

Computação Gráfica

Imagens



Imagem

Percepção visual

Cor

Formação de imagens

Observadores (câmeras)

Representação

Tipos de imagens



Etch-a-Sketch copy of "A Sunday on La Grande Jatte." Photo by Jane Labowitch via Google Images.

discussão:

o que é uma imagem?

x^2 - 2.3

Função:

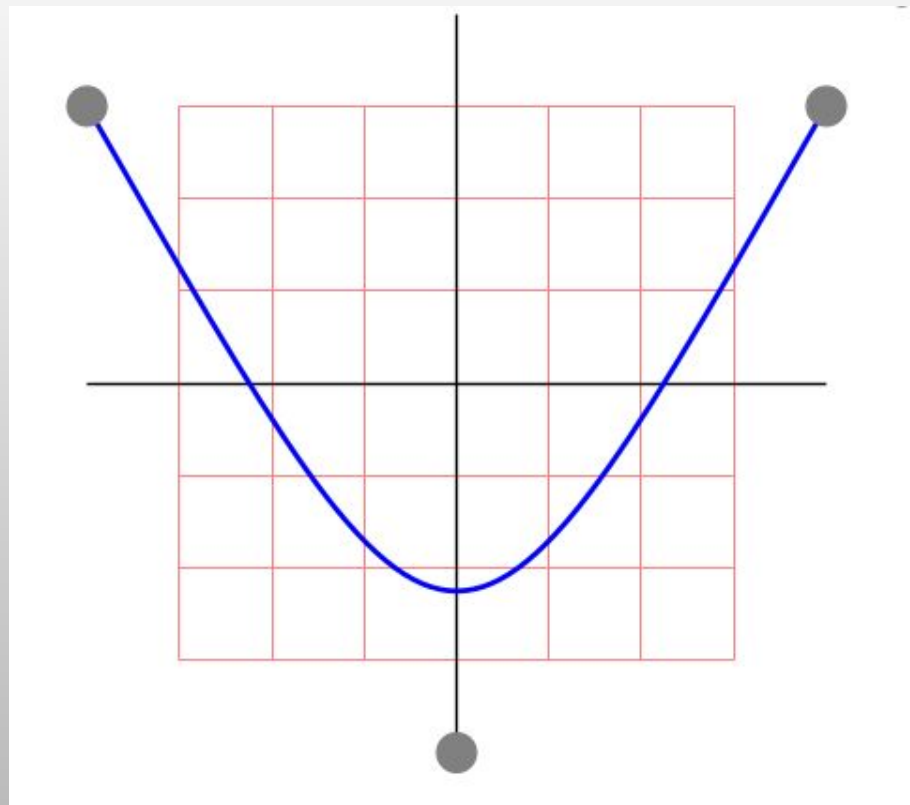
- relação entre 2 conjuntos
- $f : A \rightarrow B$

Domínio: conjunto A

Contradomínio: conjunto B

Conjunto Imagem:

- subconjunto de B, formado pela imagem de cada um dos elementos de A.



Formação

Geração

Visão humana:

percepção de
"coisas"

