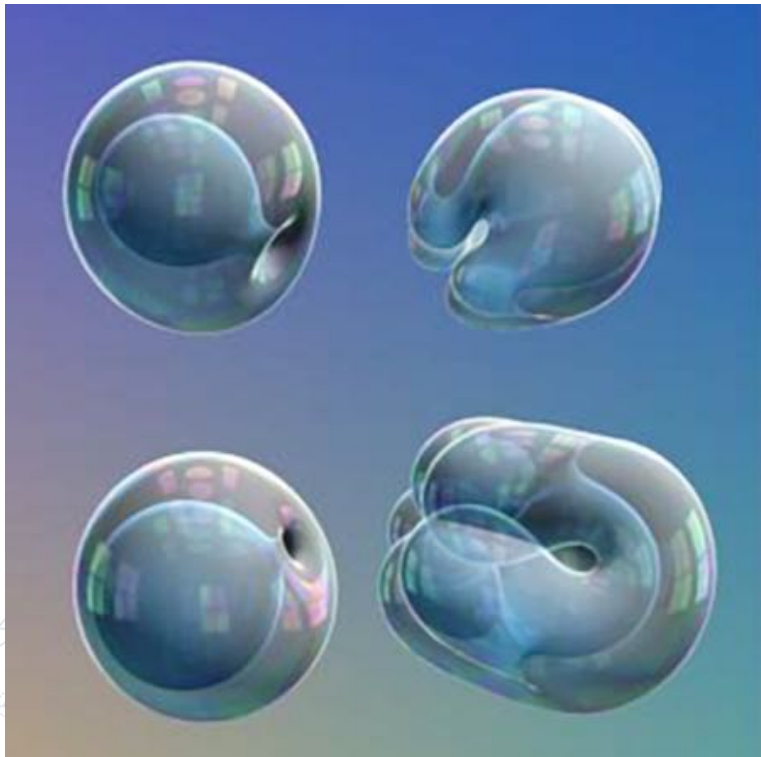


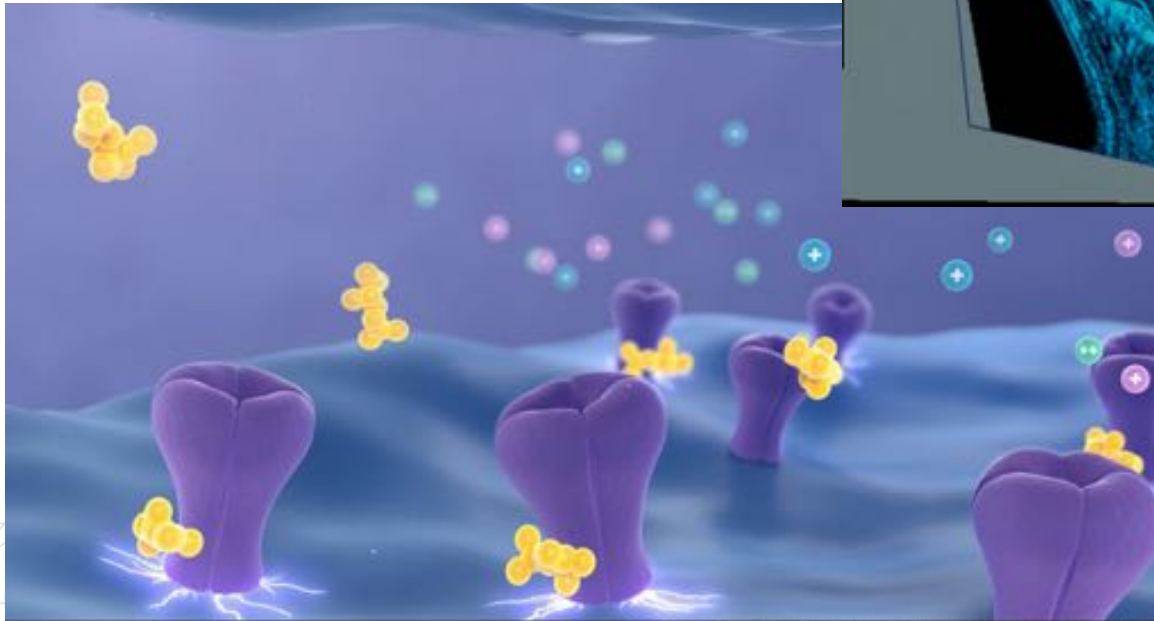
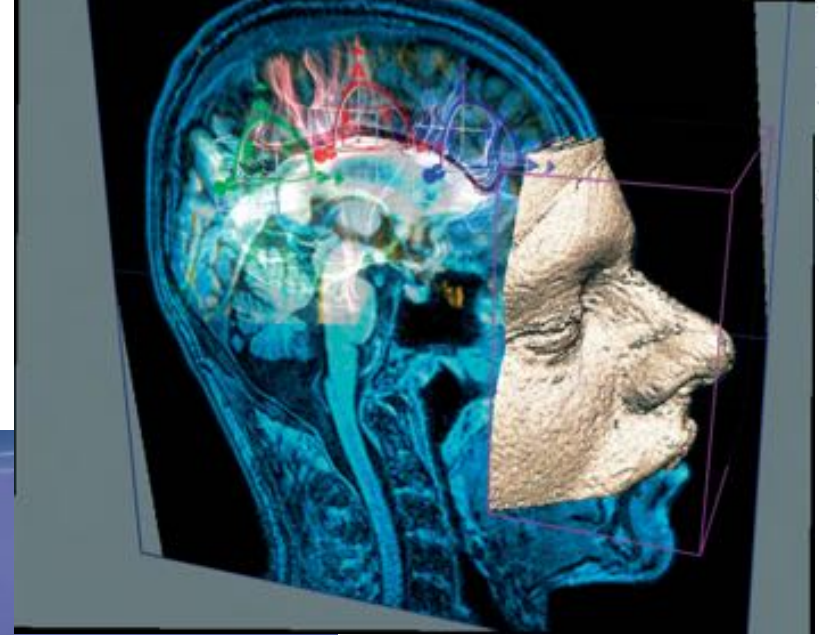


