

VISÃO COMPUTACIONAL

GRUPO DE ESTUDOS

31-10-24 | CARLOS ROCHA E RAFAEL BATISTA

Apoio



CENTRO DE COMPETÊNCIA EMBRAPII
EM TECNOLOGIAS IMERSIVAS

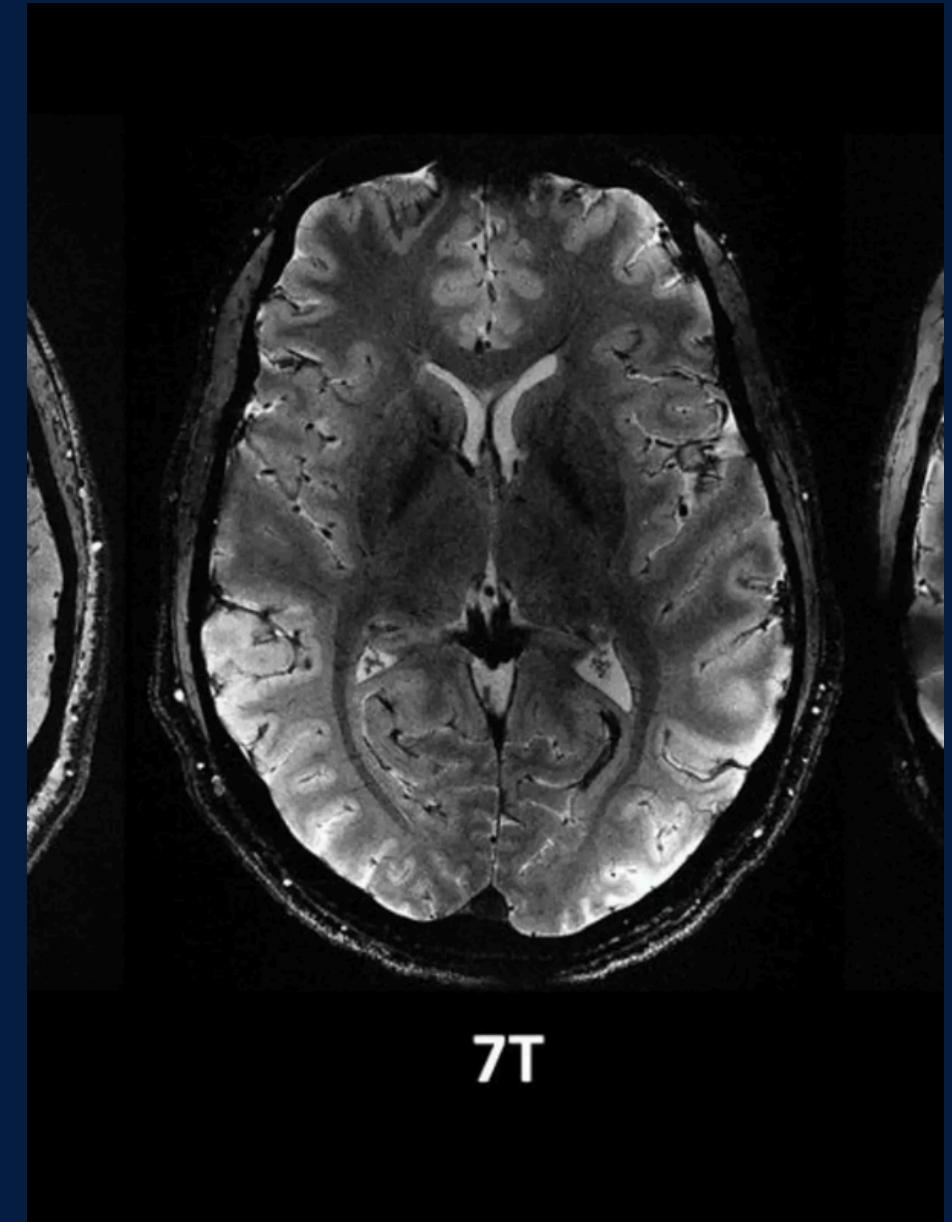


O que é Visão Computacional?

É uma área da inteligência artificial que busca desenvolver algoritmos e técnicas para que computadores possam interpretar e extrair informações significativas de **imagens** e **vídeos**, simulando a capacidade de visão humana.

O que é uma imagem?

Uma imagem é uma representação visual de uma ideia, objeto ou fenômeno, que pode ser captada por meios físicos ou criada artificialmente.



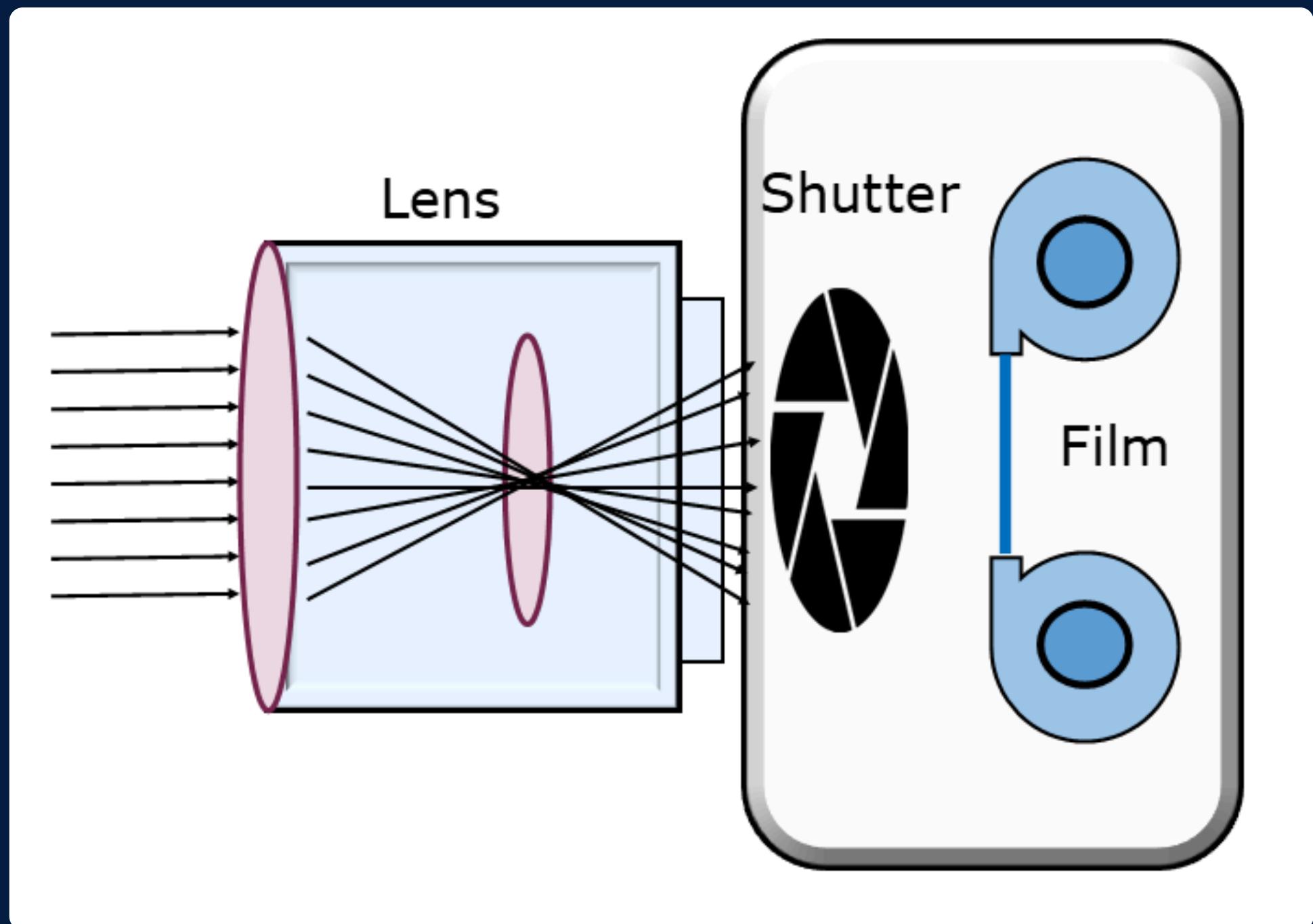
- Idade Medieval
- Projeção Pinhole
- Desenho sobre tela
- Século XIX, placa de metal de prata em câmara portátil
- Filme em solução de revelação

História da Fotografia

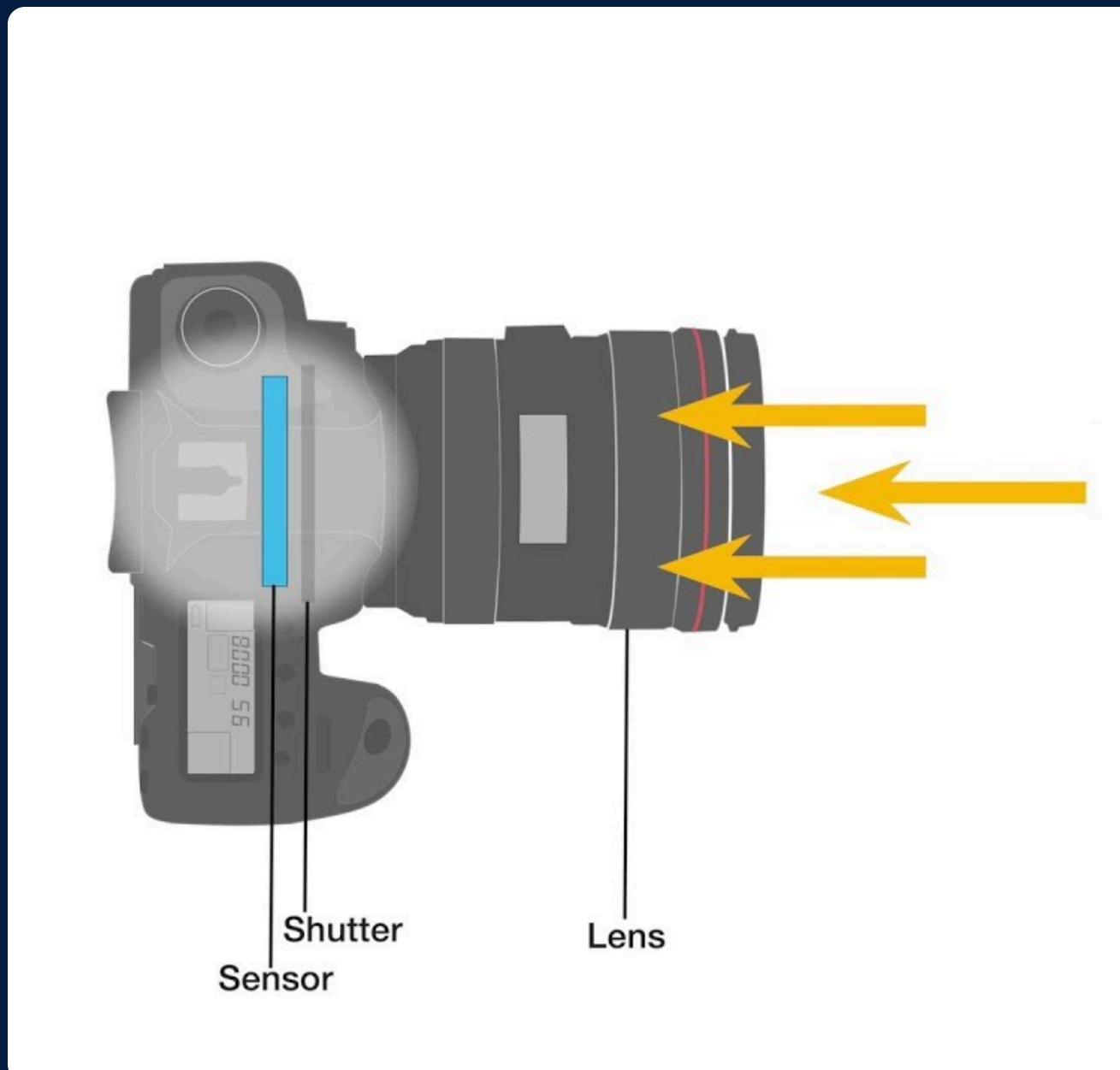


Como funciona a captura de uma imagem analógica?

1. Abertura do obturador
2. Passagem de luz pela lente
3. Exposição do filme



E em uma câmera digital?



- **Sensores CMOS (Complementary Metal-Oxide-Semiconductor)**
- **Captam luz e a transformam em sinais elétricos, que são convertidos em dados digitais para formar uma imagem.**



E em uma câmera digital?

Sensor Functions:

1. Photoelectric Conversion

Converts photons into electrons

2. Charge Accumulation

Collects generated charge as signal charge

3. Transfer Signal

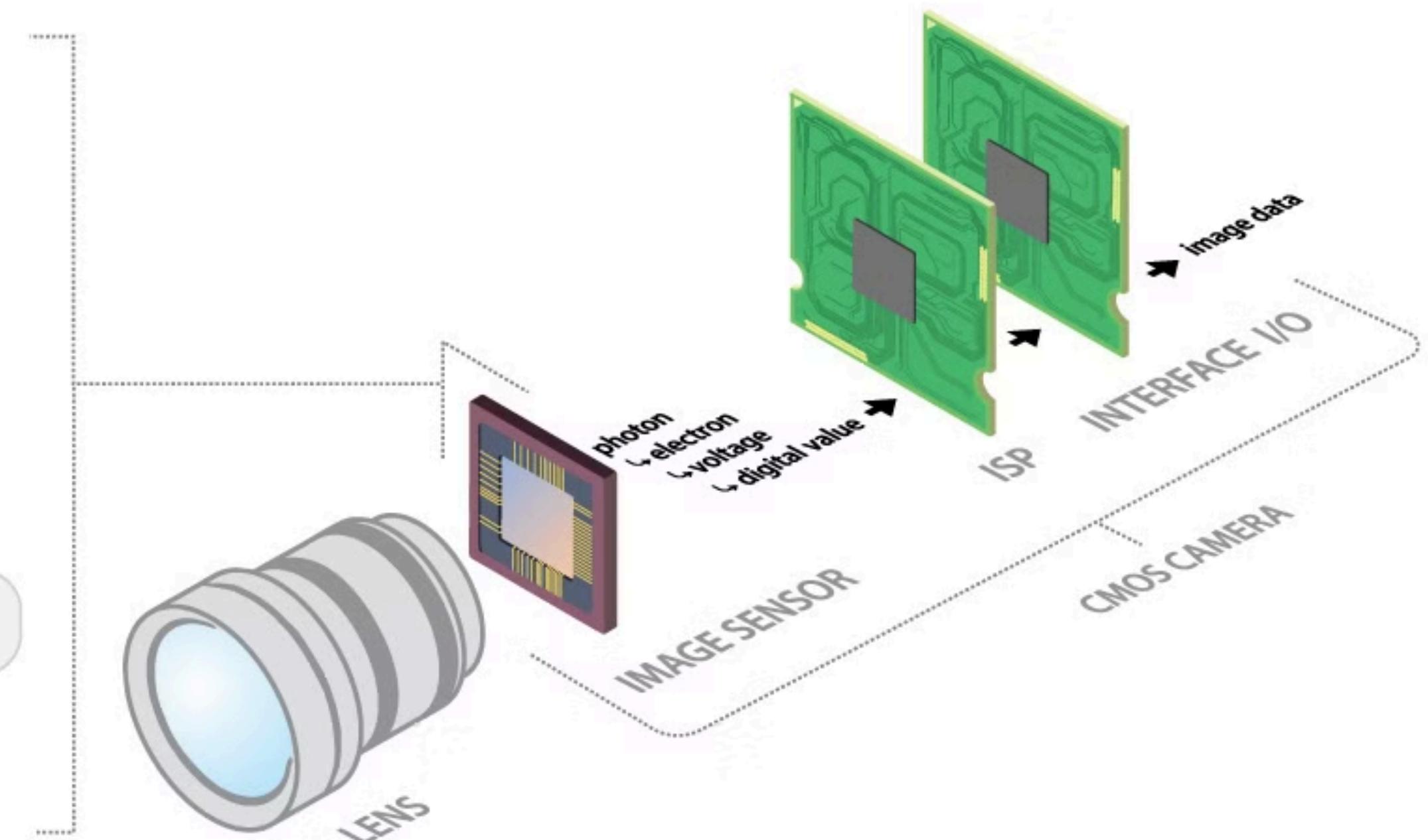
Moves signal charge to detecting node

4. Signal Detection

Converts signal charge into electrical signal (voltage)