mas também
coisas
abstratas



Aquisição

Representação

Transmissão

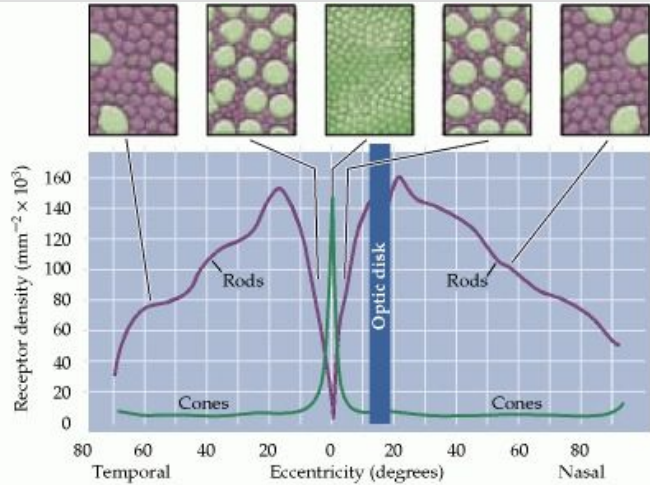
Exibição

Visão Humana

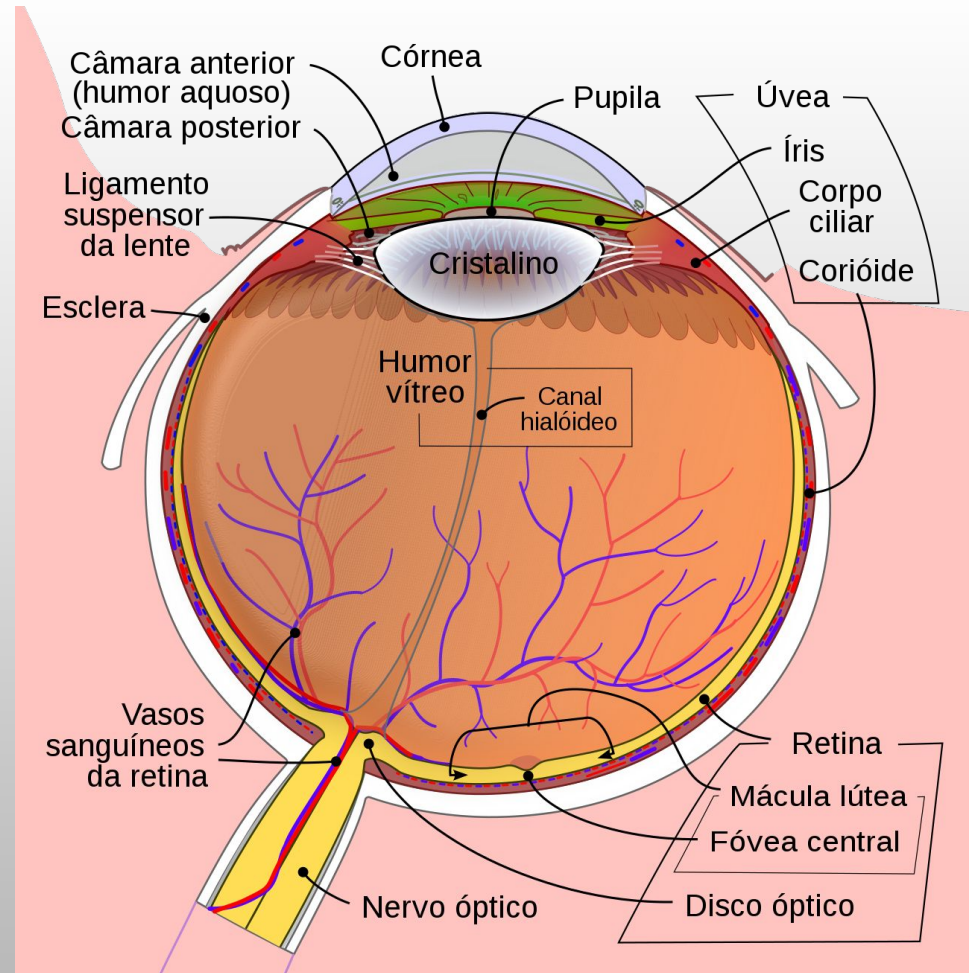
Olho humano

Luz penetra o olho pela abertura na íris chamada pupila.

A luz é focalizada pelo cristalino sobre a fóvea, a região da retina que concentra os cones, células responsáveis pela percepção de "cor".

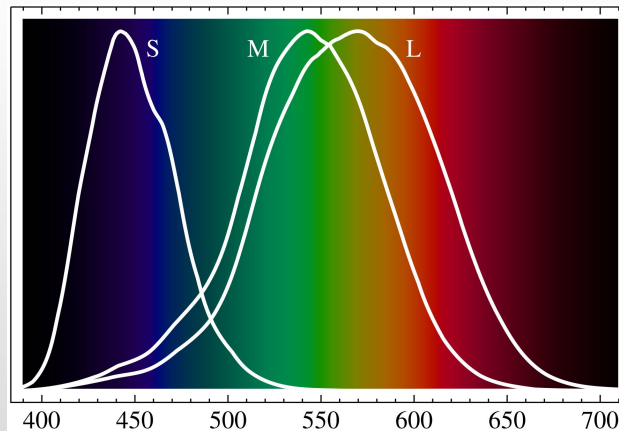


[https://www.ncbi.nlm.nih.gov/books/NBK10848/#:~:text=Distribution%20of%20ods%20and%20cones%20in%20the%20human%20retina.,%2C%20\(more...\)](https://www.ncbi.nlm.nih.gov/books/NBK10848/#:~:text=Distribution%20of%20ods%20and%20cones%20in%20the%20human%20retina.,%2C%20(more...))



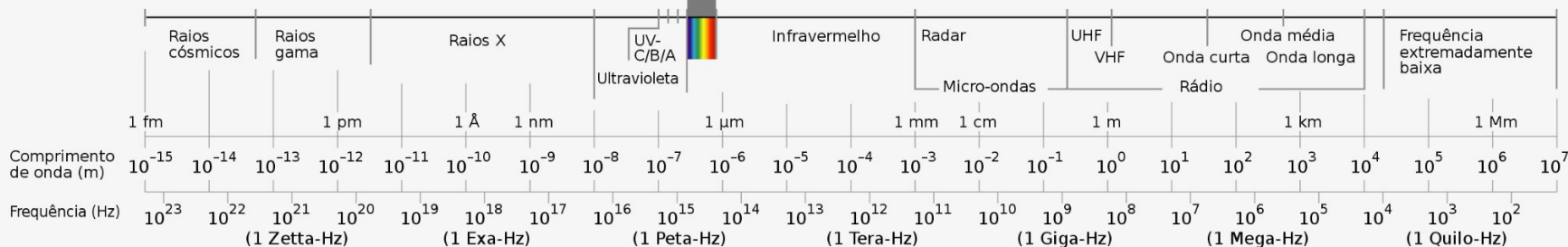
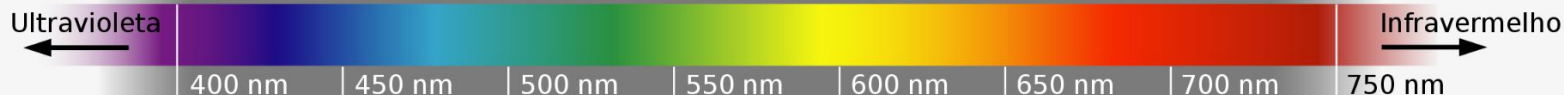
<https://pt.wikipedia.org/wiki/Olho>