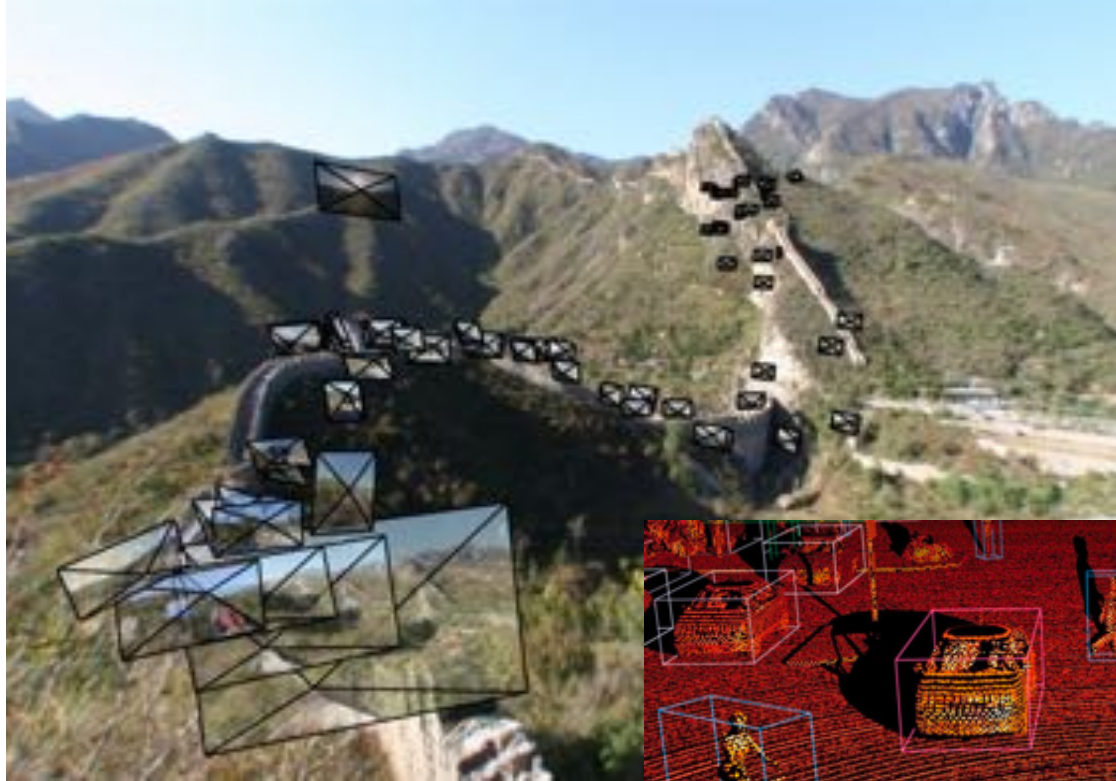


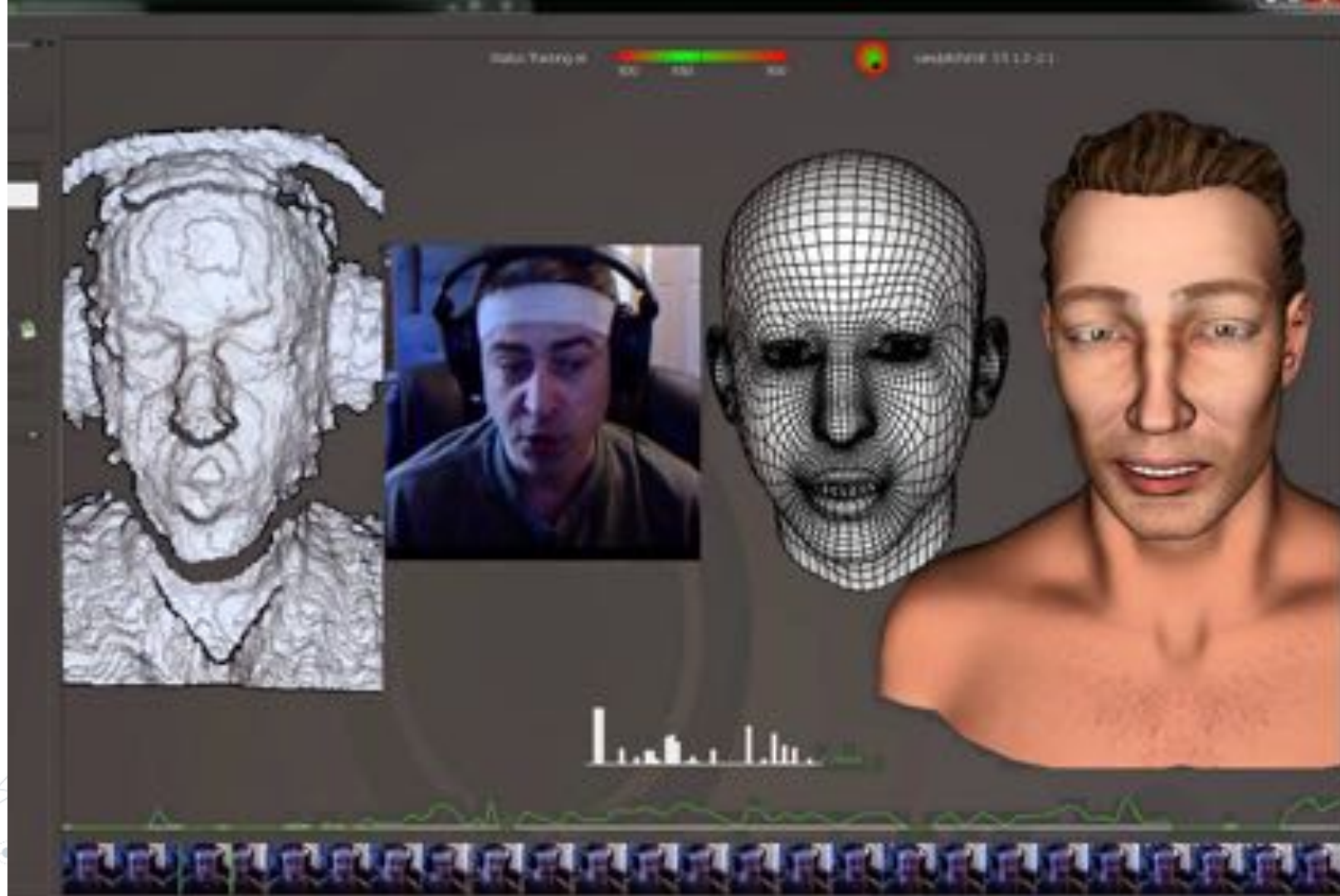
Courtesy : BMW











Fundamentos da computação gráfica

◎ Todas essas aplicações exigem **teorias** e sistemas sofisticados

◎ Teoria

- Representações básicas (como você codifica digitalmente uma forma, um movimento?)
- Amostragem e aliasing (como você adquire e reproduz um sinal?)
- Métodos numéricos (como você manipula os sinais numericamente?)