5. Analog to Digital Conversion

Converts voltage into digital value



CAMERA SENSOR SIZE COMPARISON CHART

	MEDIUM FORMAT	FULL-FRAME	APS-C	MICRO 4/3	1"	1/2.55"
PICTURE						
SENSOR SIZE	53.0 X 40.20 MM	35.00 X 24.00 MM	23.6 X 15.60 MM	17.00 X 13.00 MM	12.80 X 9.60 MM	6.17 X 4.55 MM
CROP FACTOR	0.64	1	1.52	2	2.7	5.62
CAMERA						

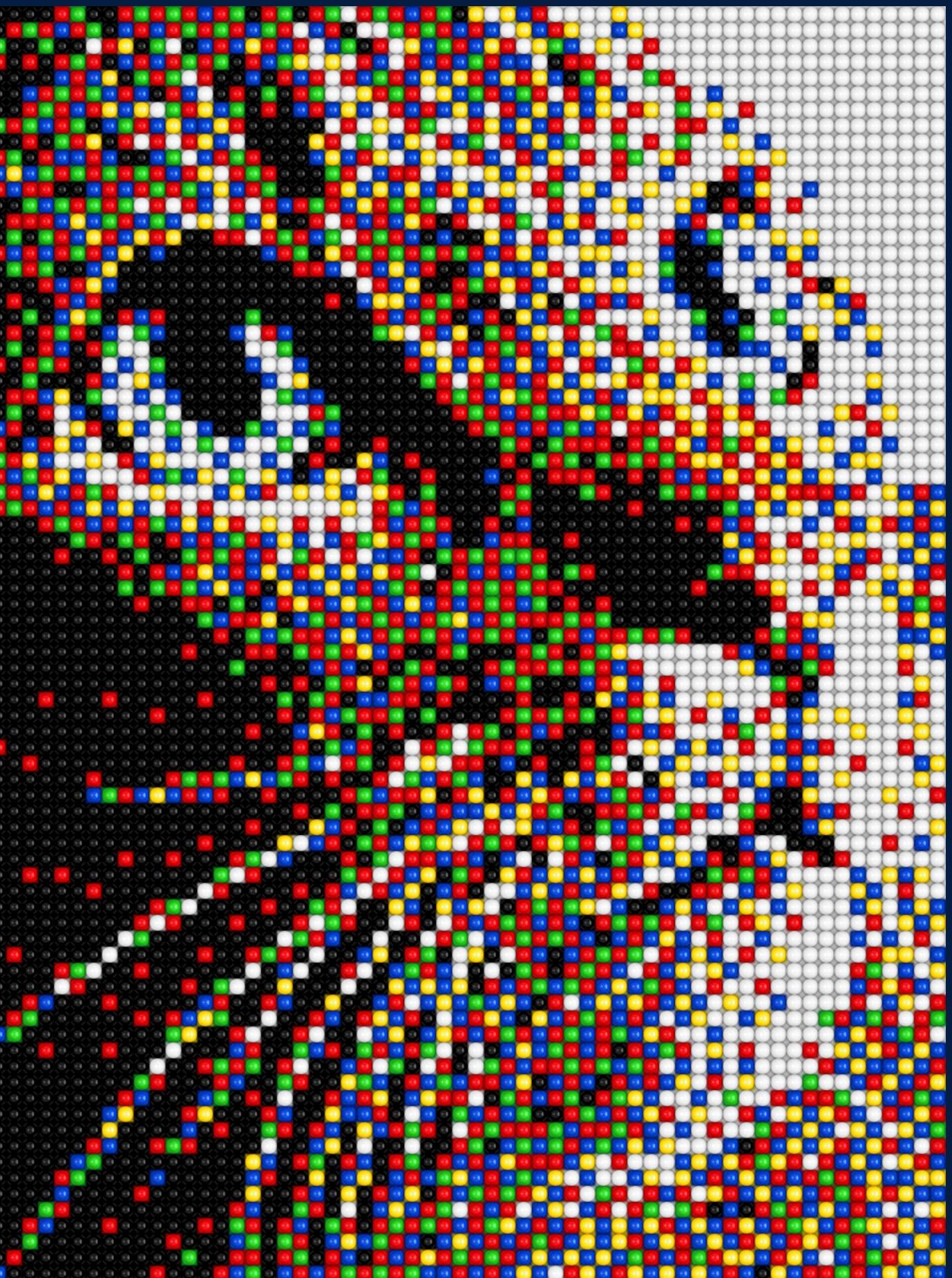
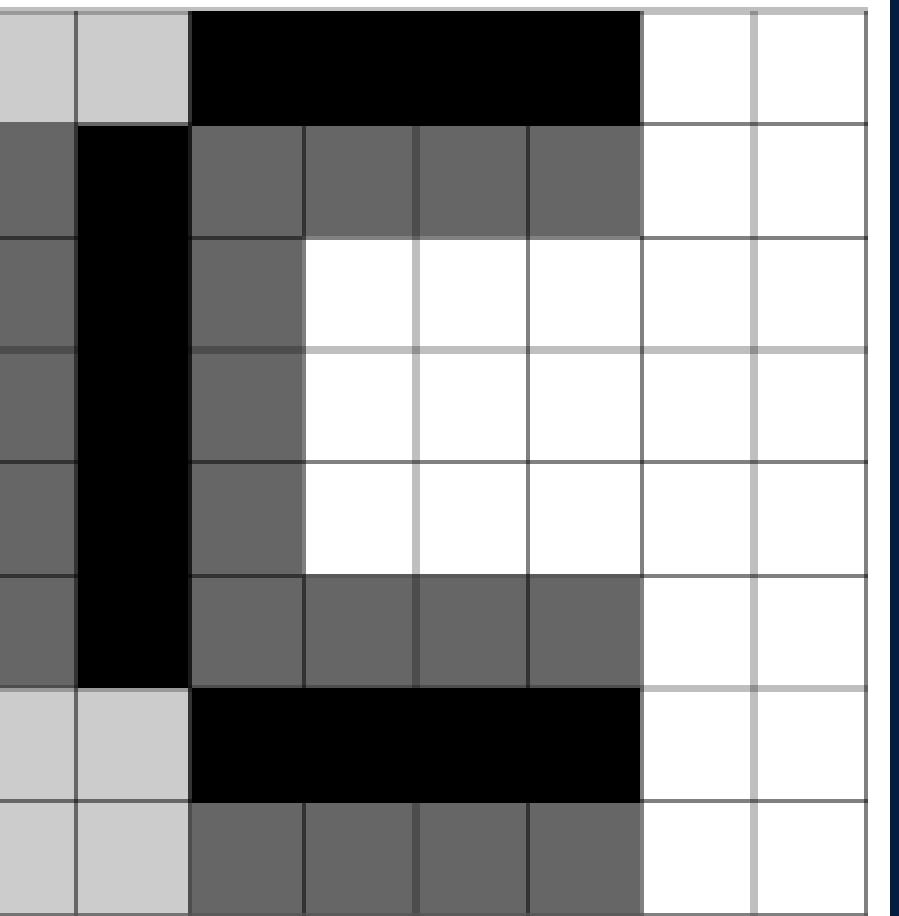
APS-C vs Full-Frame



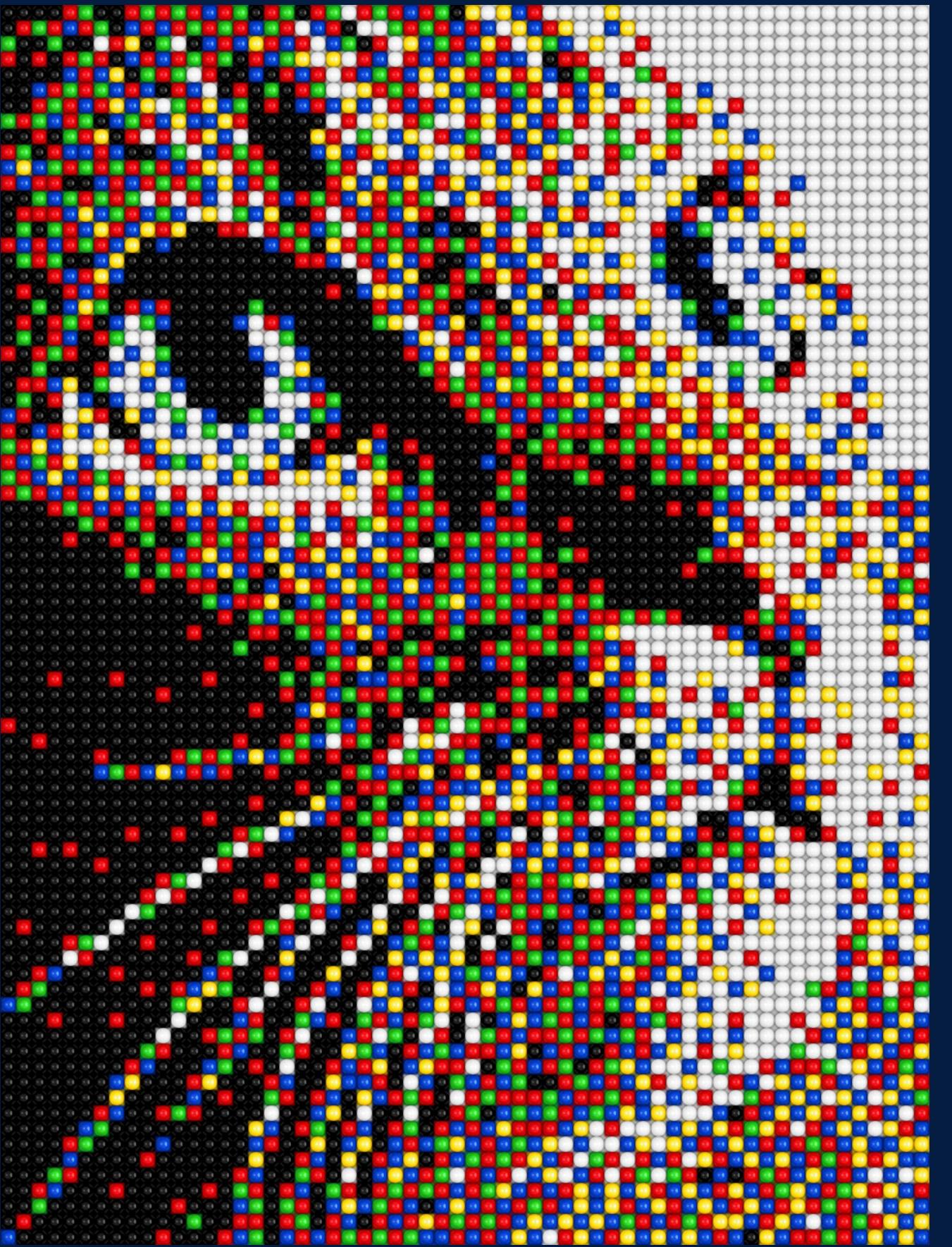
Por que “imagem digital”?

Os computadores representam uma imagem como uma matriz de pixels.

100	100	0	0	0	0	255	255
50	0	50	50	50	50	255	255
50	0	50	255	255	255	255	255
50	0	50	255	255	255	255	255
50	0	50	255	255	255	255	255
50	0	50	50	50	50	255	255
100	100	0	0	0	0	255	255
100	100	50	50	50	50	255	255



pixel = picture elements



Metadados

Informações adicionais que descrevem aspectos da própria imagem, como suas características, origem, configuração e contexto de uso.



Formatos mais comuns:

- EXIF (**dados da captura**)
- IPTC (**copyright**)
- XMP (**personalizado**)

Metadados



```
from PIL import Image
from PIL.ExifTags import TAGS

# Carregue a imagem
image_path = 'pasta/imagem.jpg'
image = Image.open(image_path)

# Extraír os metadados EXIF
exif_data = image._getexif()

# Converter os dados de EXIF em formato legível (se possível)
exif_info = {}
if exif_data:
    for tag, value in exif_data.items():
        tag_name = TAGS.get(tag, tag)
        exif_info[tag_name] = value

print(exif_info)
```

Metadados

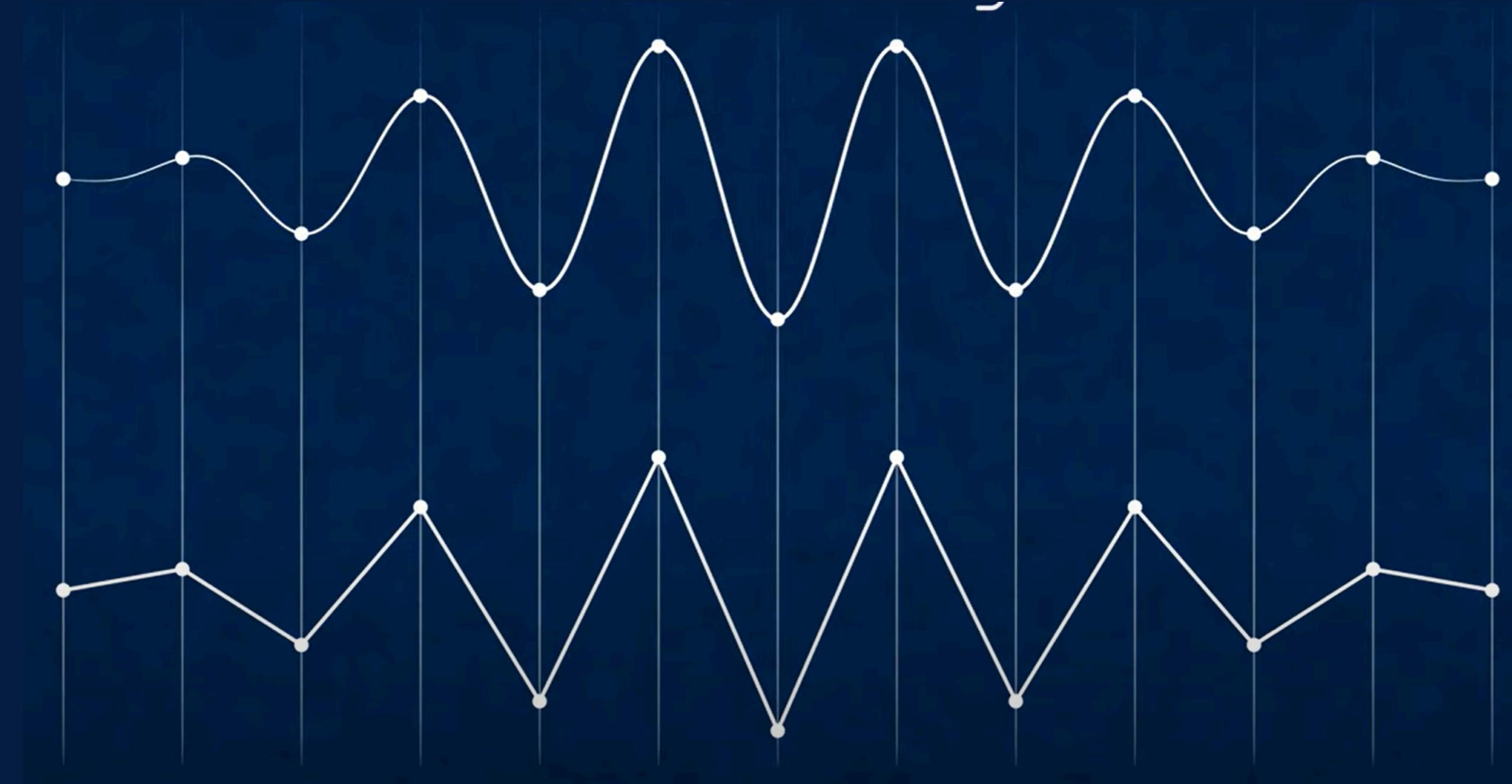


```
{'ImageWidth': 8000,  
 'ImageLength': 6000,  
 'BitsPerSample': (8, 8, 8),  
 'GPSInfo': {0: b'\x02\x03\x00\x00',  
 1: 'S',  
 2: (16.0, 39.0, 56.3423),  
 3: 'W',  
 4: (49.0, 15.0, 17.4592),  
 5: b'\x00',  
 6: 787.94},  
 'ResolutionUnit': 2,  
 'ExifOffset': 332,  
 'ImageDescription': 'default',  
 'Make': 'DJI',  
 'Model': 'MAVIC2-ENTERPRISE-ADVANCED',  
 'Software': '10.00.03.12',  
 'Orientation': 1,  
 'YCbCrPositioning': 1,  
 'DateTime': '2023:10:16 10:37:22',  
 'SamplesPerPixel': 3,  
 'XResolution': 72.0,  
 'YResolution': 72.0,  
 'ExifVersion': b'0230',  
 'FlashPixVersion': b'0100',  
 'DateTimeOriginal': '2023:10:16 10:37:22',  
 'DateTimeDigitized': '2023:10:16 10:37:22',  
 'ExposureBiasValue': 0.0,  
 'MaxApertureValue': 2.971,  
 'MeteringMode': 2,  
 'LightSource': 1,  
 'Flash': 0,
```

```
'FocalLength': 4.5,  
 'ColorSpace': 1,  
 'ExifImageWidth': 8000,  
 'ExifInteroperabilityOffset': 824,  
 'FocalLengthIn35mmFilm': 24,  
 'SceneCaptureType': 0,  
 'ExifImageHeight': 6000,  
 'Contrast': 0,  
 'Saturation': 0,  
 'Sharpness': 0,  
 'FileSource': b'\x03',  
 'ExposureTime': 0.0015625,  
 'FNumber': 2.8,  
 'SceneType': b'\x01',  
 'ExposureProgram': 2,  
 'ISOSpeedRatings': 100,  
 'ExposureMode': 0,  
 'WhiteBalance': 0,  
 'LensSpecification': (24.0, 24.0, 2.8, 2.8),  
 'DigitalZoomRatio': 1.0,  
 'GainControl': 0,  
 'GimbalDegree(Y,P,R)': -879,-777,  
 'FlightDegree(Y,P,R)': -1055,67,9,  
 'FlightSpeed(X,Y,Z)': (0, 0, 0)  
 }
```

Amostragem (Sampling)

- Processo de converter um sinal contínuo em um sinal discreto tomando amostras dos sinais em intervalos regular.
- O sampling divide a informação de variação de luz e cor (contínua) em uma grade finita e fixa de pontos.