Percepção de cor



https://pt.wikipedia.org/wiki/Espectro_vis%C3%ADvel#/media/Ficheiro:Electromagnetic_spectrum_-_pt.svg

Espectro visível pelo olho humano (Luz)



monitores ou displays

Display: tubo de raio catódicos

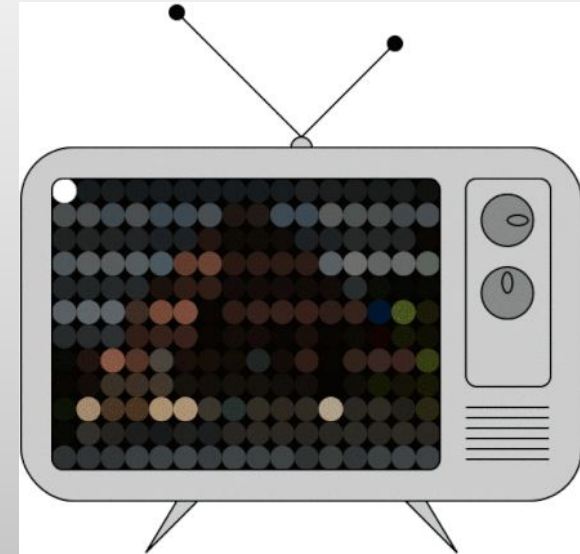
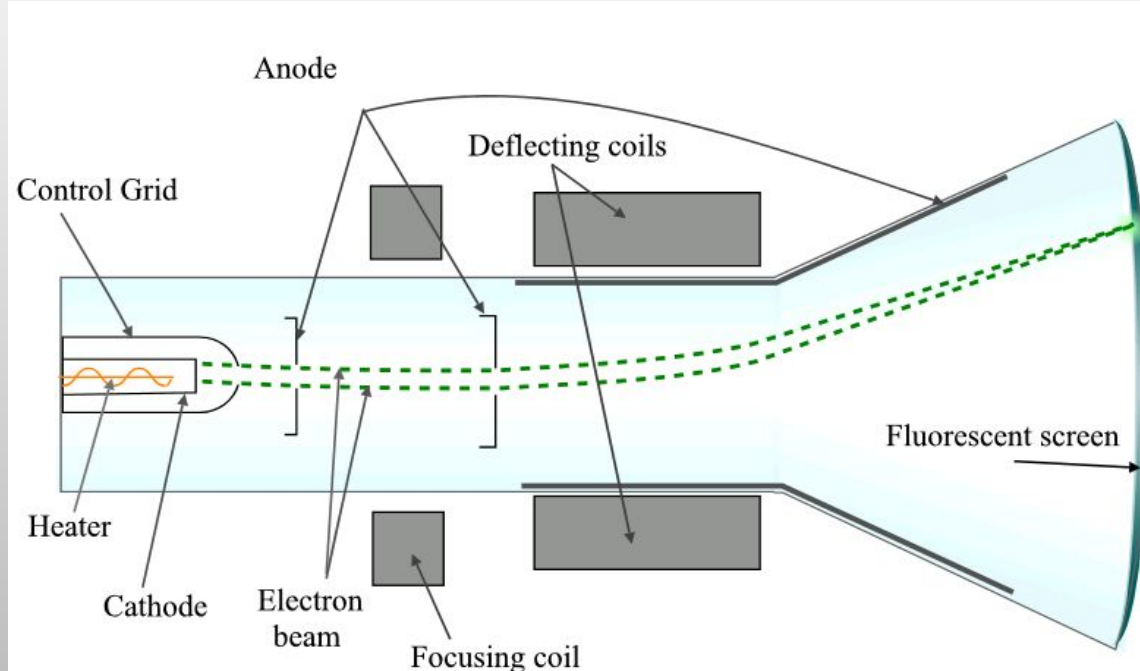
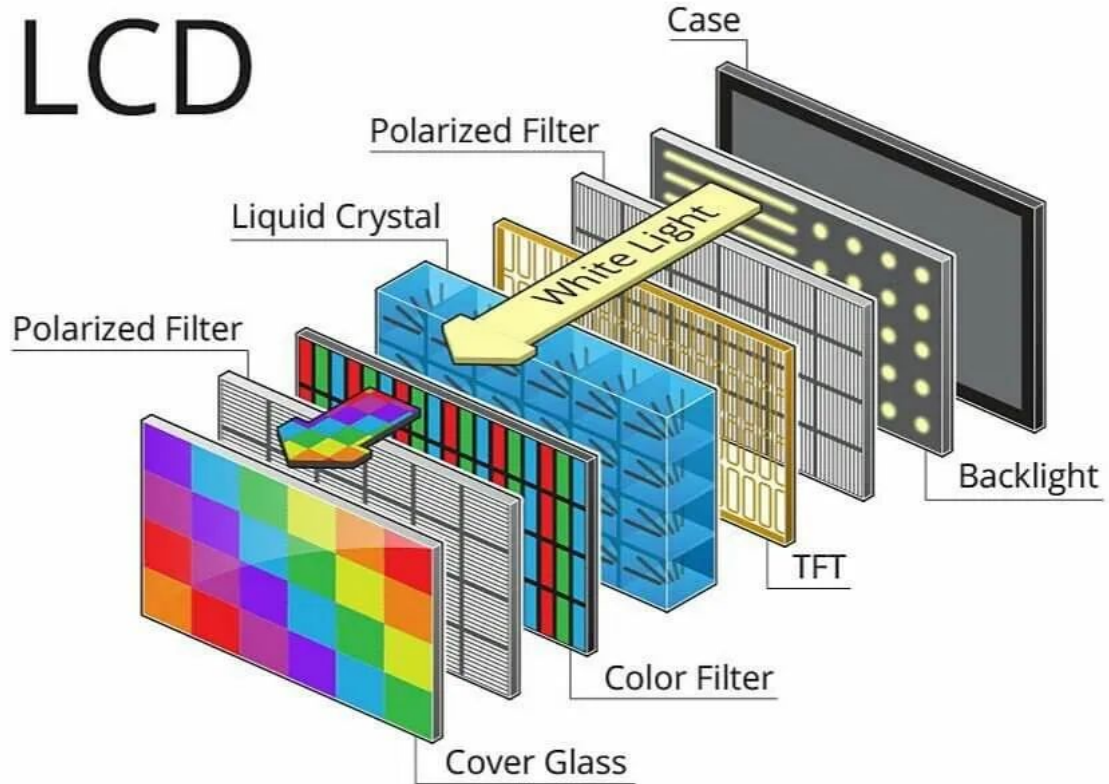