Radiometria e transporte de luz (como a luz se comporta?)

Percepção (como tudo isso se relaciona com os humanos?)

Fundamentos da computação gráfica

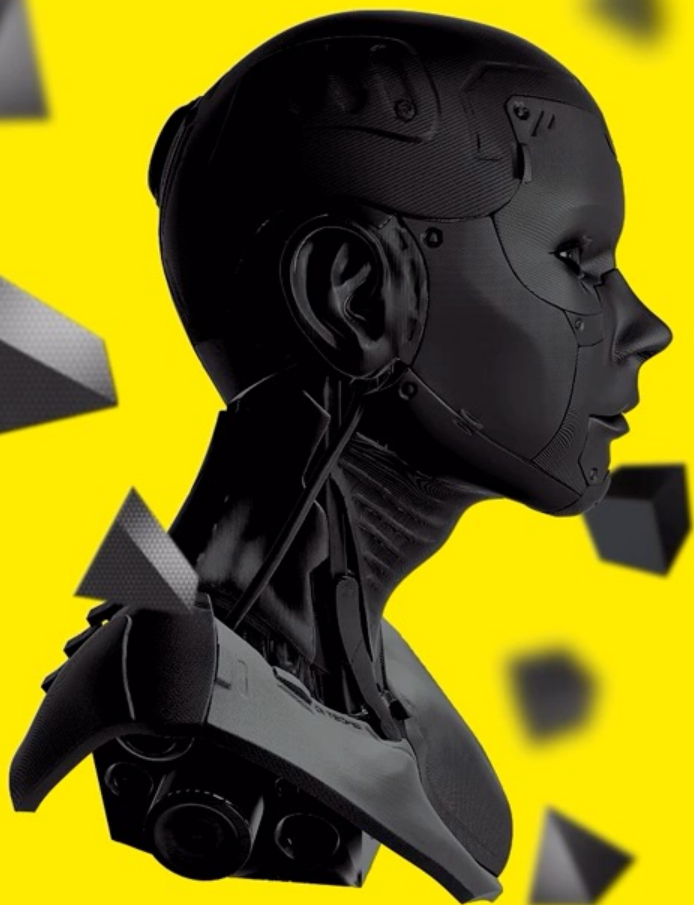
- © Todas essas aplicações exigem teorias e **sistemas** sofisticados
- © **Sistemas**
 - processamento paralelo e heterogêneo
 - linguagens de programação com especificações gráficas





SIGGRAPH 2021

→ THE PREMIER **CONFERENCE & EXHIBITION** IN
COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES



Reflita e defina... Como modelar um cubo?

- ◎ Suponha um cubo definido como...
 - Centralizado na origem $(0,0,0)$
 - Possui dimensões $2 \times 2 \times 2$
 - Bordas alinhadas nos eixos $x/y/z$
- ◎ Quais as coordenadas dos vértices dos cubos?
- ◎ Quais os segmentos de reta que definem essas bordas?

Reflita e defina... Como desenhar este cubo?

- ◎ Como desenhemos este cubo 3D como uma imagem 2D (plana)?
- ◎ Estratégia básica:
 - 1. mapeie vértices 3D para pontos 2D na imagem
 - 2. conecte pontos 2D com linhas retas
 - ...Ok mas como?

