

# VISÃO COMPUTACIONAL

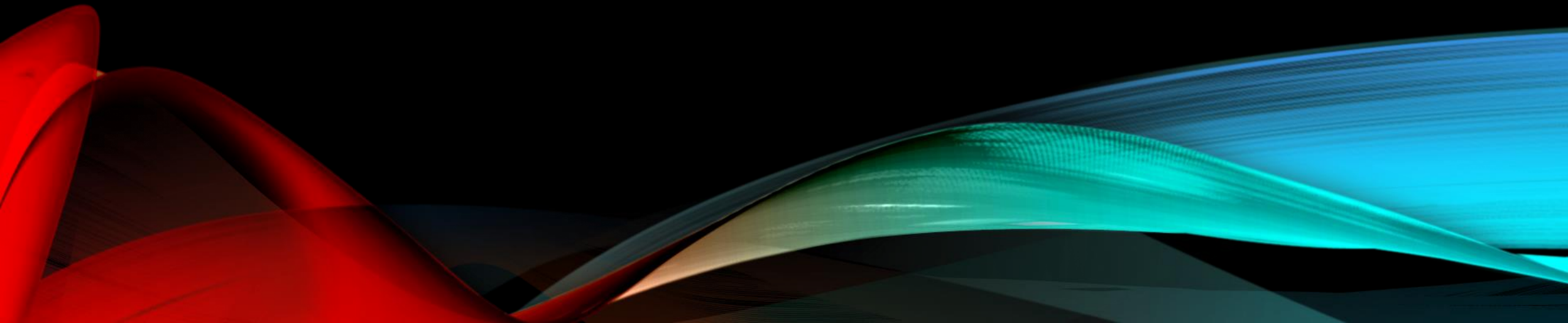
Tópicos selecionados e aplicações com a biblioteca openCV



# IMAGENS

1. Câmeras
  1. O modelo da câmera escura simples.
2. Imagens
  1. Amostragem.
  2. Quantização.
3. Imagens coloridas
  1. RGB.
  2. CMY.
  3. YUV.
  4. HLS.
  5. Outros espaços coloridos.
  6. Algumas aplicações com cores.

CÂMERAS



# MODELO DE CÂMERA ESCURA SIMPLES

- Modelo simples porém razoavelmente realista.
- Todos os raios de luz são tratados como raios retilíneos e uniformes.
- Câmeras reais introduzem distorção, porém este modelo pode ser estendido para tratar essas questões.
- Um ponto no espaço 3D real( $x,y,z$ ) pode ser mapeado no plano da imagem ( $i,j$ ) no modelo de câmara escura simples como mostrado a seguir:

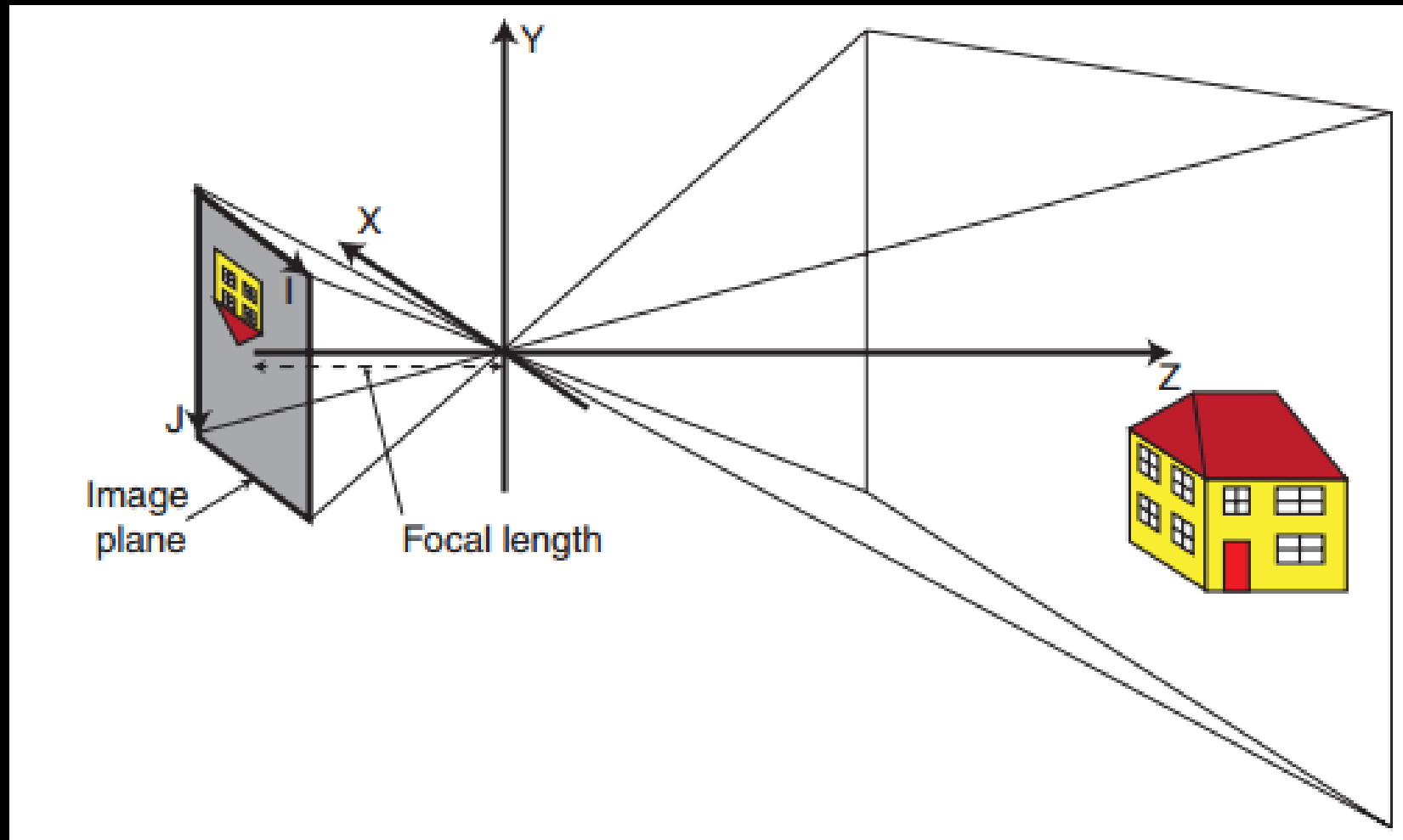
$$\begin{bmatrix} i.w \\ j.w \\ w \end{bmatrix} = \begin{bmatrix} f_i & 0 & c_i \\ 0 & f_j & c_j \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