Cada pixel é uma amostra da variação de luz (ou cor) em um ponto específico da imagem.



Largura x Altura



w

h

w = 736

h = 736

w . h = 541 696

imagem com
0,5 megapixel

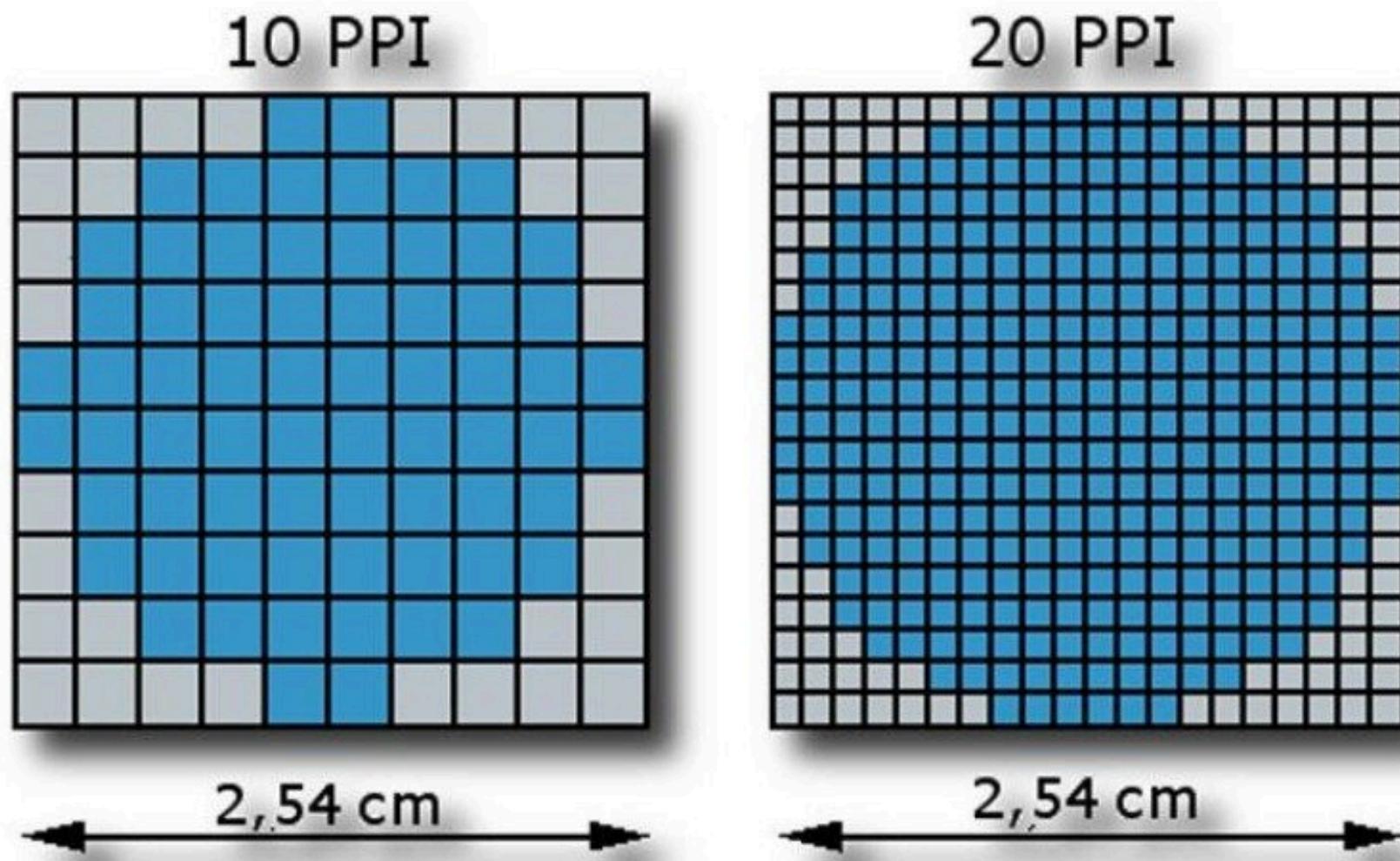
Mais megapixels, melhor?

Mais megapixels, melhor? Não necessariamente!



+ DETALHES
+ PESADO
+ NITIDEZ ??

Pixels Por Polegada (PPI)



- número de pixels presentes em uma polegada linear de uma tela digital

1 polegada = 1.54 cm

- Impressão de alta qualidade 300 PPI
- Tela de computador/móveis 72-150 PPI
- Outdoors/painéis grandes 10-30 PPI

Pontos Por Polegada (DPI)

- pontos de tinta que uma impressora pode colocar em uma polegada de papel



Aliasing

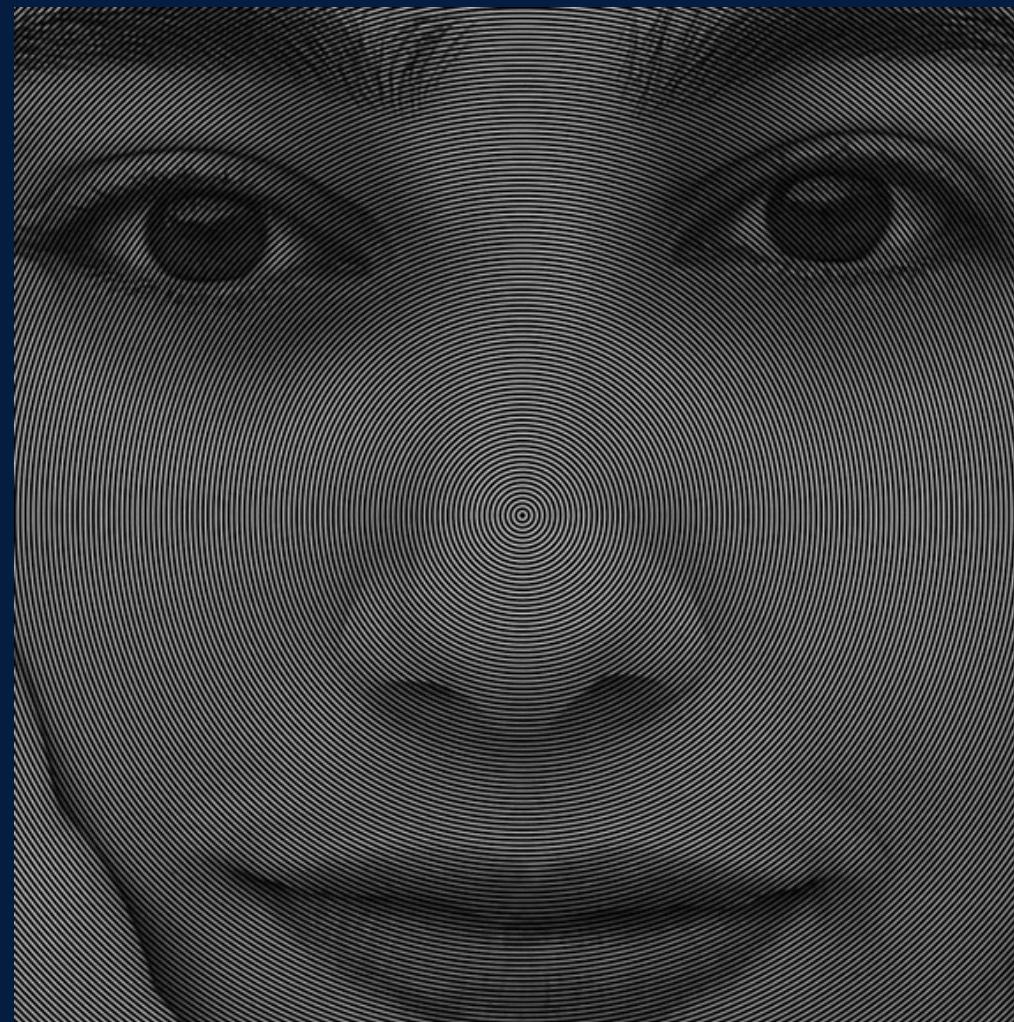
- amostragem insuficiente
- não captura a transição de luz e cor nas bordas da imagem
- principalmente em linhas diagonais, curvas e detalhes finos

Bordas Serrilhadas

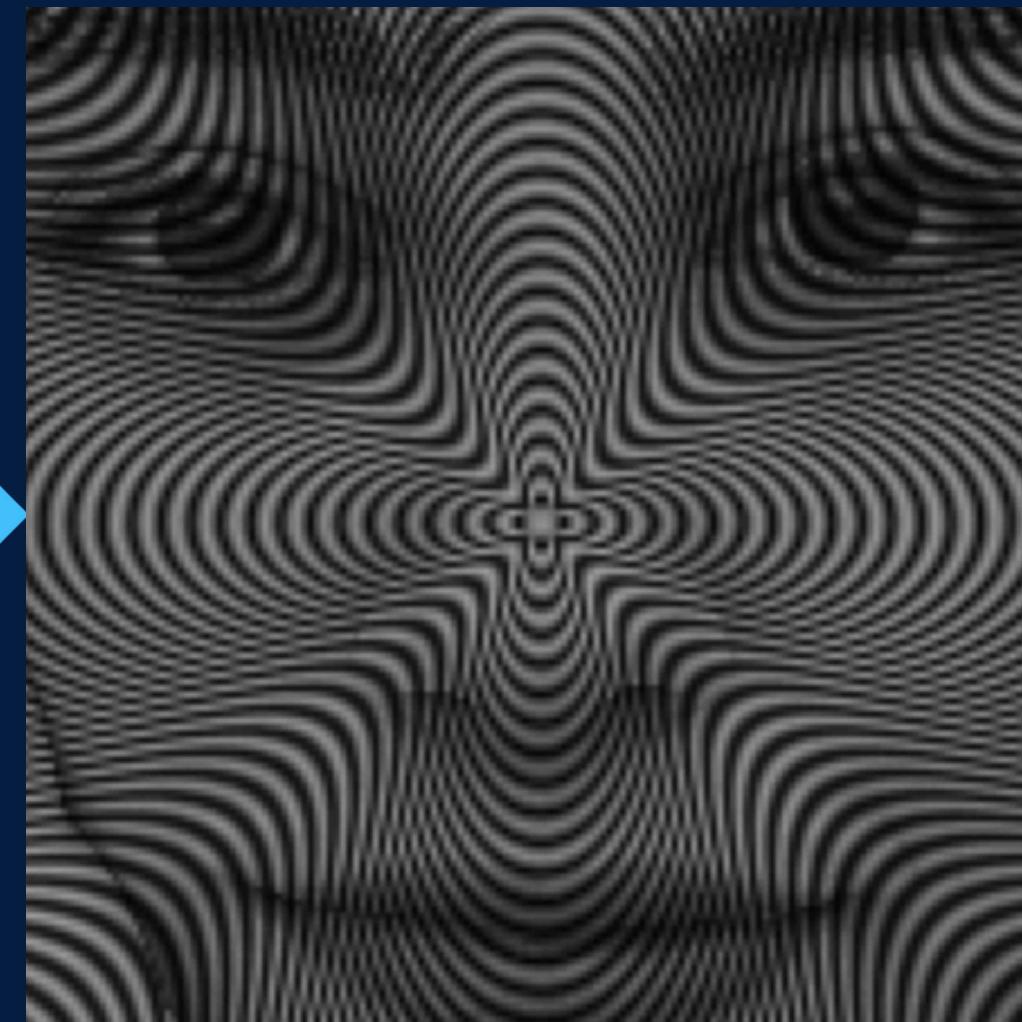


Aliasing

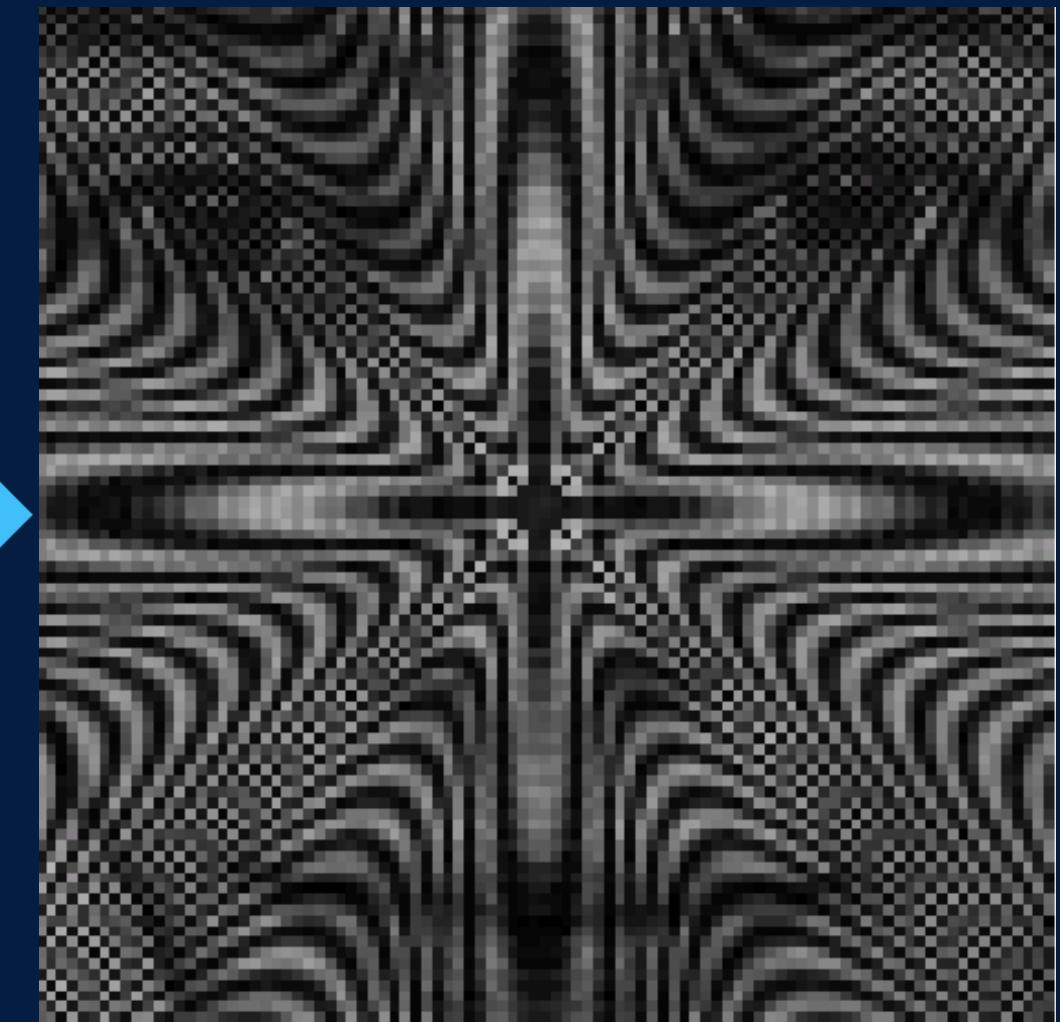
Efeito Moiré



Original



Resize 25%



Resize 15%

Antialiasing e Interpolação

Interpolação: cria pixels intermediários, ajudando a suavizar as bordas ou preencher lacunas em áreas de transição.