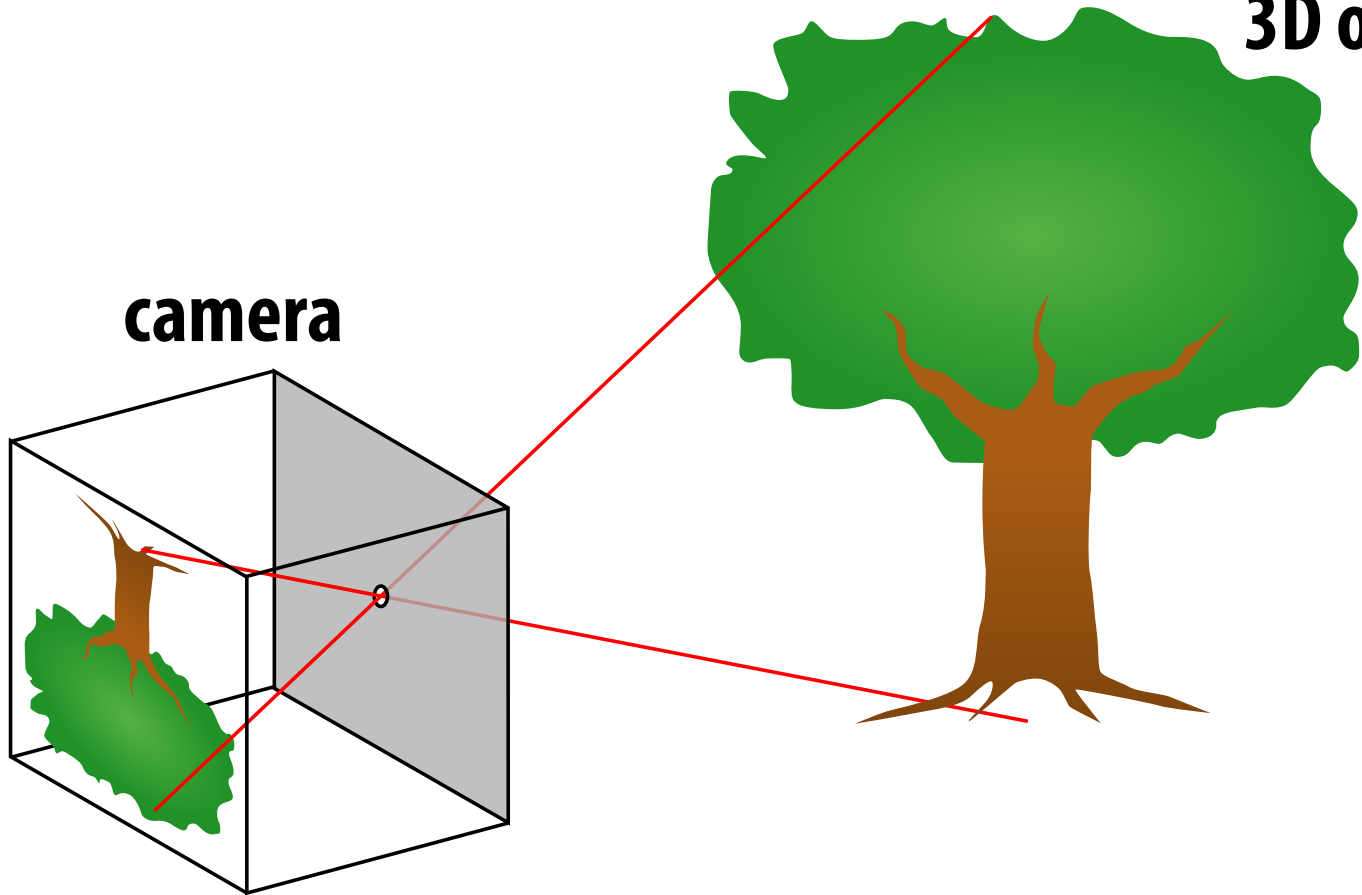
Projeção de Perspectiva

- ◎ Os objetos parecem menores à medida que se afastam (“perspectiva”)
- ◎ Por que isso acontece?
- ◎ Considere o modelo simples (“pinhole”) de uma câmera:

3D object

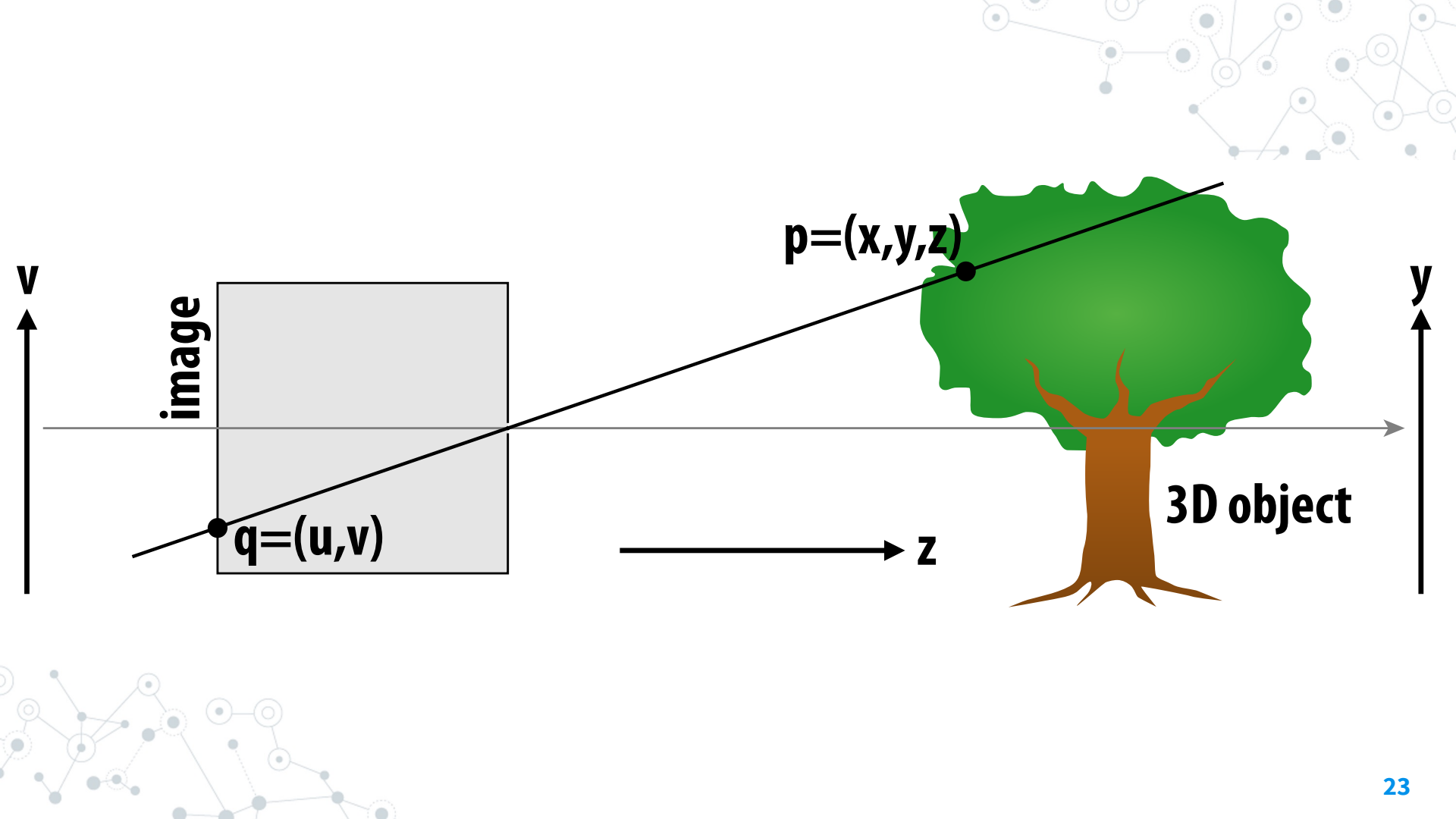
camera

2D image



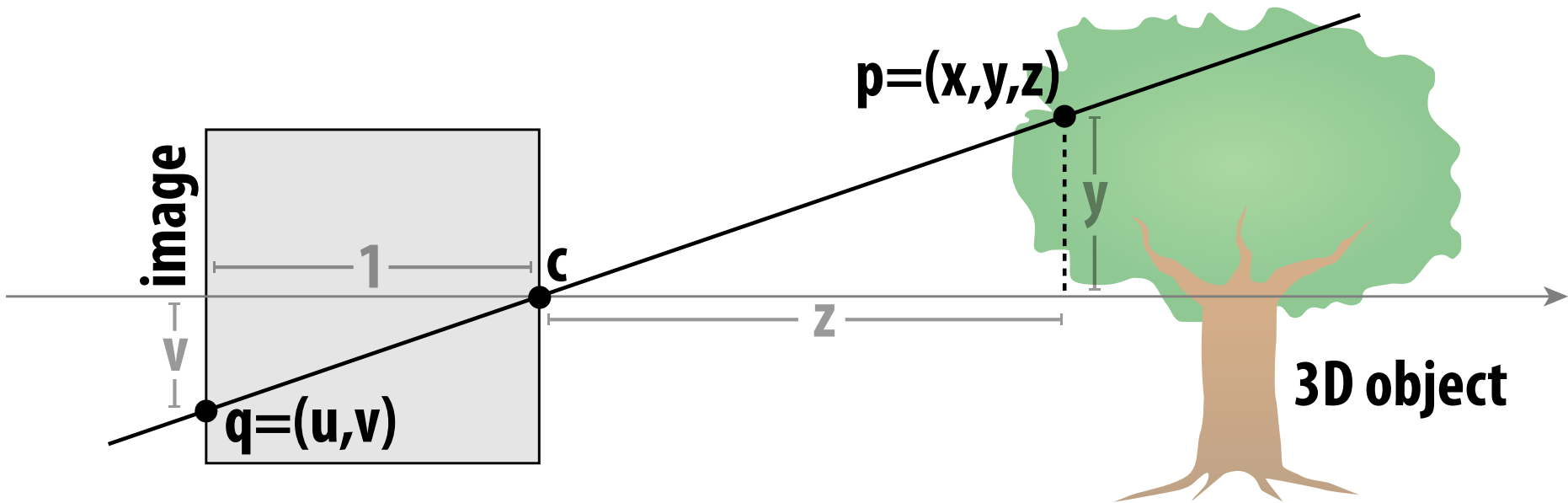
Projeção em perspectiva: visão lateral

- ⊙ Onde exatamente um ponto $p = (x,y,z)$ termina na imagem?
- ⊙ Vamos chamar o ponto da imagem $q=(u,v)$



Projeção em perspectiva: visão lateral

- ◎ Observe dois triângulos semelhantes:
- ◎ Suponha que a câmera tenha tamanho unitário, a origem esteja no *pinhole* c
- ◎ Então $v/1 = y/z$, ou seja, a coordenada vertical é apenas a inclinação y/z
- ◎ Da mesma forma, a coordenada horizontal é $u=x/z$



Reflita e defina... Como desenhar este cubo?

- ◎ Repita o mesmo algoritmo simplesmente 12 vezes
 - Uma vez para cada aresta
 - Suponha que a câmera esteja em $c=(2,3,5)$
 - Converta (X,Y,Z) de ambos os terminais para (u,v) :
 - 1. subtraia a câmera c do vértice (X,Y,Z) para obter (x,y,z)
 - 2. divida (x,y) por z para obter (u,v) —escreva como uma fração

Desenhe uma linha entre (u_1,v_1) e (u_2,v_2)

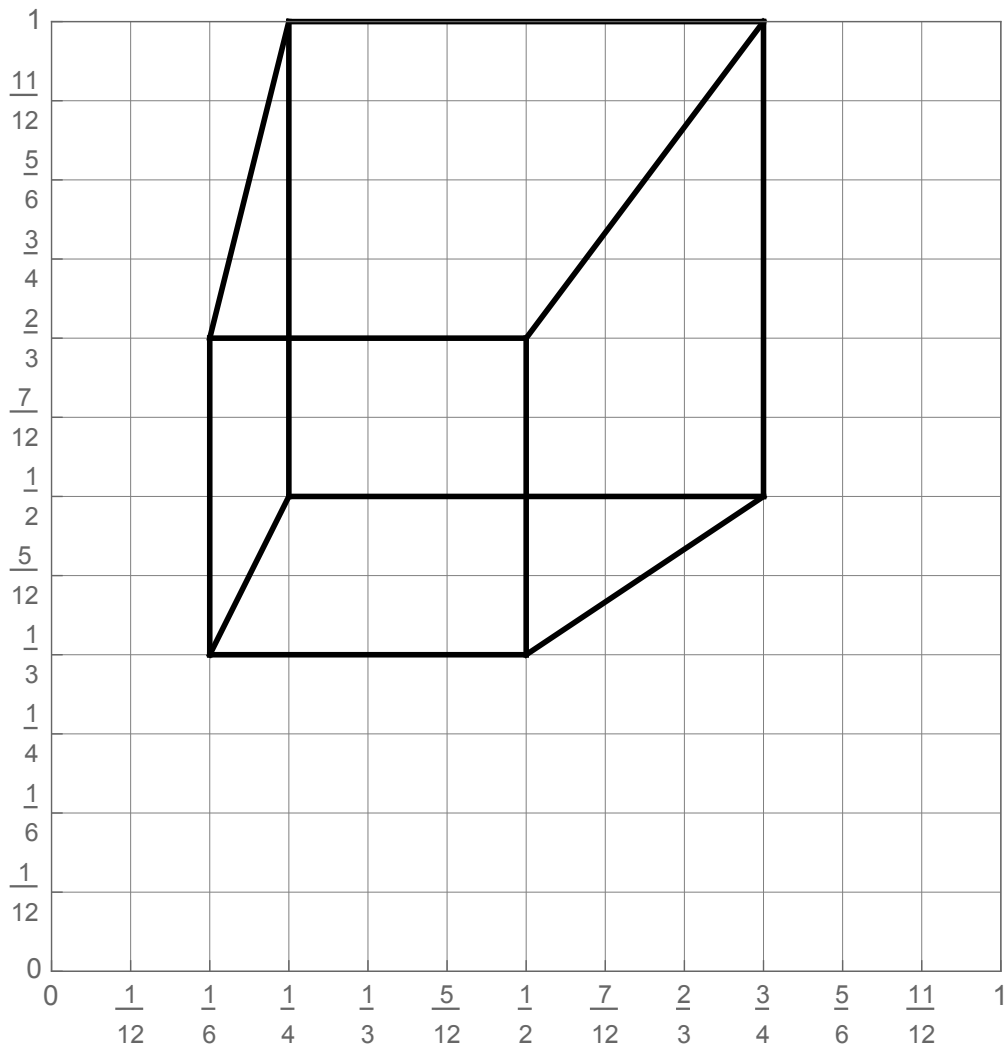
Dados do modelo

◎ Vértices

- A:(1,1,1) E:(1,1,-1)
- B:(-1,1,1) F:(-1,1,-1)
- C:(1,-1,1) G:(1,-1,-1)
- D: (-1,-1, 1) H: (-1,-1,-1)