Imagem
Analógica

Display LCD



Imagens em monitores

Imagem raster

Imagem é representada como um array de pixels

Resolução: $W \times H$

Aspect ratio: $a = W / H$

Profundidade: número de pixels para representar a "cor" de um pixel

- 1 bit => imagem binária (Binary image)
- 8 bits => imagem em nível de cinza (Gray level image)
- 24 bits => imagem colorida RGB
- 32 bits => imagem colorida RGBA

Exemplos

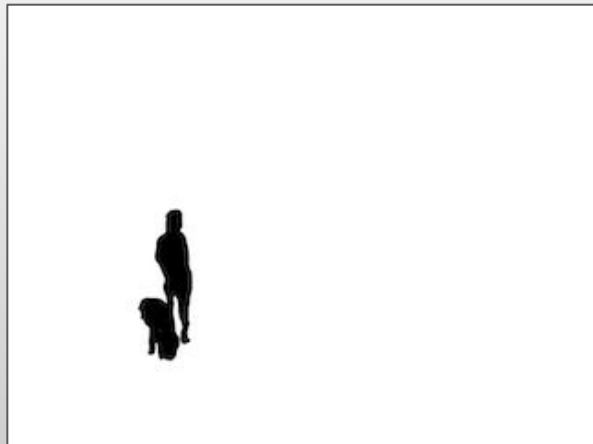


Imagem
Binária
1 bit



Imagem
Cinza
8 bits



Imagem
Colorida
24 bits

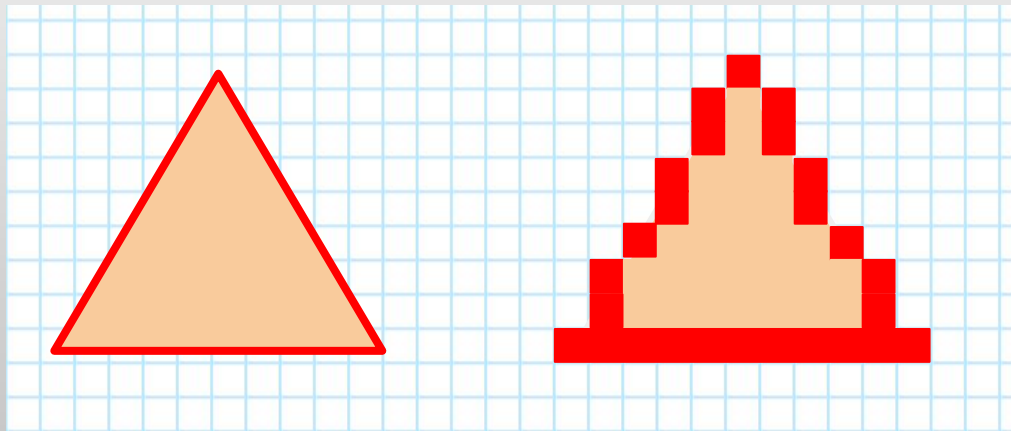
Imagens raster x vetorial

Imagem vetorial

- é formada por segmentos (vetores).
- representação compacta,
- fácil de escalar para qualquer resolução.