Ex: Nearest Neighbor, Bilinear, Bicúbica.

Antialiasing: usada para suavizar bordas serrilhadas, reduzindo o contraste entre pixels nas bordas, adicionando gradientes ou amostras extras

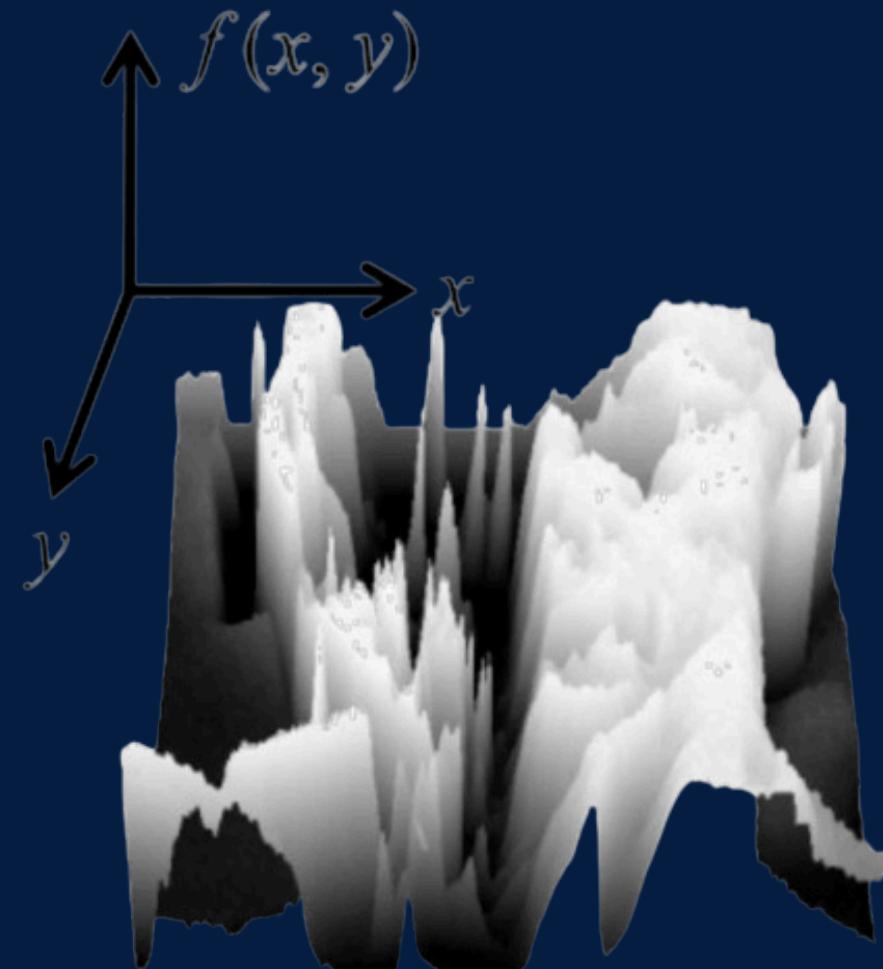
Ex: MSAA, FXAA, TAA, SSAA*



Multi-Sample Anti-Aliasing
Fast Approximate Anti-Aliasing
Temporal Anti-Aliasing
Super-Sampling Anti-Aliasing

Imagen como Função

Cada ponto (x, y) representa um pixel na imagem, e o valor $f(x, y)$ indica a intensidade da cor ou do brilho nesse ponto



Nesse caso, $f(x, y)$ indica a intensidade do brilho para uma imagem em preto e branco, variando de 0 (preto) até 1 (branco)

Transformações de Imagem



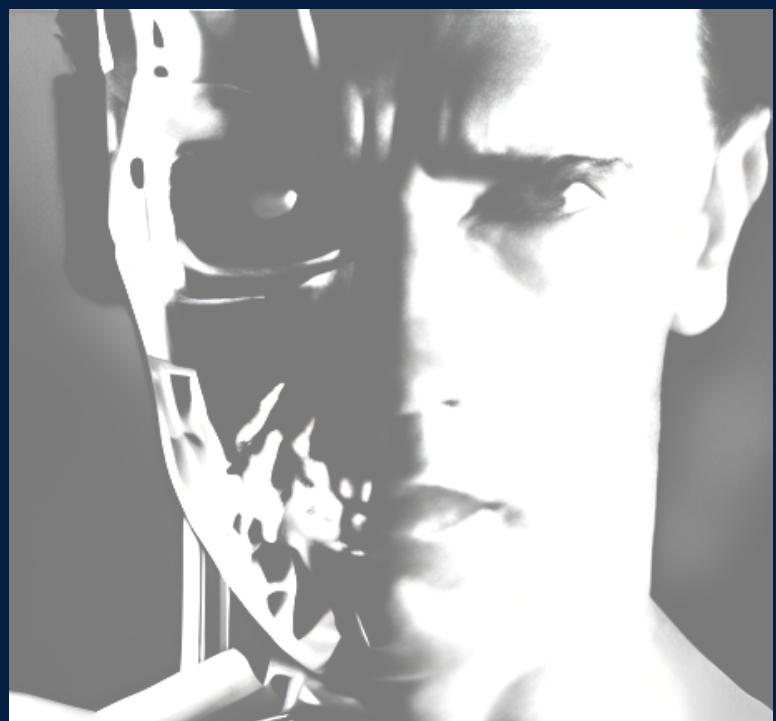
Original

$$f(x,y) = f(x,y)$$



**Rotação
(Espelhamento)**

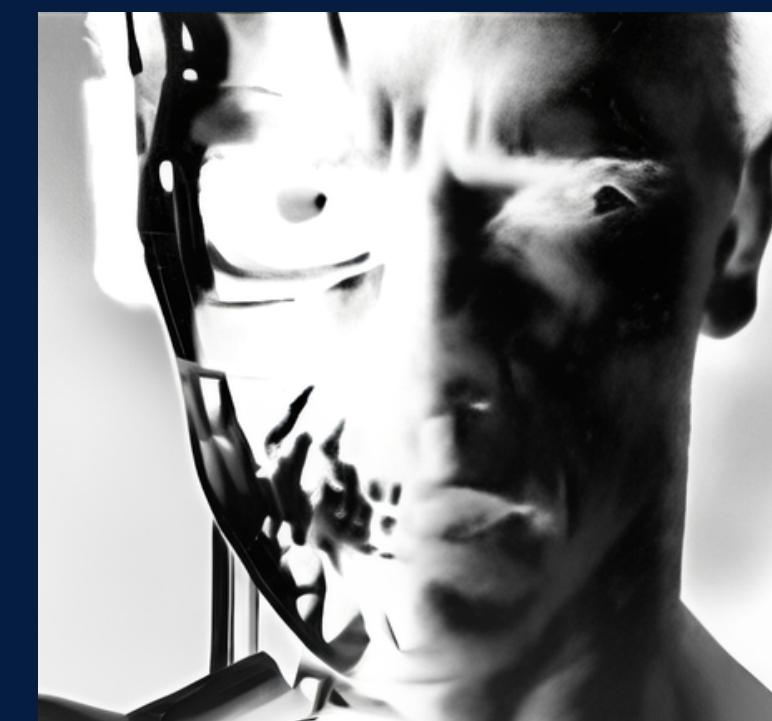
$$f(x,y) = f(-x,y)$$



Aumento dos Valores

$$f(x,y) = f(x,y) + 0.5$$

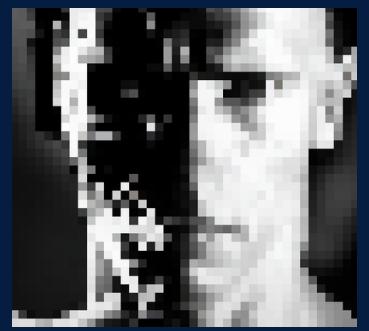
Obs: Limite = 1



Inversão de Cores

$$f(x,y) = 1 - f(x,y)$$

Mudando Tamanho de Imagem

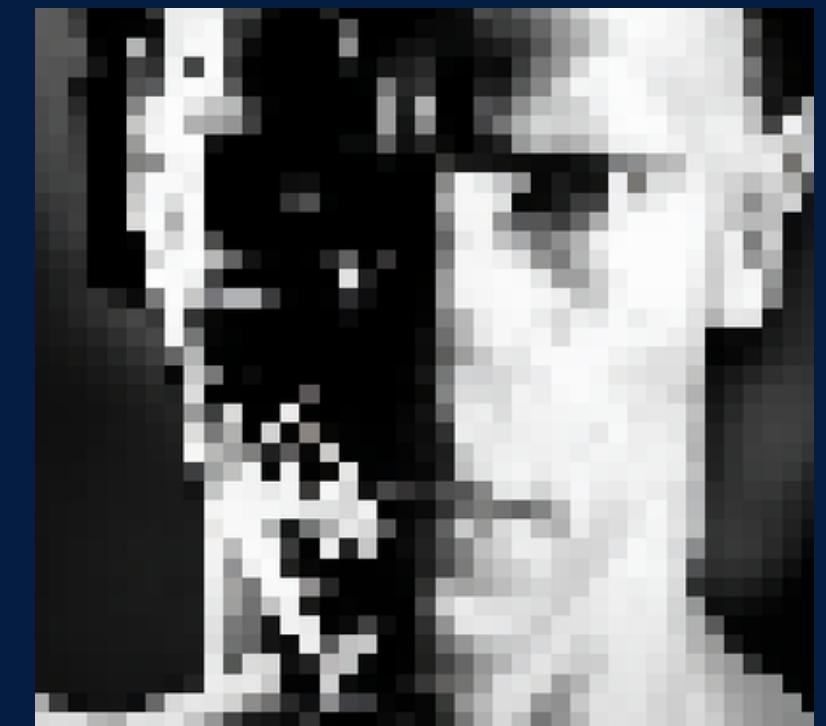
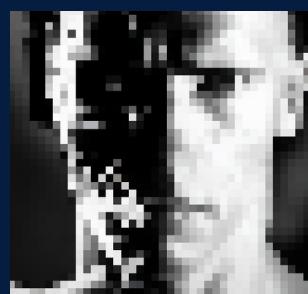


Diminuindo

Compactamos a função, pegando menos pontos da imagem original para representar a nova imagem.

Aumentando

Eticamos a função, usando interpolação para preencher novos pontos com valores estimados dos pixels vizinhos.

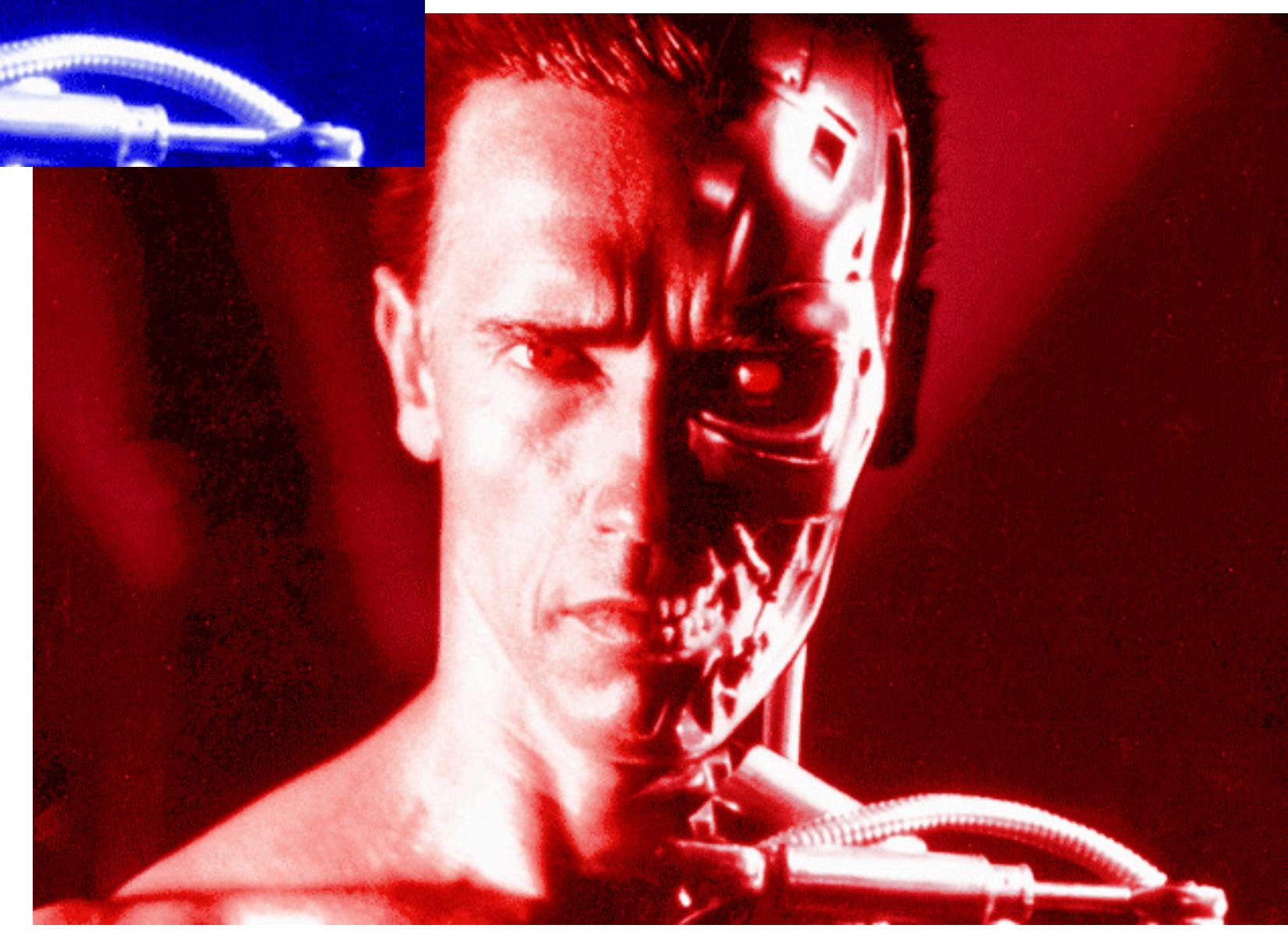
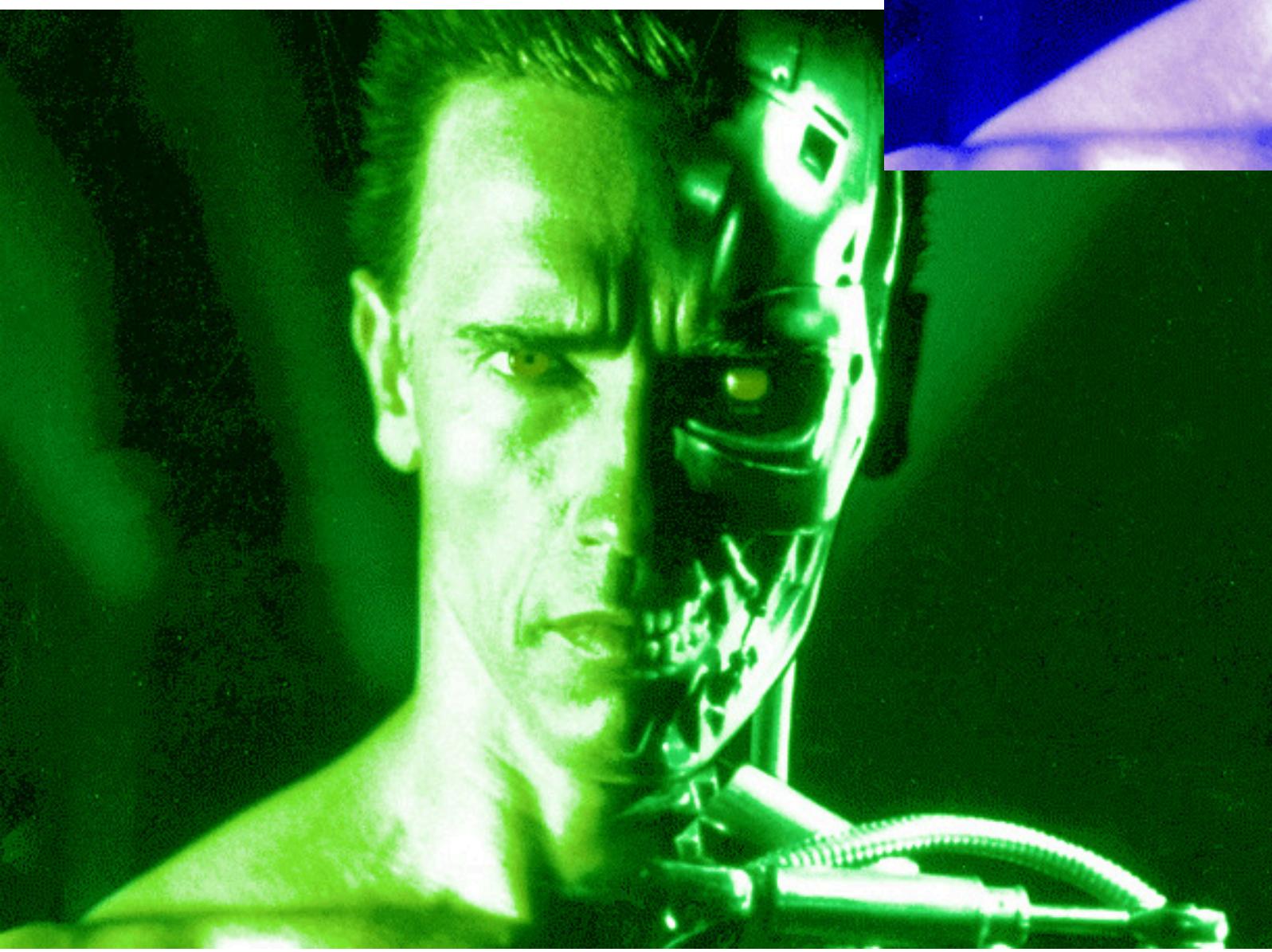