◎ Segmentos da borda

- AB,CD,EF,GH,
- AC,BD,EG,FH,
- AE, CG, BF, DH



A: $\frac{1}{4}$, $\frac{1}{2}$

B: $\frac{3}{4}$, $\frac{1}{2}$

C: $\frac{1}{4}$, 1

D: $\frac{3}{4}$, 1

E: $\frac{1}{6}$, $\frac{1}{3}$

F: $\frac{1}{2}$, $\frac{1}{3}$

G: $\frac{1}{6}$, $\frac{2}{3}$

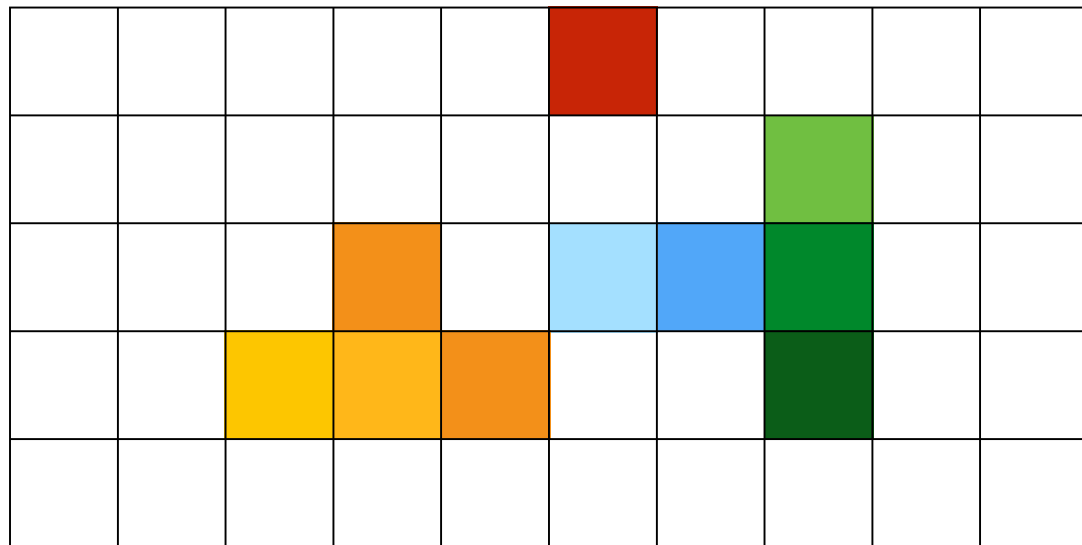
H: $\frac{1}{2}$, $\frac{2}{3}$

O que alcançamos?

- ◎ Muito bom! Vocês transformaram informações puramente digitais em informações puramente visuais, usando um procedimento completamente algorítmico.
- ◎ Como desenhamos linhas em um computador?

Saída para um display raster

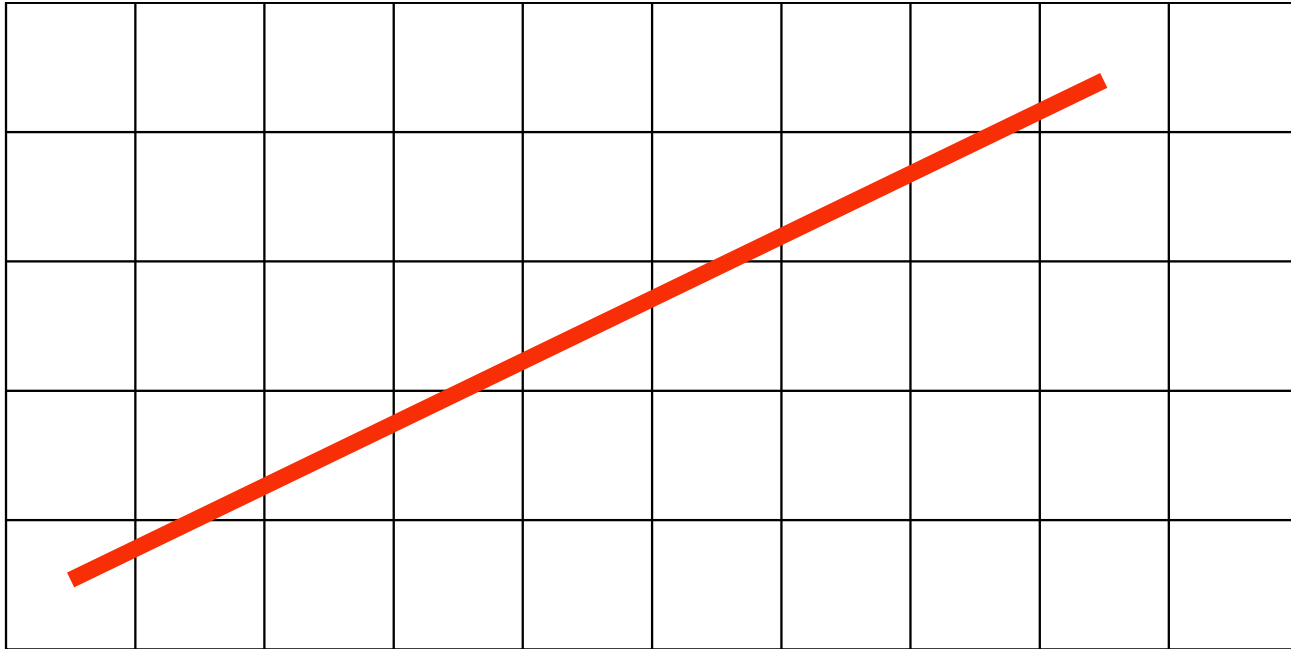
- ◎ Abstração comum de uma exibição raster:
- ◎ Imagem representada como uma grade 2D de “pixels” (elementos de imagem) **
- ◎ Cada pixel pode assumir um valor de cor único



** Vamos desafiar fortemente essa noção de um pixel “como um pequeno quadrado” em breve. Mas vamos nessa ideia por enquanto.