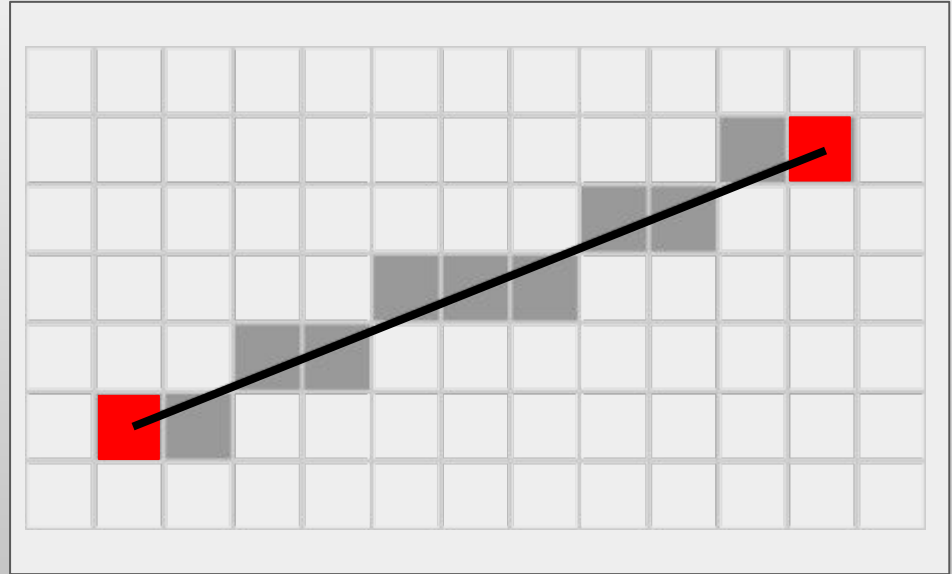
Imagem raster

- é formada por pixels.
- representação genérica pois cada pixel é independente
- uma imagem de grande resolução requer grande espaço de memória



Rasterização

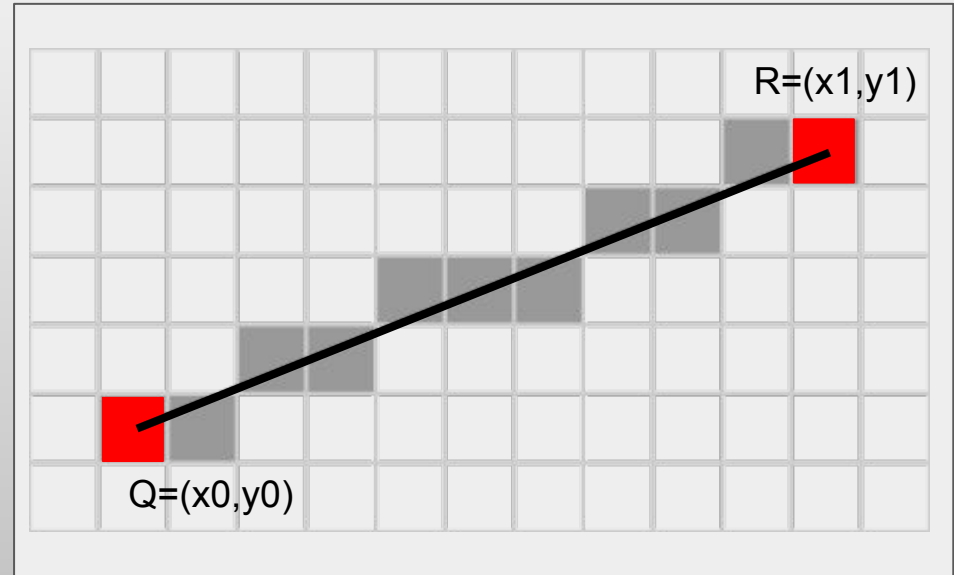
é o processo de conversão de segmentos de linhas e outros elementos gráficos como círculos, elipses, polígonos etc., para uma imagem raster.



Rasterização

EXERCÍCIO EM PARES

Dado um segmento definido por dois pontos $Q=(x_0, y_0)$ e $R=(x_1, y_1)$, escreva um algoritmo (em pseudo linguagem) que pinte todos os pixels do segmento.



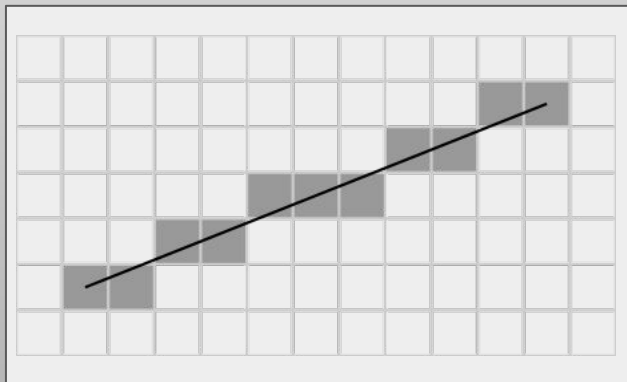
Opcional: considere que o coeficiente angular do segmento está no intervalo $[0, 45^\circ]$

Algoritmo de Bresenham para linhas

https://en.wikipedia.org/wiki/Bresenham%27s_line_algorithm

Ideias iniciais:

- Como desenhar uma linha horizontal?
 - inclinação = 0
- Como desenhar uma linha diagonal?
 - inclinação = 45°
- Como desenhar uma linha com inclinação entre 0 e 45°?