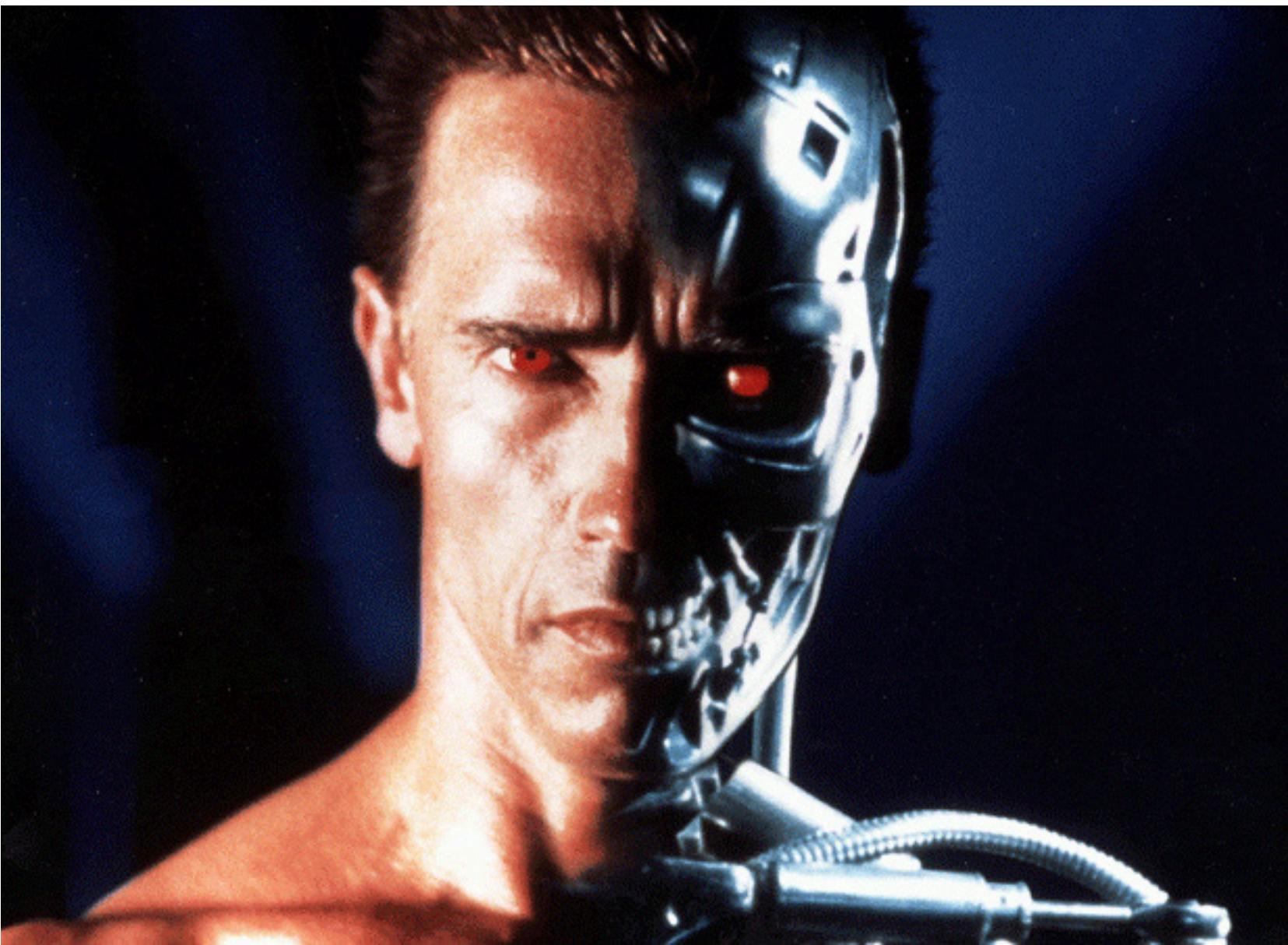
Como Representar uma Imagem Colorida ?

Em vez de armazenar apenas a intensidade de brilho (como nas imagens em preto e branco), cada pixel de uma imagem colorida armazena três valores: intensidade de vermelho, verde e azul.

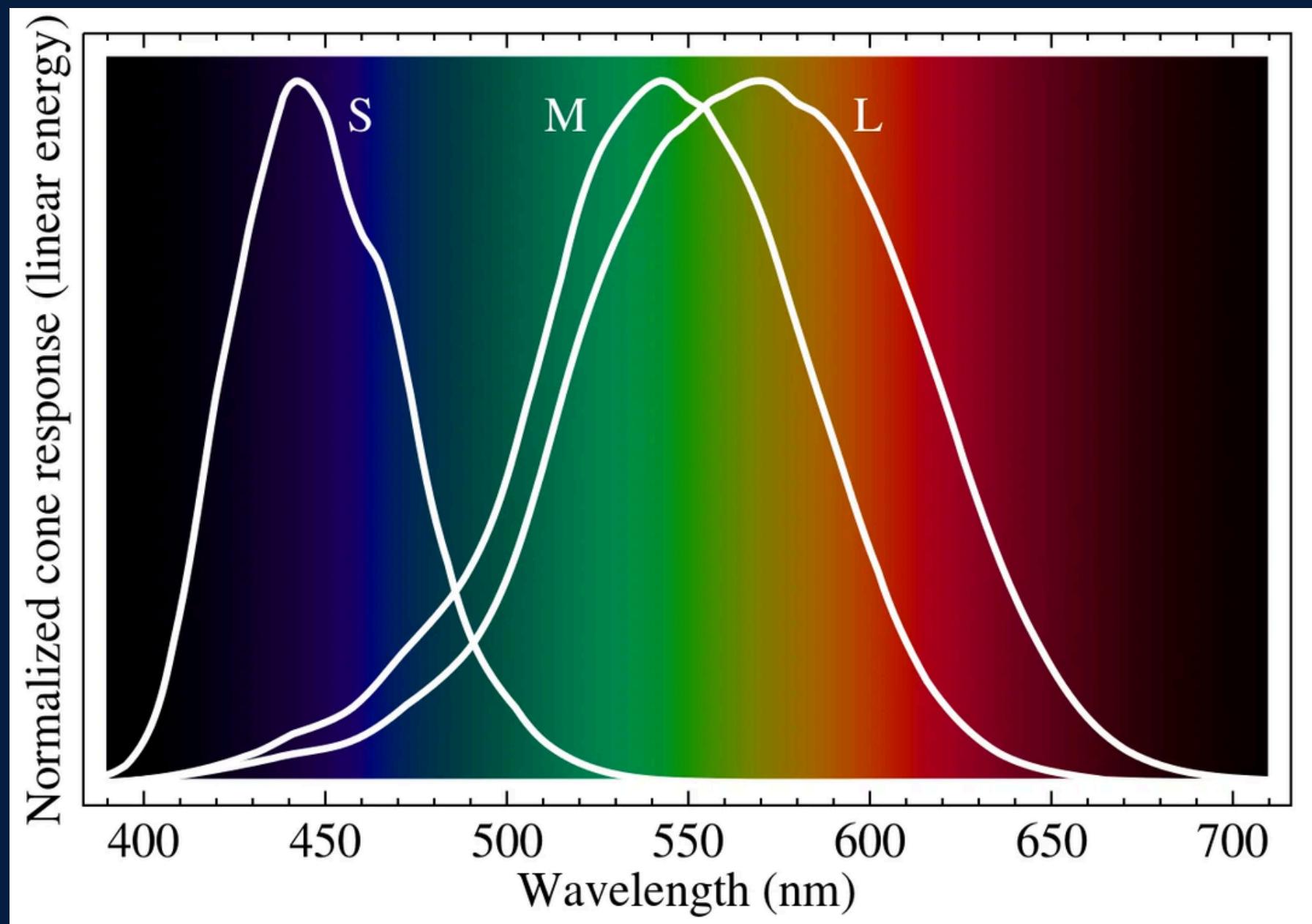
$$f(x,y) = [R(x,y), G(x,y), B(x,y)]$$



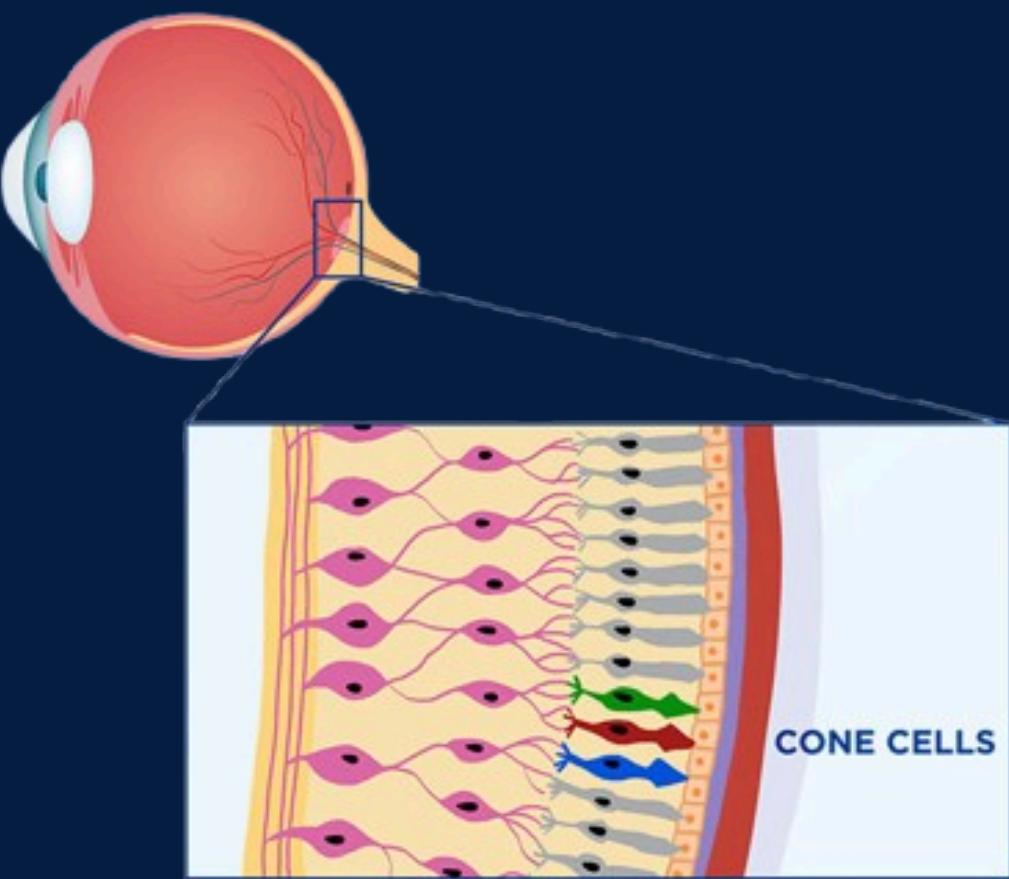




Por que RGB ?

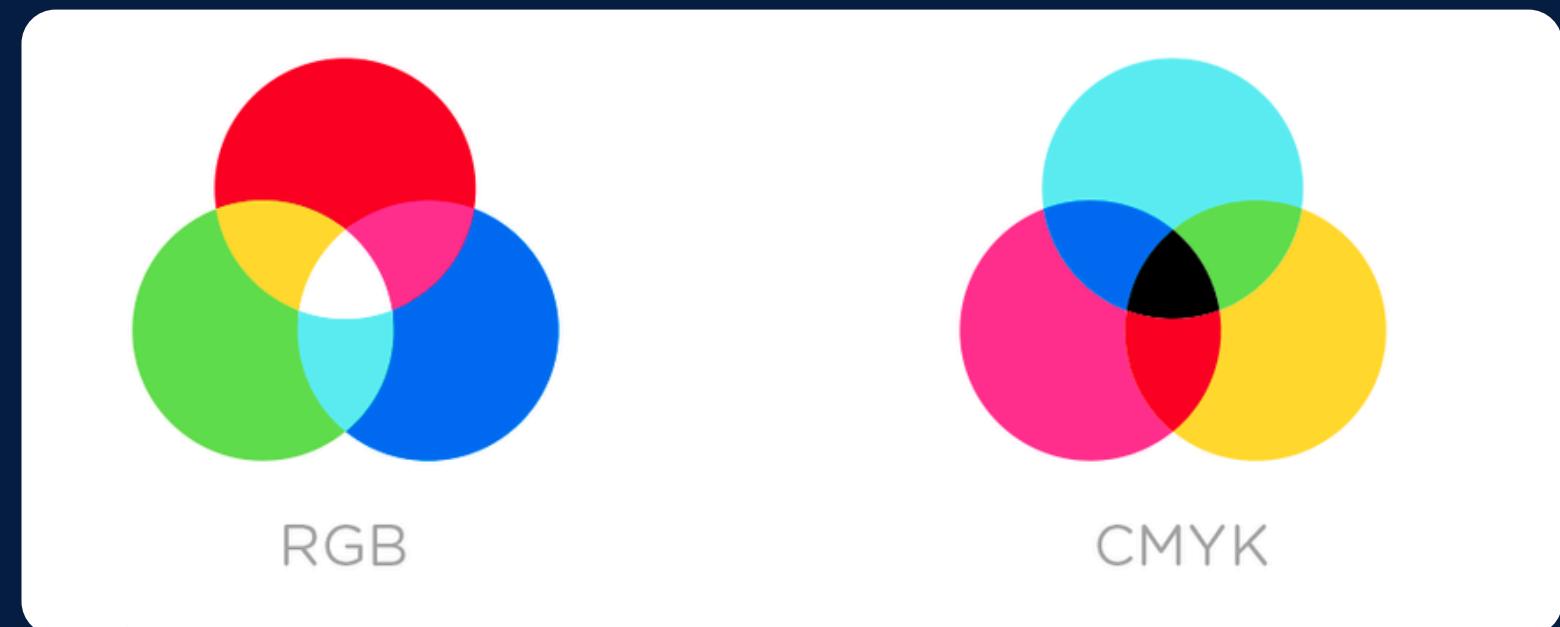


Usamos RGB porque é baseado na percepção humana. Nossos olhos têm três tipos de cones (L, M, S) que são sensíveis ao vermelho, verde e azul. Combinando essas cores de luz, conseguimos simular a maioria das cores visíveis.



Cores: Aditivas x Subtrativas

- As cores aditivas são criadas pela adição de luz.
- As três cores primárias no sistema aditivo são vermelho, verde e azul (RGB). Quando combinadas ao máximo, produzem luz branca.
- Esse sistema é utilizado em dispositivos que emitem luz.
- As cores subtrativas são criadas pela subtração de luz.
- As três cores primárias subtrativas são ciano, magenta e amarelo (CMY). Quando combinadas, filtram a luz branca.
- O sistema subtrativo é utilizado em dispositivos que refletem luz.