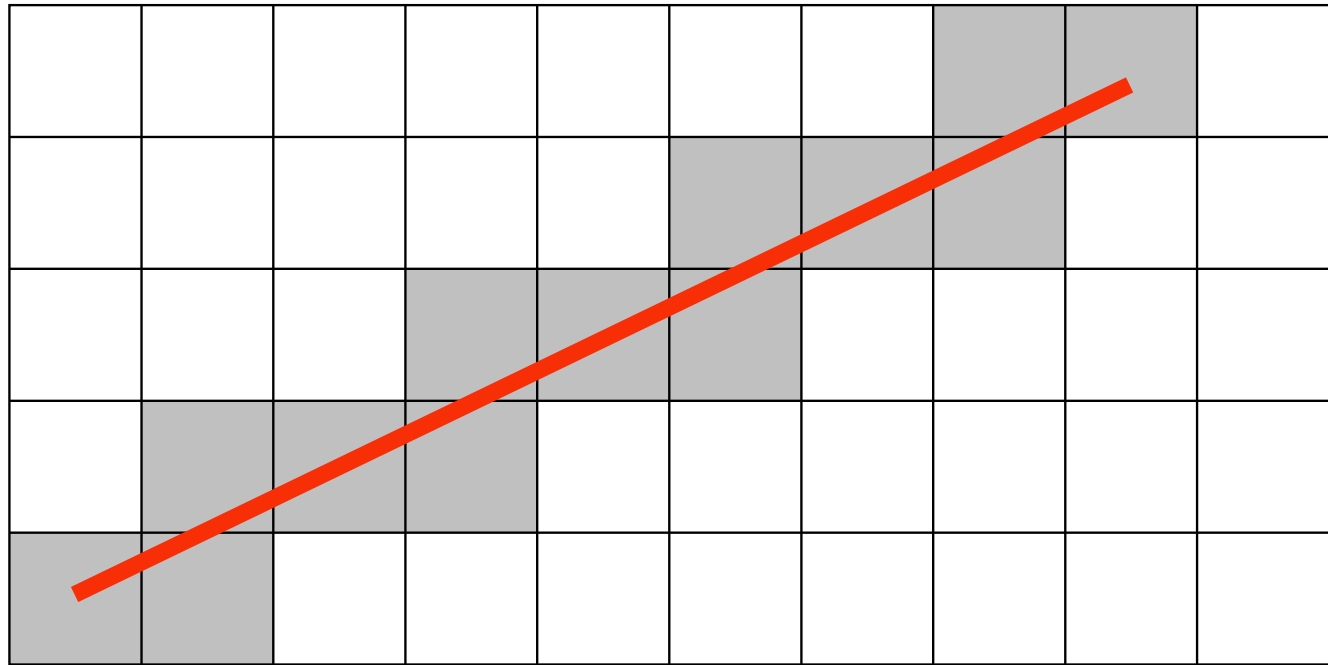
Quais pixels devemos colorir para representar uma linha?

- “Rasterização”: processo de conversão de um objeto contínuo em uma representação discreta em uma grade raster (grade de pixels)



Quais pixels devemos colorir para representar uma linha?

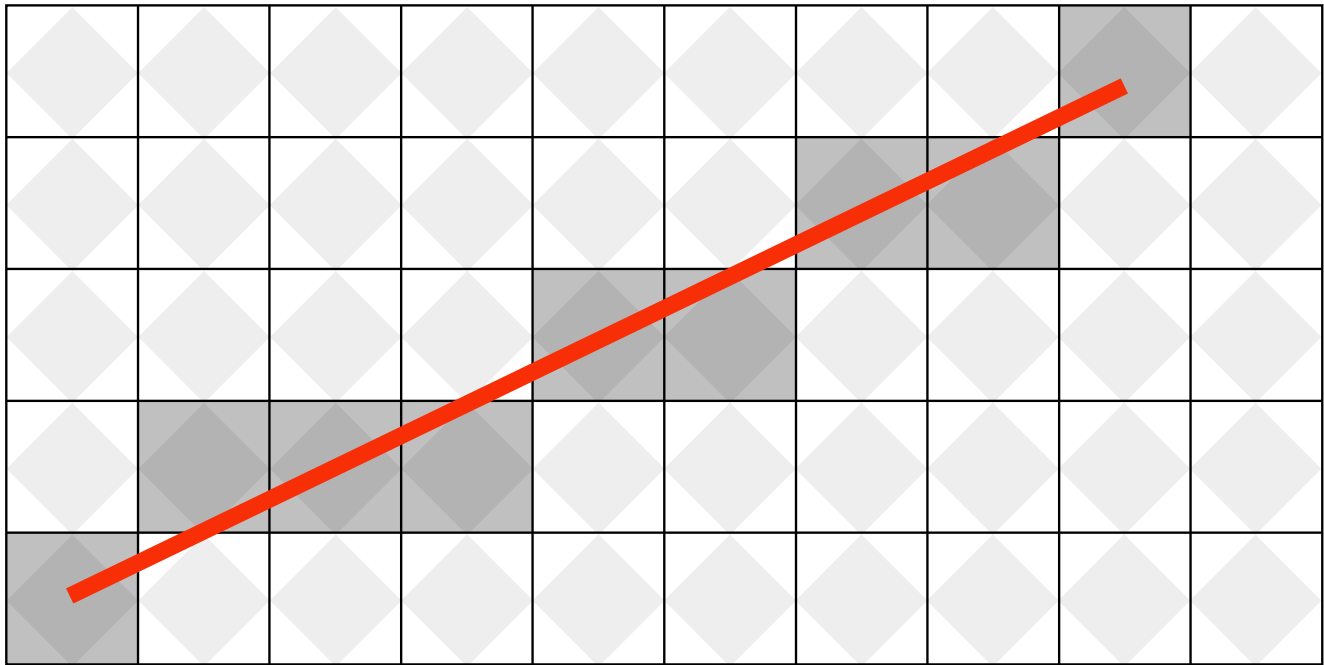
- © Destacar todos os pixels interceptados pela linha?



Quais pixels devemos colorir para representar uma linha?

- ◎ Regra de diamante/losango (usada por GPUs modernas): Destacar pixel se a linha passar pelo diamante associado

E se a linha tivesse uma espessura maior?



Como encontramos os pixels que satisfazem uma regra de rasterização escolhida?