Equação de uma linha:

$$y = m x + b$$

onde $m = dy / dx = (y1 - y0) / (x1 - x0)$

Outra forma:

$$A x + B y + C = 0$$

onde $A = dy$, $B = -dx$, $C = b * dx$

Interpretação: o valor de um ponto sobre a linha é zero. Mas qual o valor dessa equação para

- pontos acima da linha?
- pontos abaixo da linha?

Como decidir entre y_0 e y_0+1 ?

Seja $f = Ax + By + C$

Considere $M = f(x_0+1, y_0+\frac{1}{2})$

- Quando $M > 0$, a linha passa acima de M e assim devemos pintar (x_0+1, y_0+1) .
- Caso contrário, pintamos (x_0+1, y_0)

Ao invés de M , o algoritmo usa a diferença

$$D = f(x_0+1, y_0+\frac{1}{2}) - f(x_0, y_0)$$

$$= A + B/2 = dy - dx/2$$

Isso permite o uso de aritmética inteira.

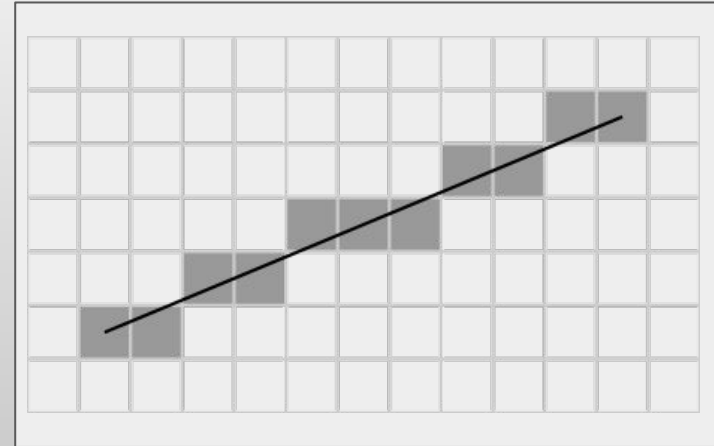
Da mesma forma, se $D > 0$,

- pintamos o pixel (x_0+1, y_0+1)
 - e nesse caso o próximo D pode ser calculado como $D = D + dy - dx$
 - verifique!
- caso contrário: pintamos (x_0+1, y_0)
 - e nesse caso o próximo D pode ser calculado como $D = D + dy$

Algoritmo de Bresenham para linhas

https://en.wikipedia.org/wiki/Bresenham%27s_line_algorithm

```
function Bresenham(x0, y0, x1, y1) {  
    dx = x1 - x0;  
    dy = y1 - y0;  
    D = 2*dy - dx;  
    y = y0;  
  
    for (let x = x0; x<=x1; x++) {  
        pinte(x, y);  
        if (D > 0) {  
            y = y + 1;  
            D = D - 2*dx;  
        }  
        D = D + 2*dy;  
    }  
}
```



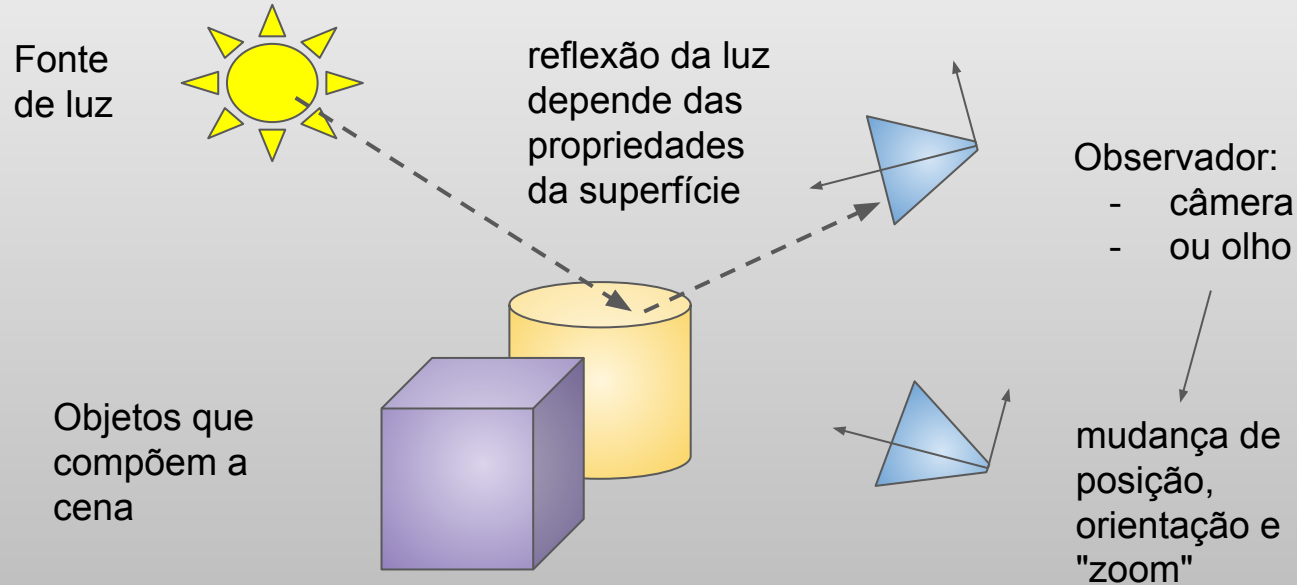
Simule o algoritmo com os valores da figura.