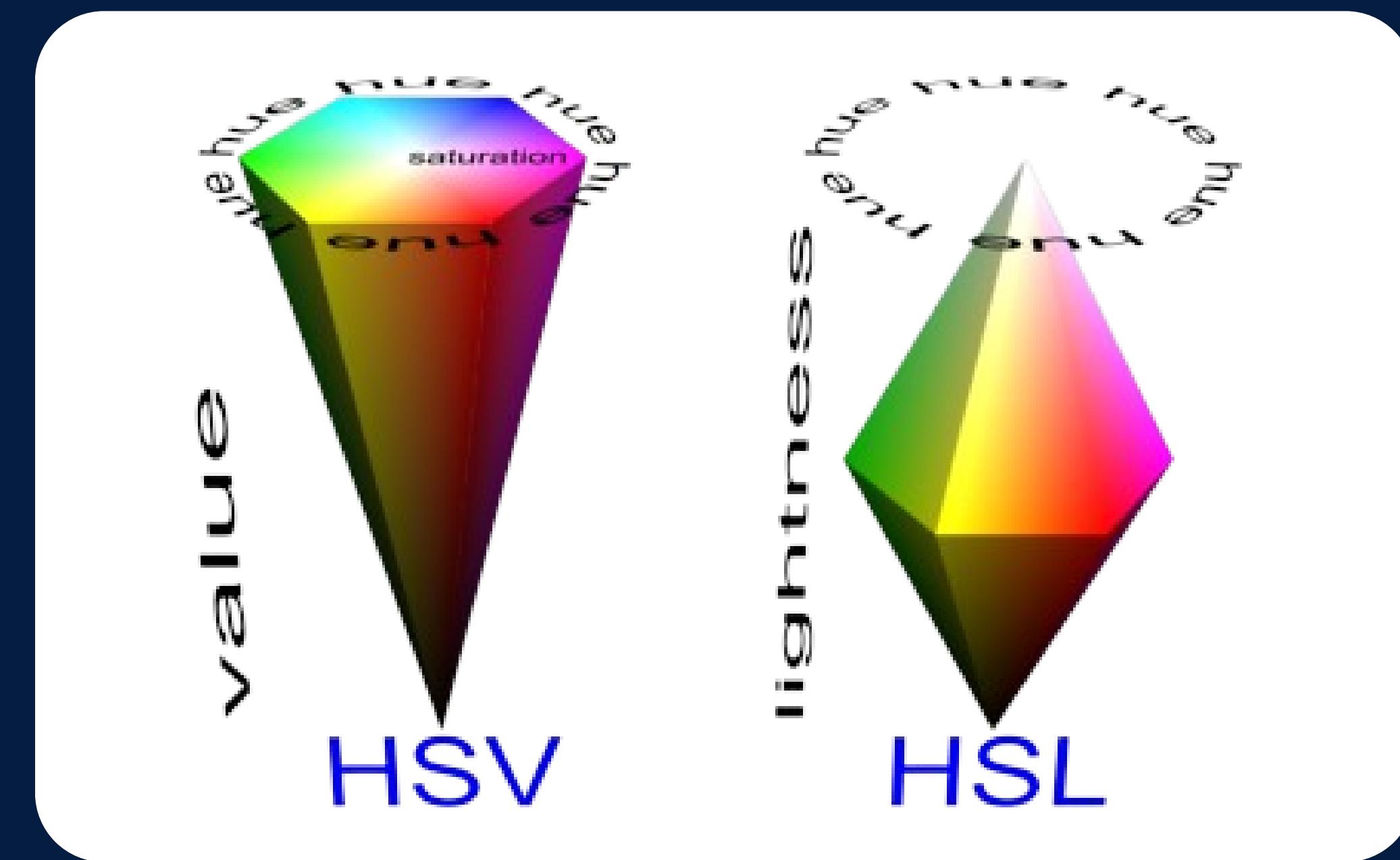
Espaços de Cor

Um sistema para representar as cores de maneira que elas possam ser reproduzidas e interpretadas consistentemente por diferentes dispositivos e processos.

HSV e HSL

Hue
Saturation
Value
Lightning

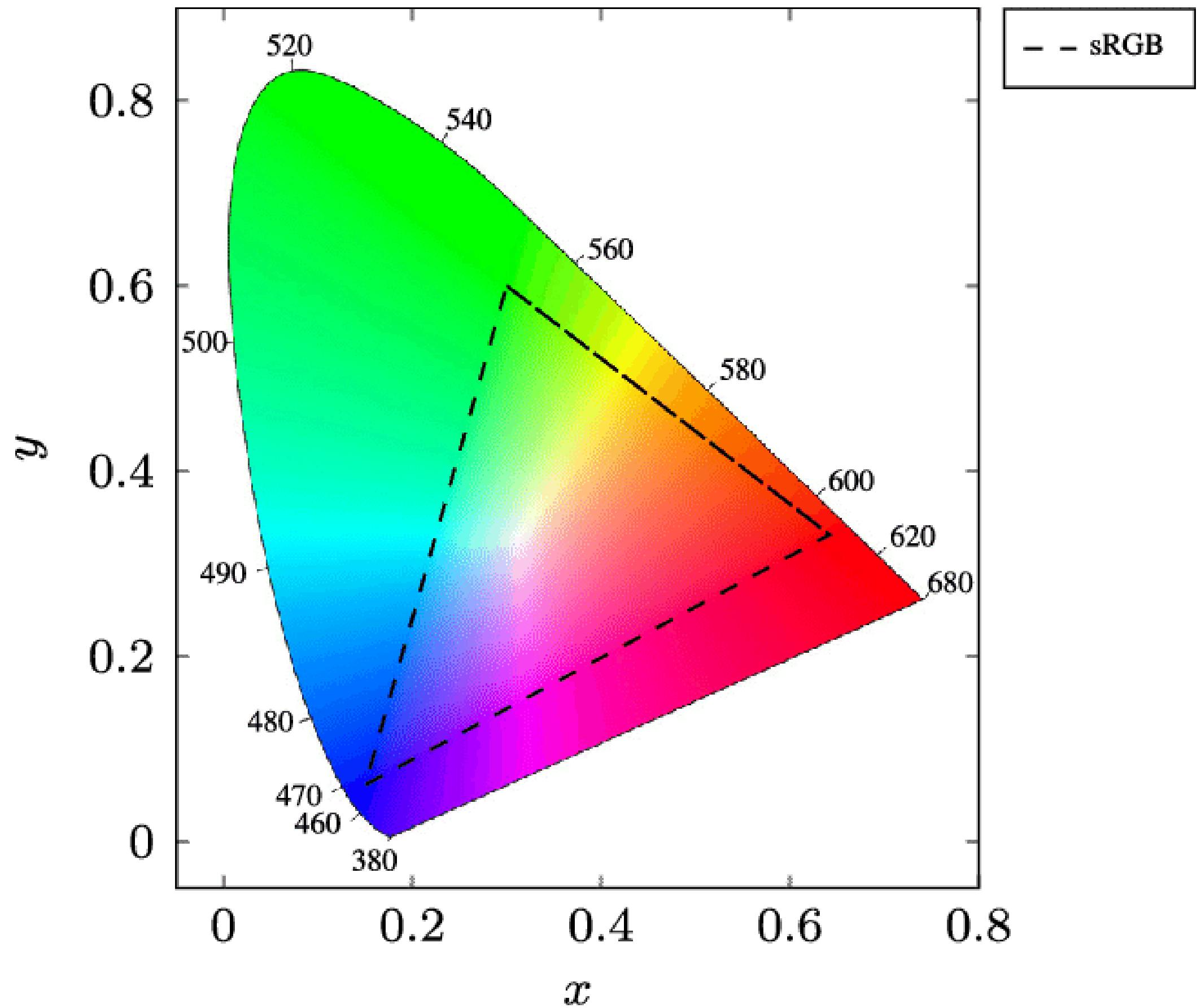
HSV
HSL
AMBOS



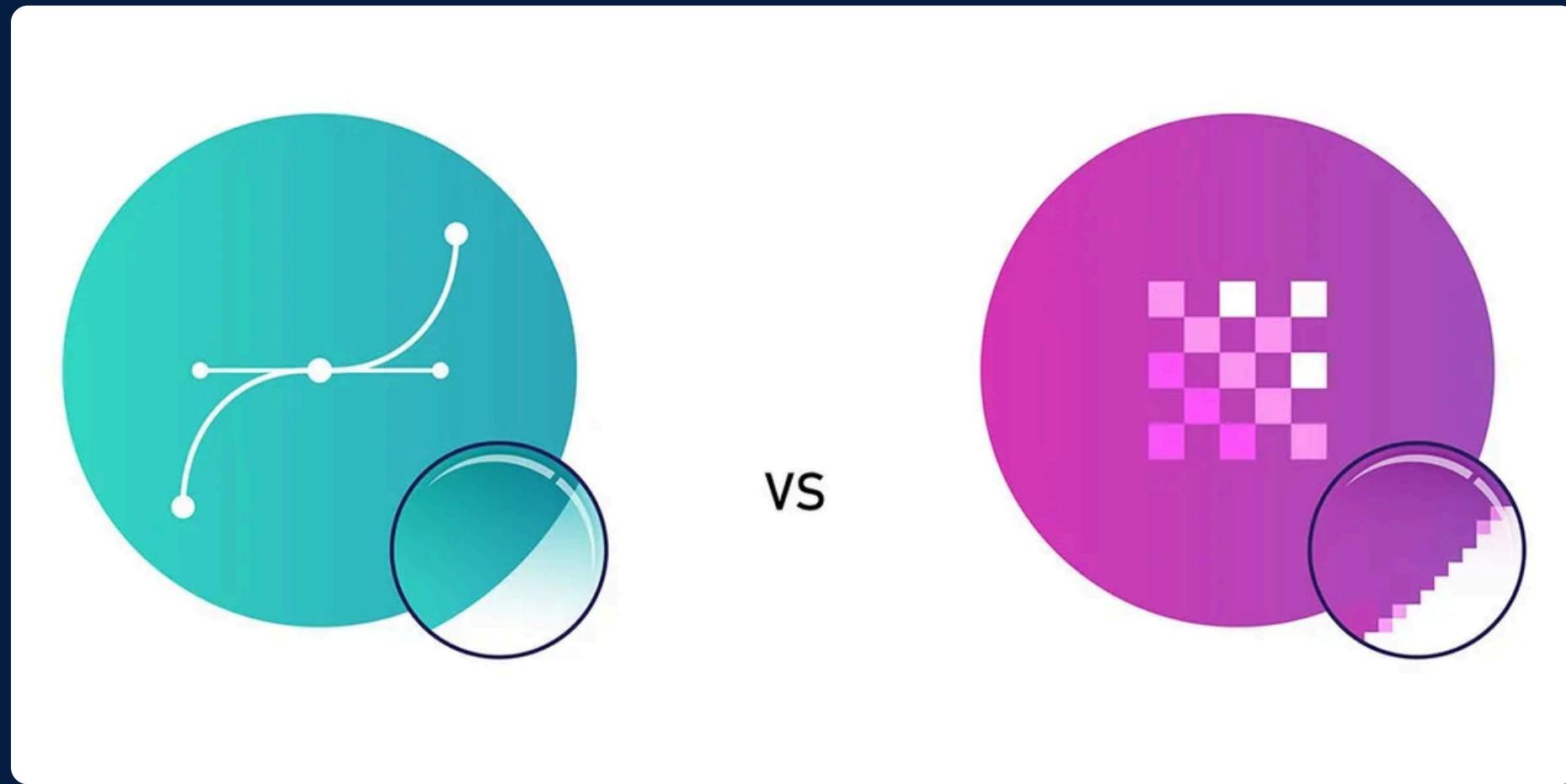
CIE 1931

Comissão Internacional de Iluminação

O modelo CIE 1931 é um espaço de cor que tenta representar todas as cores visíveis para o olho humano.



Raster x Vetorial



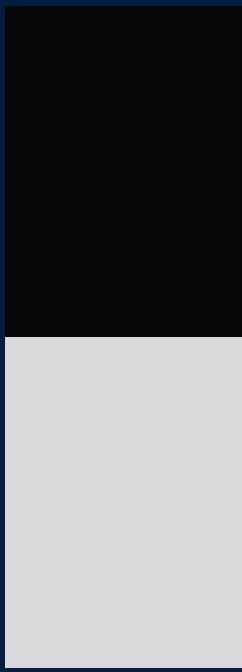
As imagens raster são compostas por uma grade de pixels, onde cada pixel representa uma cor. Em uma imagem raster, a resolução e a quantidade de detalhes dependem do número de pixels que a compõem.

As imagens vetoriais são baseadas em equações matemáticas que descrevem formas geométricas como linhas, curvas, polígonos e círculos.

Formatos de Imagem

Métodos de Compactação

Memória Ocupada



Monocromática

0 ou 1
 $1 \times (1 \text{ bit}) = 1 \text{ bit}$



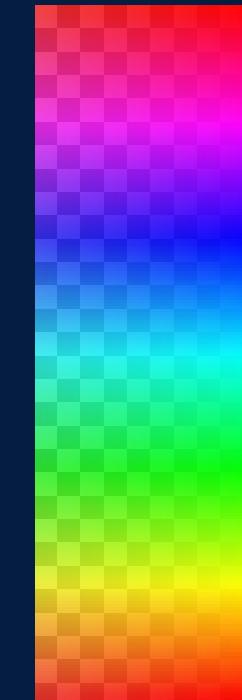
Escala de Cinza

0 a 255
 $1 \times (8 \text{ bits}) = 8 \text{ bits}$



Escala RGB

0 a 255
 $3 \times (8 \text{ bits}) = 24 \text{ bits}$



Escala RGBA

0 a 255
 $4 \times (8 \text{ bits}) = 32 \text{ bits}$



1.000 pixels

$$\text{total} = 1.000 \times 1.000 = 1.000.000 \text{ pixels}$$

1.000 pixels



Monocromático:
1.000.000 bits
125.000 bytes
125 KB

Escala de Cinza:
8.000.000 bits
1.000.000 bytes
1 MB

Escala RGB:
24.000.000 bits
3.000.000 bytes
3 MB

Escala RGBA:
32.000.000 bits
4.000.000 bytes
4 MB

RAW



Armazena os dados capturados diretamente pelo sensor da câmera, sem compressão ou processamento.

JPEG (Joint Photographic Experts Group)

- Criado em 1992
- Imagens apresentam muito ruídos
- Ocorre a compressão com perda

O segredo da compressão:

DESCATAR AQUILO QUE NÍNGUEM SE IMPORTA!

O olho humano possui 120 milhões de células bastonetes sensíveis ao brilho e 6 milhões de células cones sensíveis à cor.



O que é compressão?

Procura por padrões repetidos nos dados para substituição desses padrões por versões mais curtas.