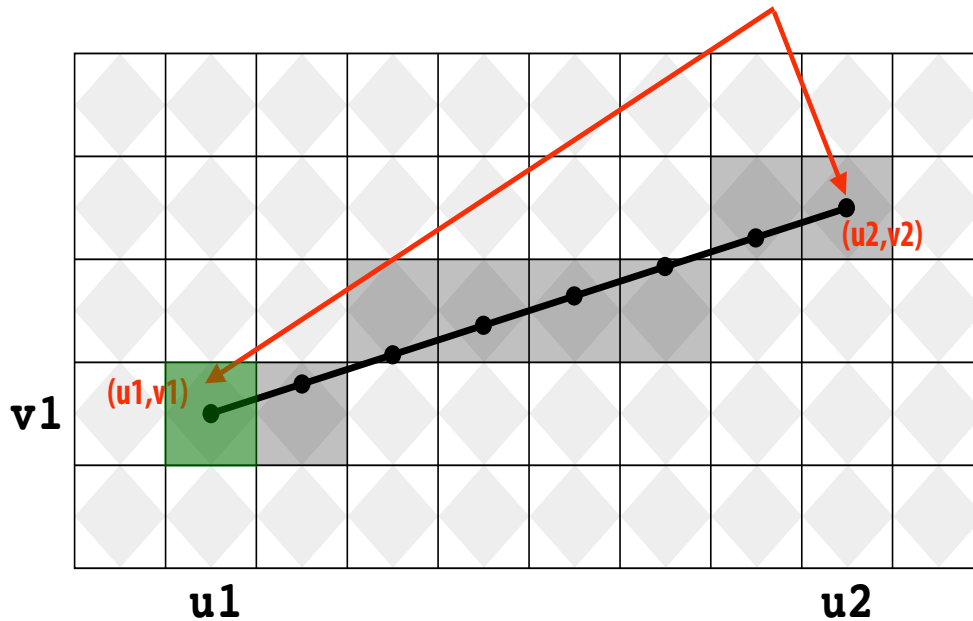
- ◎ Poderia verificar cada pixel na imagem para ver se atende à condição ...
 - $O(n^2)$ pixels na imagem vs. no máximo $O(n)$ pixels “destacados”
 - Deve ser capaz de fazer melhor! (por exemplo, trabalho proporcional ao número de pixels no desenho da linha)

Rasterização de linha incremental

- ◎ Considere que uma linha seja representada com pontos finais (endpoints) inteiros: (u_1, v_1) , (u_2, v_2)
- ◎ Inclinação da linha: $s = (v_2 - v_1) / (u_2 - u_1)$
- ◎ Considerando um caso especial simples:
 - $u_1 < u_2$, $v_1 < v_2$ (a linha aponta para o canto superior direito)
 - $0 < s < 1$ (mais mudança em x do que em y)

```
v = v1;  
for(u=u1; u<=u2; u++)  
{  
    v += s;  
    draw(u, round(v))  
}
```

Considerando que as coordenadas inteiras estejam nos centros dos pixels



Agora temos nosso primeiro algoritmo gráfico completo!

Digital information

VERTICES

A: (1, 1, 1)

B: (-1, 1, 1)

C: (1, -1, 1)

D: (-1, -1, 1)

E: (1, 1, -1)

F: (-1, 1, -1)

G: (1, -1, -1)

H: (-1, -1, -1)

EDGES

AB, CD, EF, GH,

AC, BD, EG, FH,

AE, CG, BF, DH

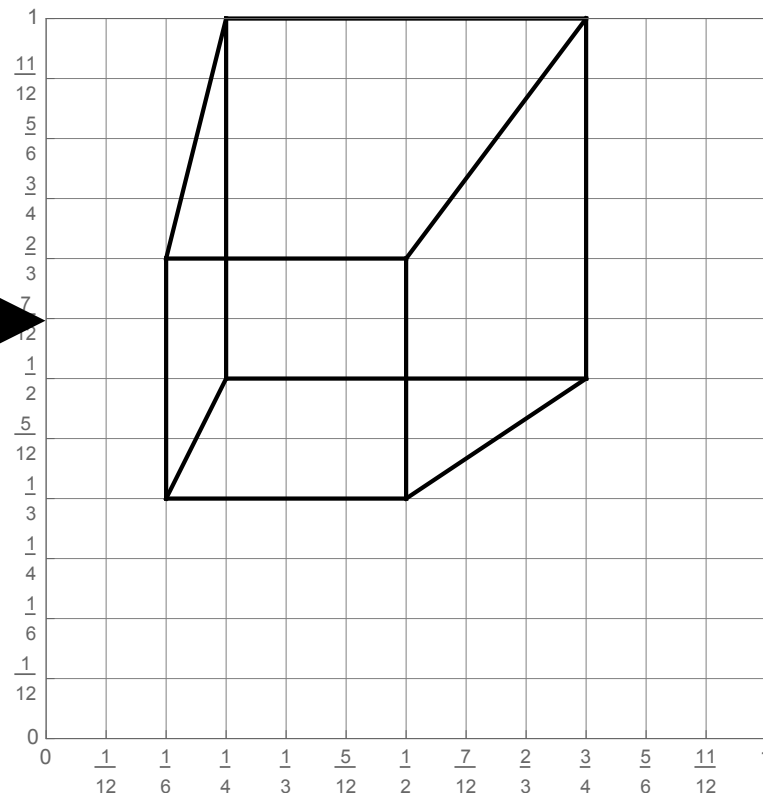
CAMERA

C = (2, 3, 5)

computation



Visual information





Até agora, apenas fiz um simples desenho de linha de um cubo.

- ◎ Para imagens mais realistas, será necessário um modelo muito mais rico do mundo:
 - GEOMETRIA
 - MATERIAIS
 - LUZES
 - CÂMERAS
 - MOVIMENTO ...
- ◎ Veremos esses e outros conteúdos ao longo do semestre.

Bibliografia

- © Jonas Gomes, Luiz Velho. Fundamentos da Computação Gráfica. IMPA, 2008.
- © Eduardo Azevedo, Aura Conci. Computação Gráfica - Geração de imagem - Volume 1. GEN LTC, 2003.
- © Aura Conci, Eduardo Azevedo, Fabiana R. Leta. Computação Gráfica - Vol. 2 - Teoria e Prática: Volume 2. GEN LTC, 2007.