# MODELO DE CÂMERA ESCURA SIMPLES

- $w$  = fator de escala
- $f_i$  e  $f_j$  = Combinação entre a distância focal e tamanho dos pixels nas direções I e J respectivamente.
- $(c_i, c_j)$  = Coordenadas em que o eixo óptico intercepta o plano da imagem (também conhecido como centro óptico)

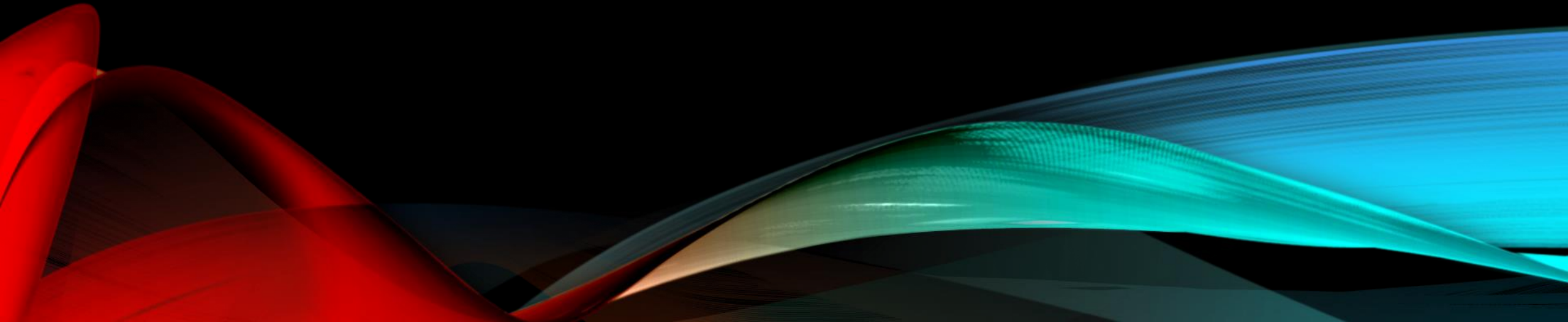
$$\begin{bmatrix} i.w \\ j.w \\ w \end{bmatrix} = \begin{bmatrix} f_i & 0 & c_i \\ 0 & f_j & c_j \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