Altas Frequências



1) Isolando Informações de Cor



Calculando o Y:

$$Y = 0,299 \times R + 0,587 \times G + 0,114 \times B$$

```
from PIL import Image
import cv2
import numpy as np

# Carregar a imagem em formato RGB
image_path = 'image.jpg'
rgb_image = Image.open(image_path)
rgb_image = np.array(rgb_image)

# Converter a imagem RGB para YCbCr usando OpenCV
ycbcr_image = cv2.cvtColor(rgb_image, cv2.COLOR_RGB2YCrCb)

# Dividir os canais Y, Cb e Cr
Y, Cb, Cr = cv2.split(ycbcr_image)

# Salvar cada canal como uma imagem em escala de cinza
cv2.imwrite("canal_Y.jpg", Y)
cv2.imwrite("canal_Cb.jpg", Cb)
cv2.imwrite("canal_Cr.jpg", Cr)

# Salvar a imagem original em YCbCr
cv2.imwrite("imagem_YCbCr.jpg", ycbcr_image)
```



Y



Cr



Cb

2) Descartar algumas informações de cor



Cr: resize(1/4)

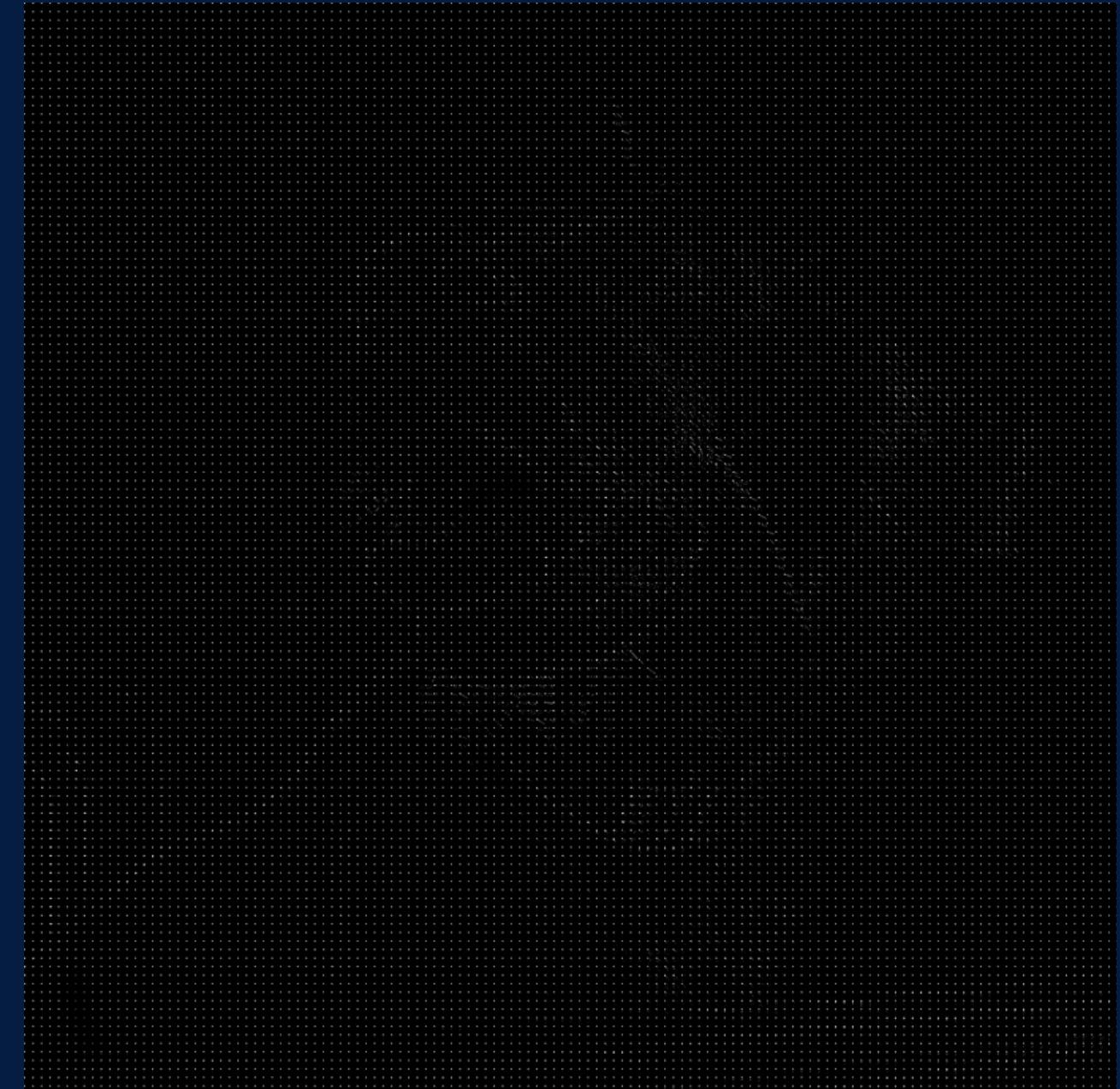


Cb: resize(1/4)

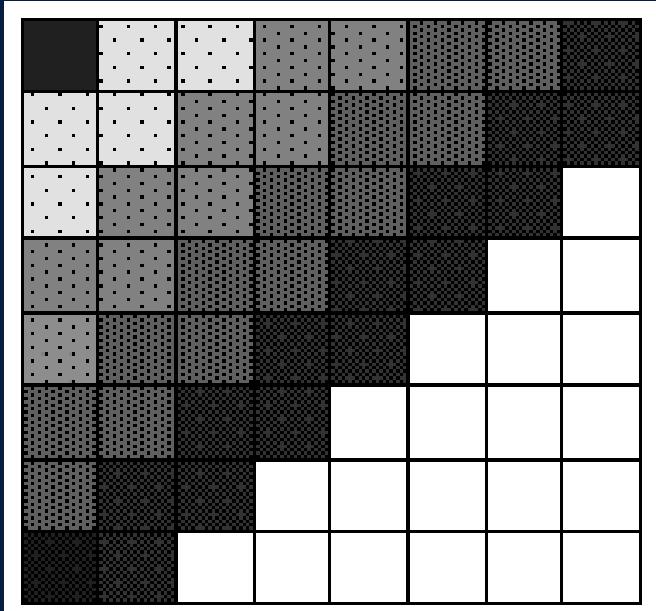


3) Converter para o domínio de frequência

- Divisão em blocos de 8x8
- Transformada Discreta de Cosseno em 2D (DCT)



4) O controle de qualidade (quantização)



Informações de alta frequência sofrem uma divisão maior

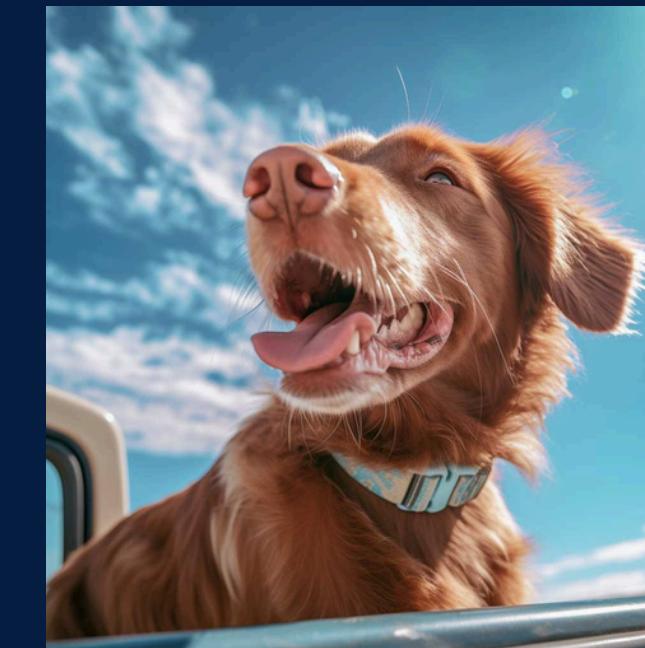
Luminance (brightness) table									Chrominance (colour) table								
26	19	22	22	29	38	78	115		27	29	38	75	158	158	158	158	
18	19	21	27	35	56	102	147		29	34	42	106	158	158	158	158	
16	22	26	35	59	88	125	152		38	42	90	158	158	158	158	158	
26	30	38	46	90	102	139	157		75	106	158	158	158	158	158	158	
38	42	64	82	109	130	165	179		158	158	158	158	158	158	158	158	
64	93	91	139	174	166	194	160		158	158	158	158	158	158	158	158	
82	96	110	128	165	181	192	165		158	158	158	158	158	158	158	158	
98	88	90	99	123	147	162	158		158	158	158	158	158	158	158	158	

5) Compressão de dados sem perdas



$$\begin{aligned}n &= 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, \dots \\n/2 &= 0, 1, 1, 2, 2, 3, 3, 4, 5, 5, 5, 6, 6, 7, 7, 8, 8, \dots \\n/16 &= 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, \dots\end{aligned}$$

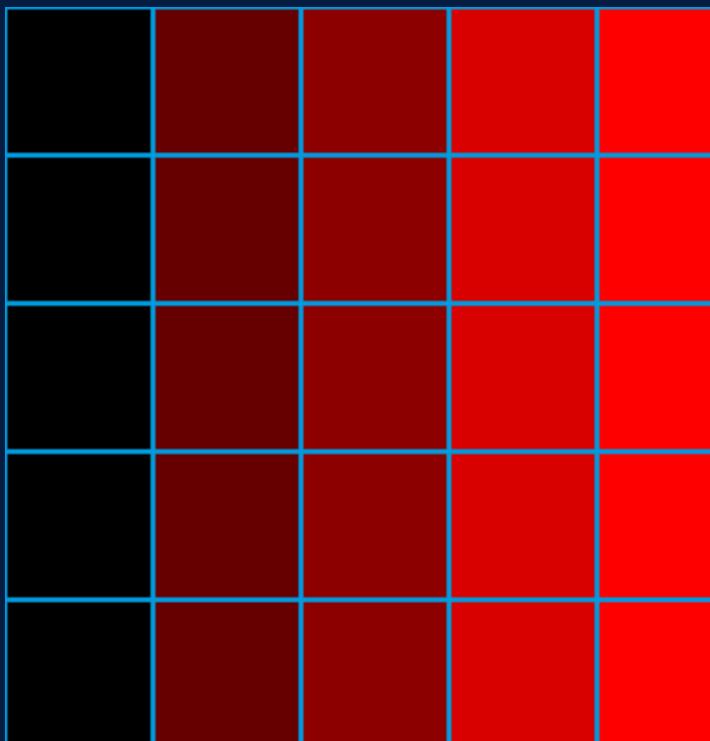
TEMOS UMA IMAGEM COMPRIIMIDA EM JPEG!



PNG (Portable Network Graphics)

E um formato de imagem digital sem perdas.
Suporta uma maior profundidade de cor e,
ao mesmo tempo, oferece uma compressão
eficiente.

1) Pré-Filtragem

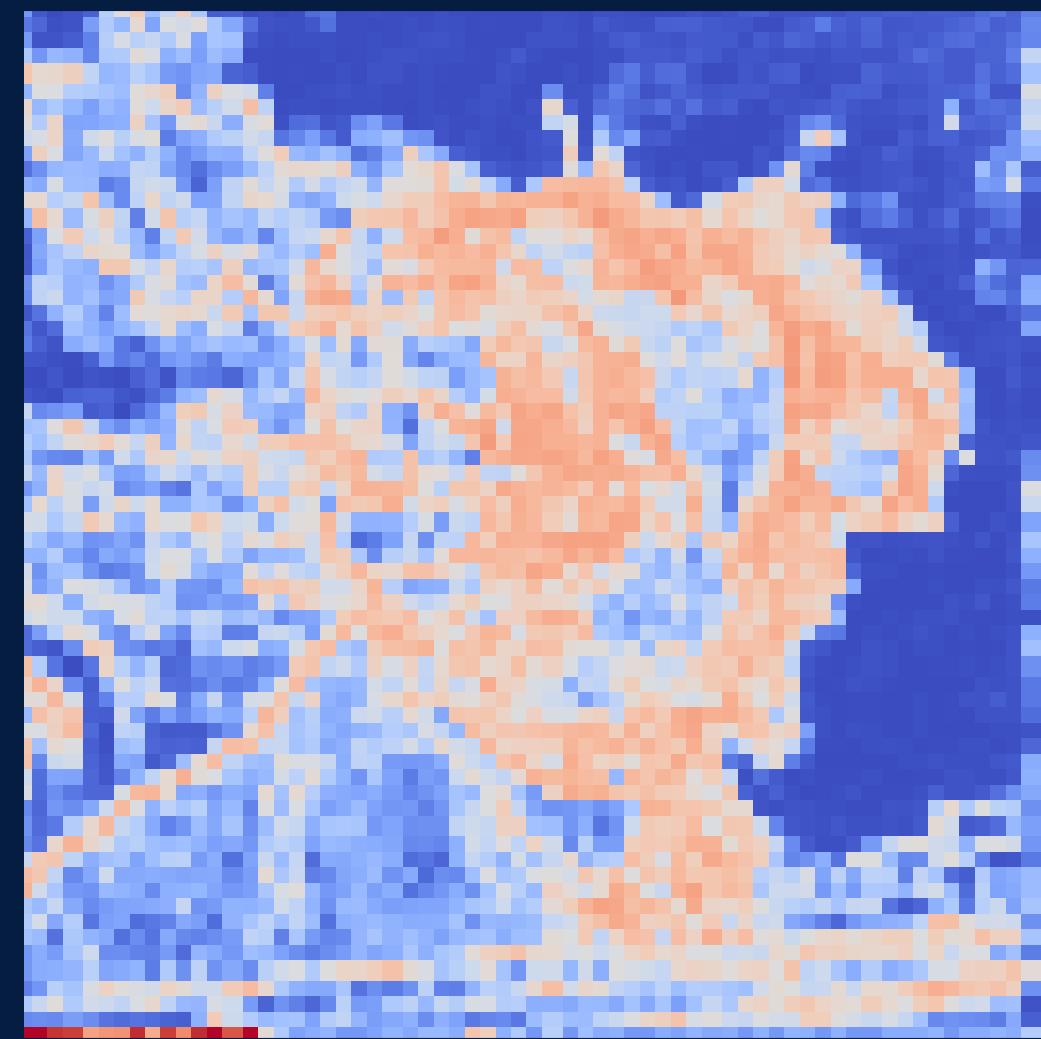
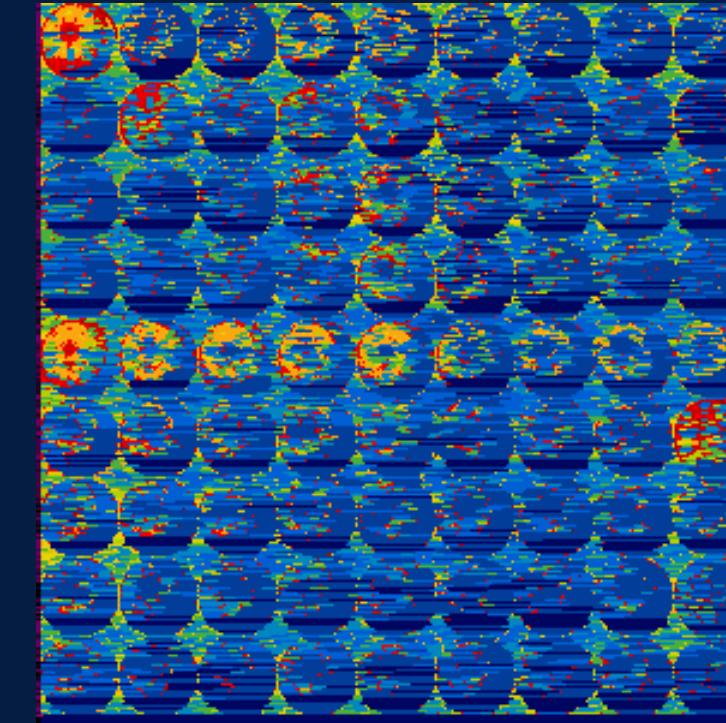
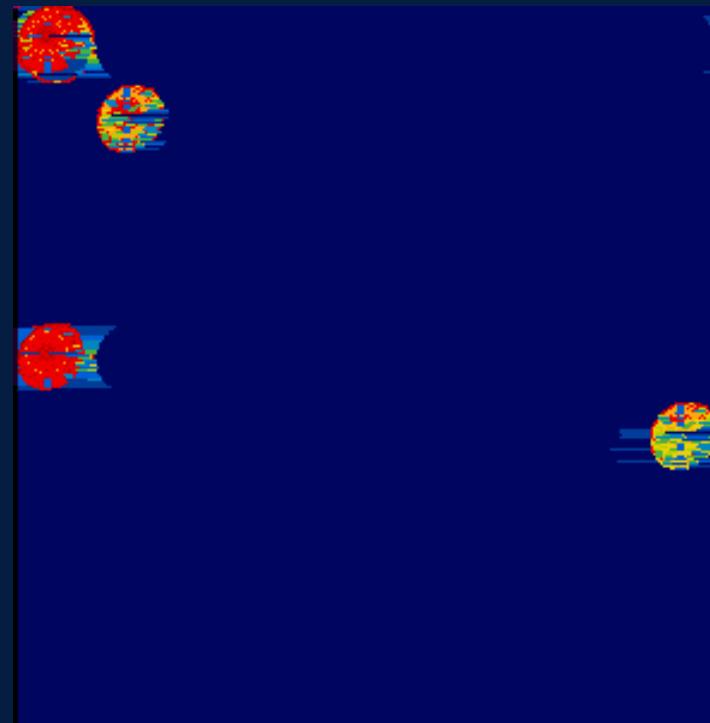
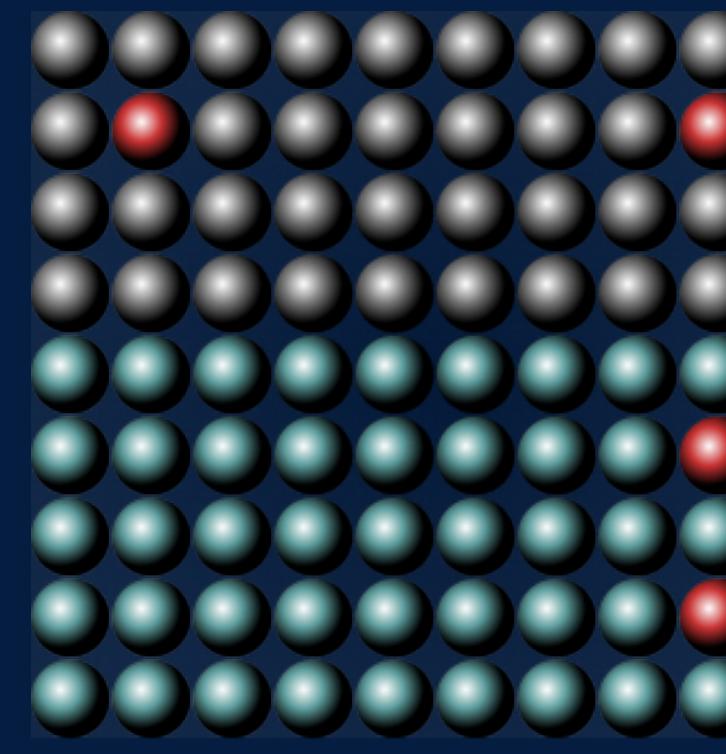
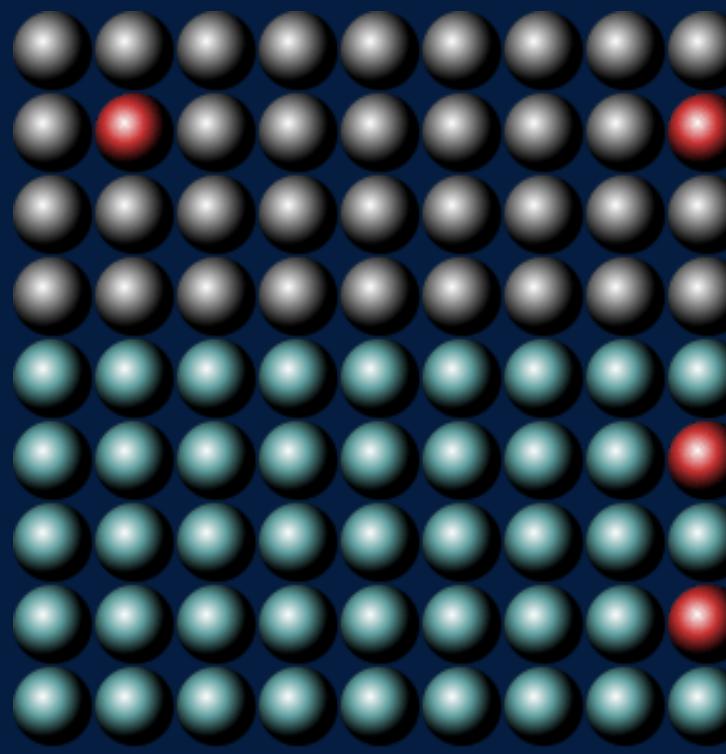