$Q = (0, 0)$

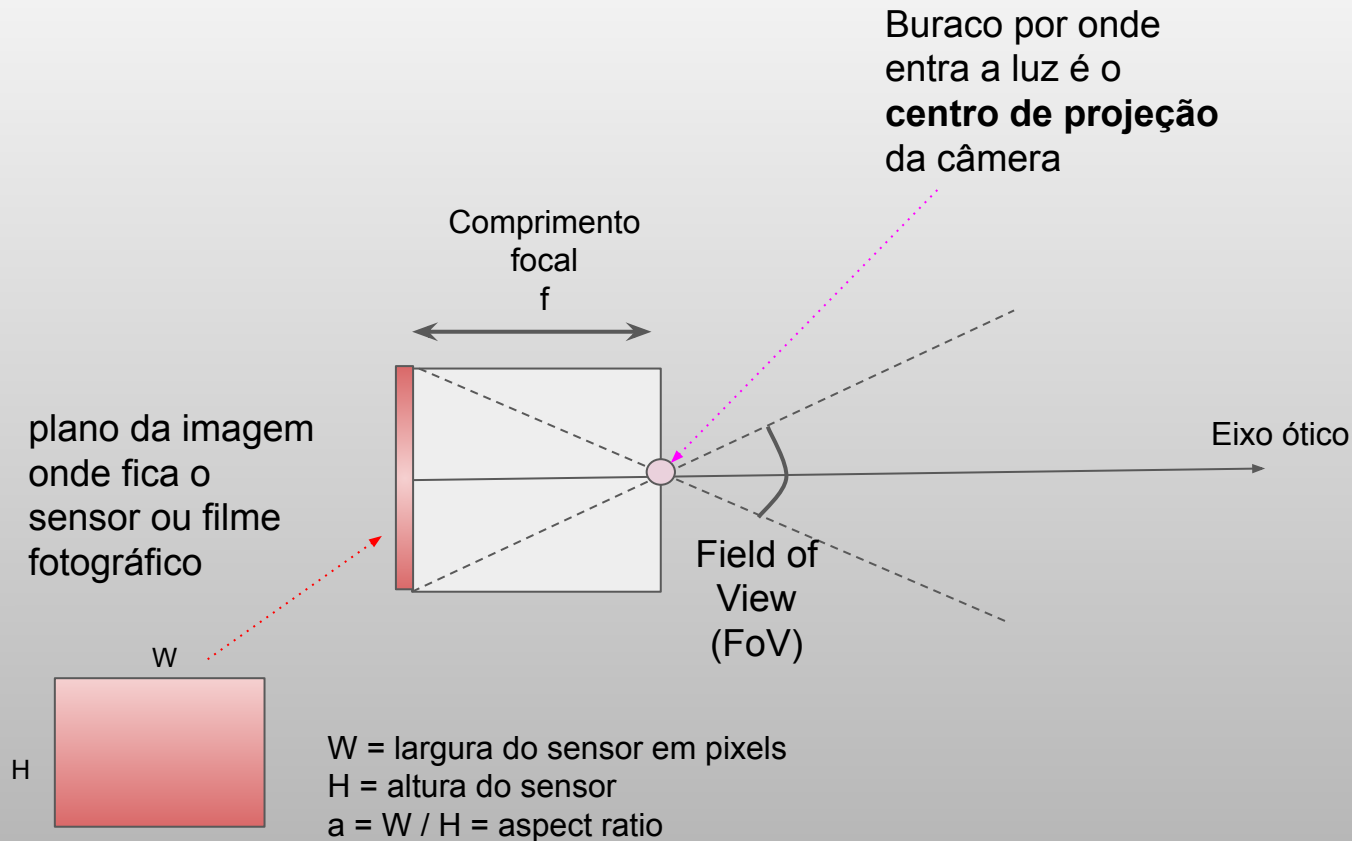
$R = (10, 4)$

Formação de imagens

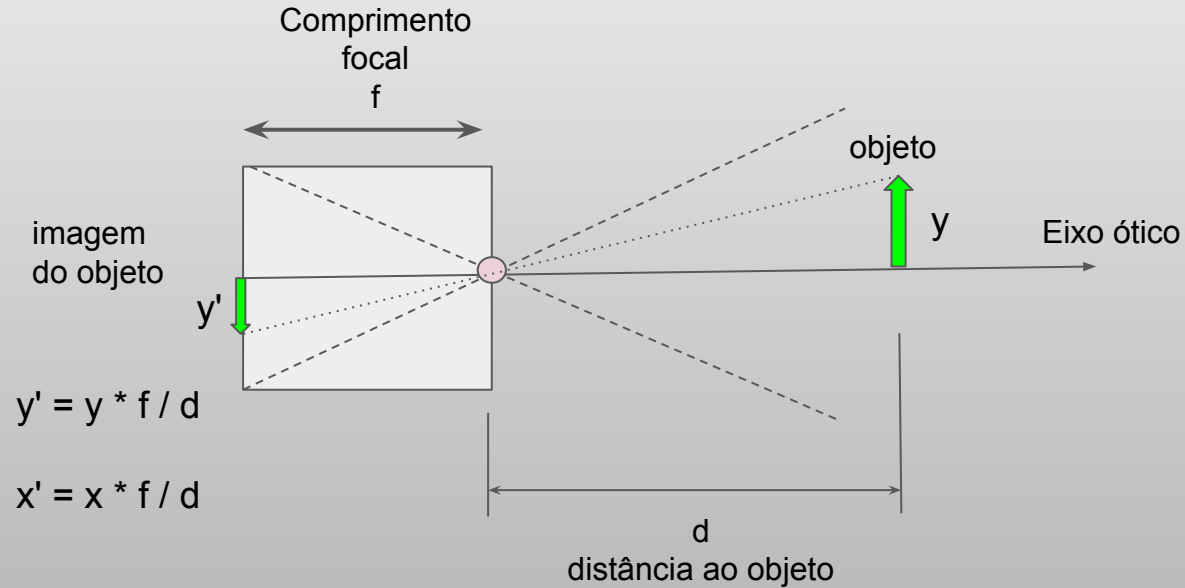
Formação de imagens (luz que chega no sensor)