1	2	3	4	5
1	2	3	4	5
1	2	3	4	5
1	2	3	4	5
1	2	3	4	5

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

Em cada linha, é usado o filtro **sub** que registra a diferença em relação ao valor anterior.

2) Codificação baseada em dicionário usando LZ77



O azul escuro indica que essa área ocupa virtualmente nenhuma memória, azul claro um pouco de memória, amarelo mais memória e vermelho muita memória.

TIFF (Tagged Image File Format)

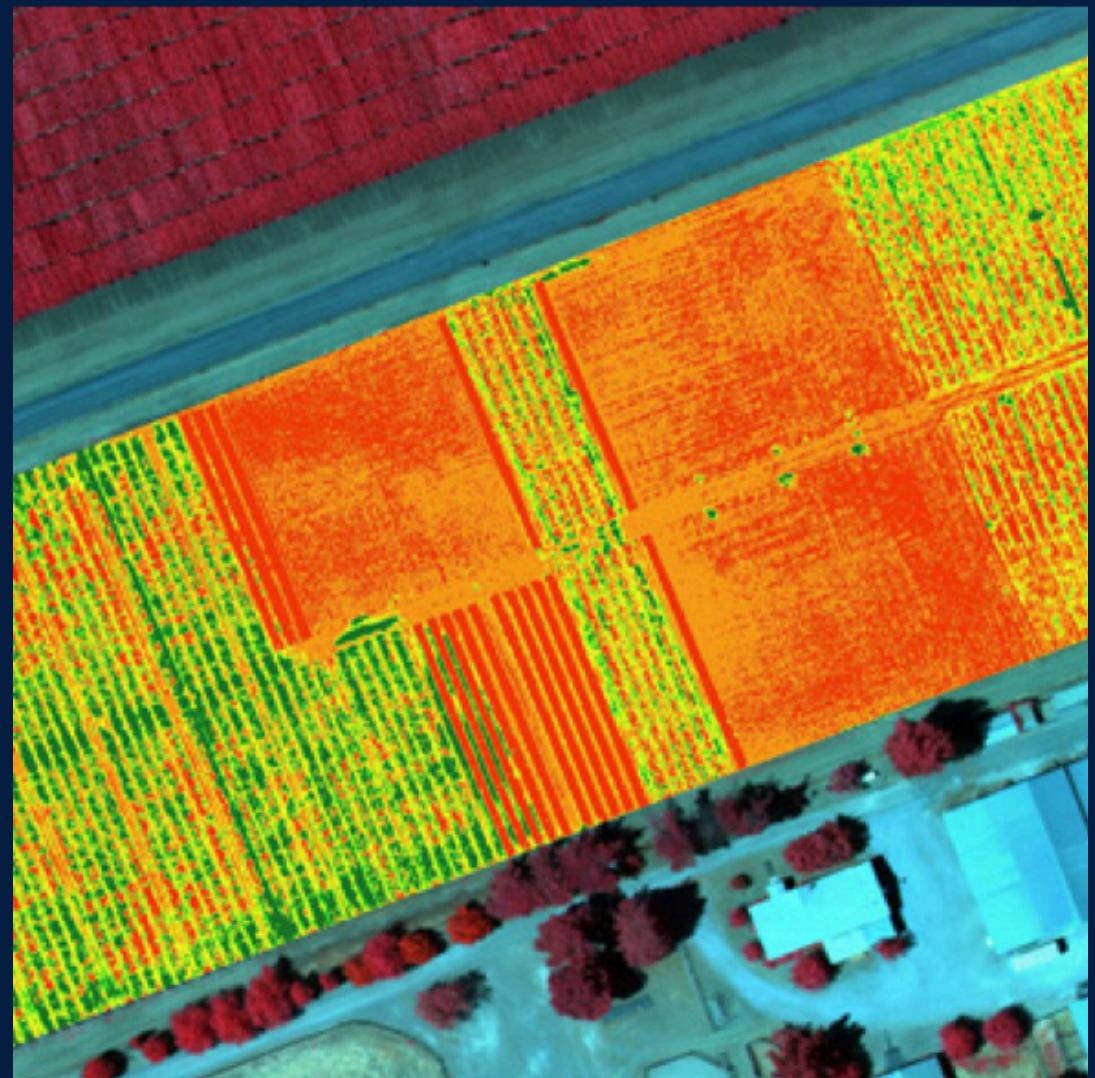
Formato de arquivo gráfico usado para imagens, identificado pela extensão de arquivo .tif. Pode ser com perda ou sem perda, suporta várias camadas e transparências.



Geração do ortomosaico a partir das imagens capturadas.

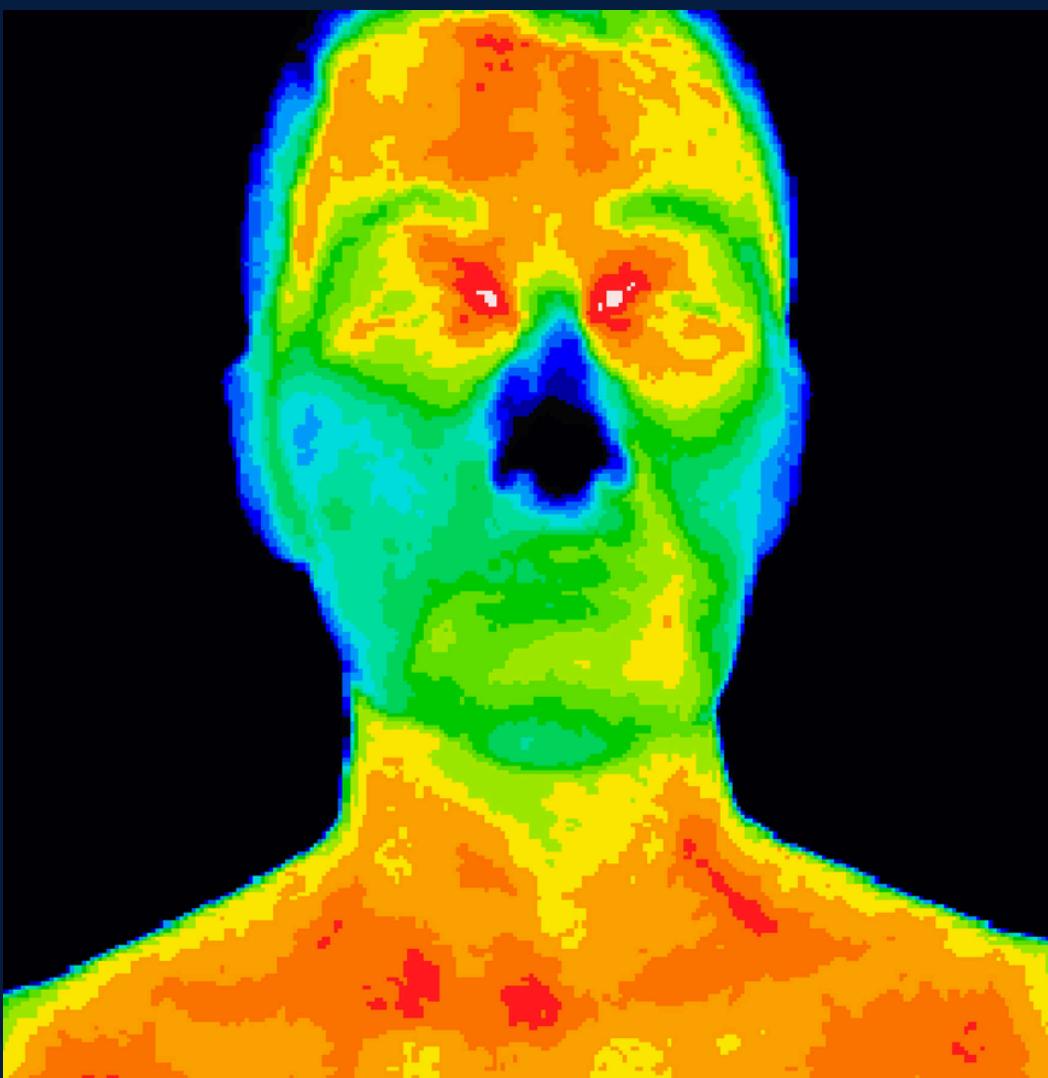
TIFF - Vantagens

Mais Camadas



Multispectros

Maior Profundidade de Cor



4.294.967.296 (32 bits)

Mais metadados

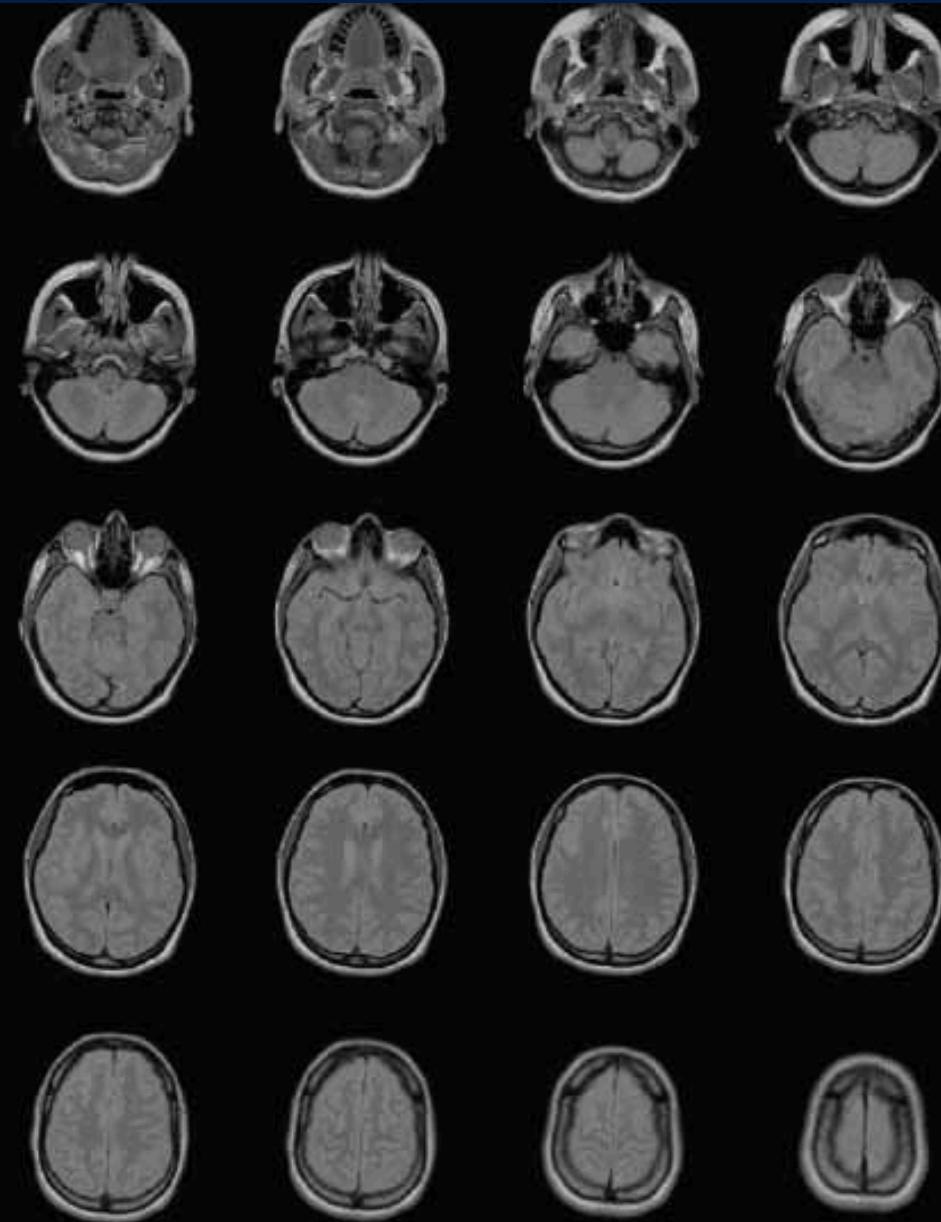
```
● ● ●
{'ImageWidth': 8000,
'ImageLength': 6000,
'BitsPerSample': (8, 8, 8),
'GPSInfo': {0: b'\x02\x03\x00\x00',
1: 'S',
2: (16.0, 39.0, 56.3423),
3: 'W',
4: (49.0, 15.0, 17.4592),
5: b'\x00',
6: 787.94},
'ResolutionUnit': 2,
'ExifOffset': 332,
'ImageDescription': 'default',
'Make': 'DJI',
'Model': 'MAVIC2-ENTERPRISE-ADVANCED',
'Software': '10.00.03.12',
'Orientation': 1,
'YCbCrPositioning': 1,
'DateTime': '2023:10:16 10:37:22',
'SamplesPerPixel': 3,
'XResolution': 72.0,
'YResolution': 72.0,
'ExifVersion': b'0230',
'FlashPixVersion': b'0100',
'DateTimeOriginal': '2023:10:16 10:37:22',
'DateTimeDigitized': '2023:10:16 10:37:22',
'ExposureBiasValue': 0.0,
'MaxApertureValue': 2.971,
'MeteringMode': 2,
```

Informações EXIF

DICOM (Digital Imaging and Communications in Medicine)

Padrão internacional para a criação, armazenamento, transmissão e visualização de imagens médicas.

Vai além de uma imagem, permite armazenar metadados específicos sobre o paciente (dados pessoais) e informações sobre o exame (data, tipo). Permite privacidade dos dados por meio de criptografia.



Dúvidas?