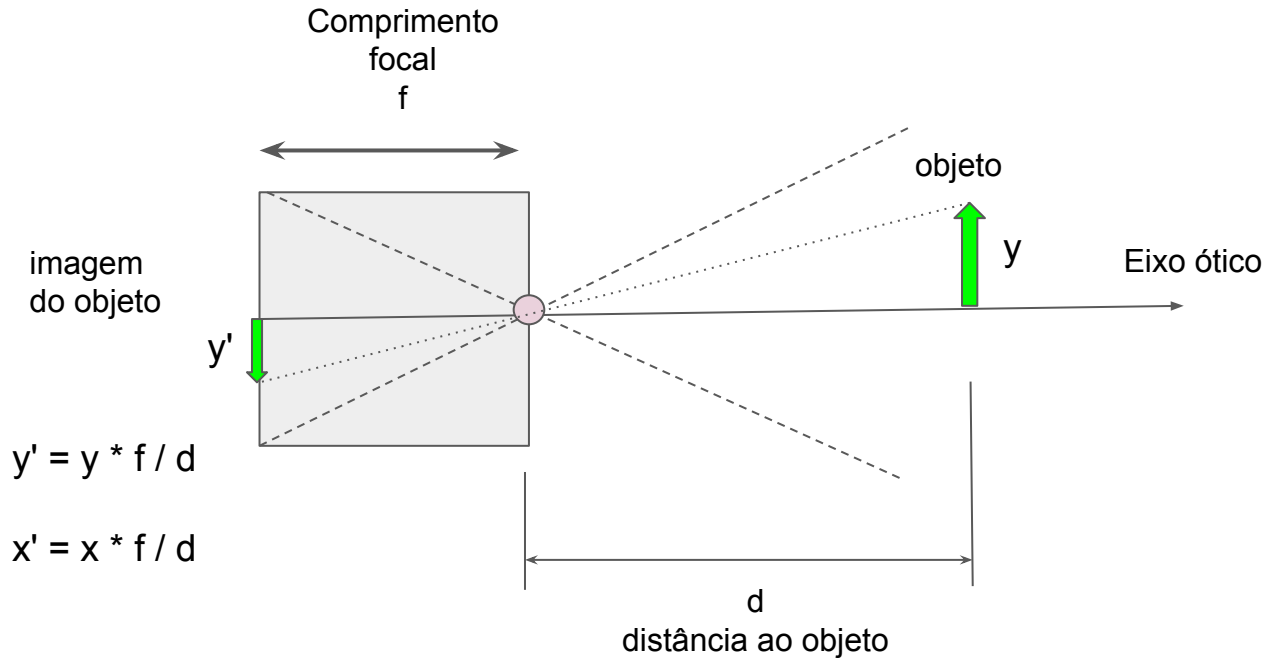
Câmara Pinhole



Câmera Pinhole e a imagem de um objeto



Câmera Pinhole e a imagem de um objeto



Até onde chegamos e para onde vamos?

Vimos nessa aula:

- Um pouco sobre a estrutura do olho humano e a percepção de cores
- Elementos que participam do processo de formação de uma imagem física
- Câmera pinhole
- Tipos de imagem: raster e vetorial
- Algoritmo de Bresenham para rasterização de segmentos de linha

Agora temos uma ideia de como uma imagem é formada e como pode ser representada na forma raster, ou seja, como um array de pixels.

Renderização é o processo de geração de uma imagem a partir de um modelo da cena. Parte desse processo pode envolver a **rasterização** de linhas e outros objetos geométricos primitivos.

Na próxima aula vamos descrever a arquitetura de **sistemas gráficos** que permitem renderizar cenas de forma computacionalmente eficiente.

O que você deve saber e fazer

- O que é uma imagem colorida (RGBA), imagem em nível de cinza e imagem binária?
- Quais as principais tecnologias usadas para exibir imagens
 - como funcionam os tipos distintos de monitores?
 - quais as vantagens e desvantagens de cada tipo?
 - O que são monitores analógicos, vetoriais e raster?
 - O que são imagens analógicas, vetoriais e raster? Como as representamos?
- Como uma imagem é formada no sensor de uma câmera pinhole?
- O que é rasterização?
- O que é renderização?