# MODELO DE CÂMERA ESCURA SIMPLES





IMAGENS



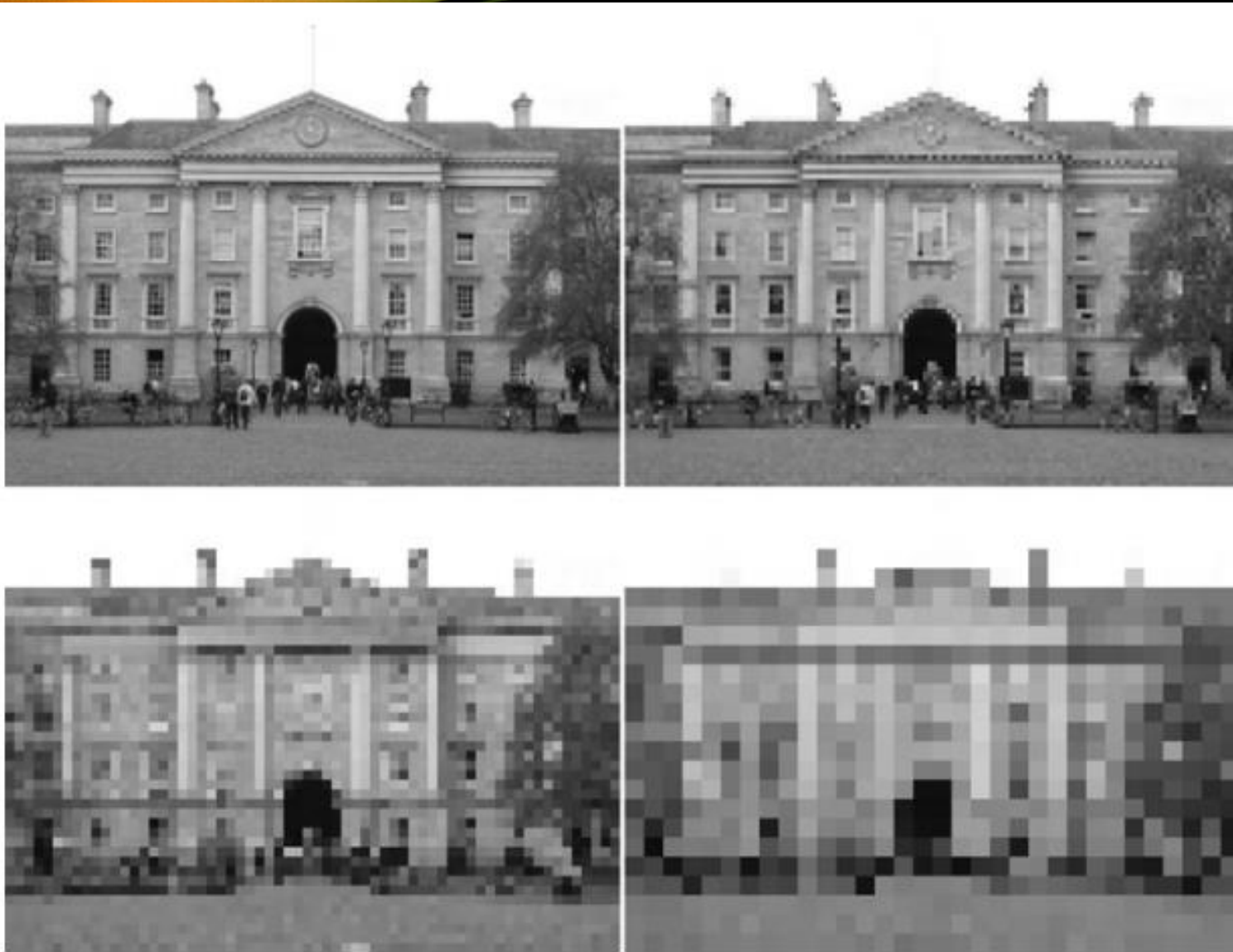
# IMAGENS

- Uma imagem é uma figura (geralmente uma projeção 2D de uma cena 3D) capturada por um sensor.
- É uma função real de dois argumentos contínuos, geralmente expressados como  $(i,j)$  ou (colunas, linhas) ou de forma confusa  $(x,y)$ .
- Contudo temos restrições para representar estas em computadores digitais. São essas restrições:
  - Amostragem – Em uma matriz de (M linhas e N colunas)
  - Quantização – Cada elemento da matriz deve apresentar um valor inteiro



# AMOSTRAGEM

- Imagens digitais são criadas pela amostragem de uma imagem contínua em elementos discretos. Sensores digitais de imagem consistem em um Array 2D de elementos fotossensíveis.
- Cada um destes elementos tem uma região limitada da qual este é estimulado, portanto informação luminosa pode ser perdida.
- Os pixels representam um valor médio de (luminosidade/crominância) em uma área discreta que no mundo real pode ser projetada por um objeto ou ser a soma da projeção de vários objetos.
- Temos de assumir um compromisso com o número de amostras, o número de amostras deve ser o suficiente, pois amostras em excesso significam maior processamento necessário.



**Figure 2.2** Four different samplings of the same image; top left 256x192, top right 128x96, bottom left 64x48 and bottom right 32x24

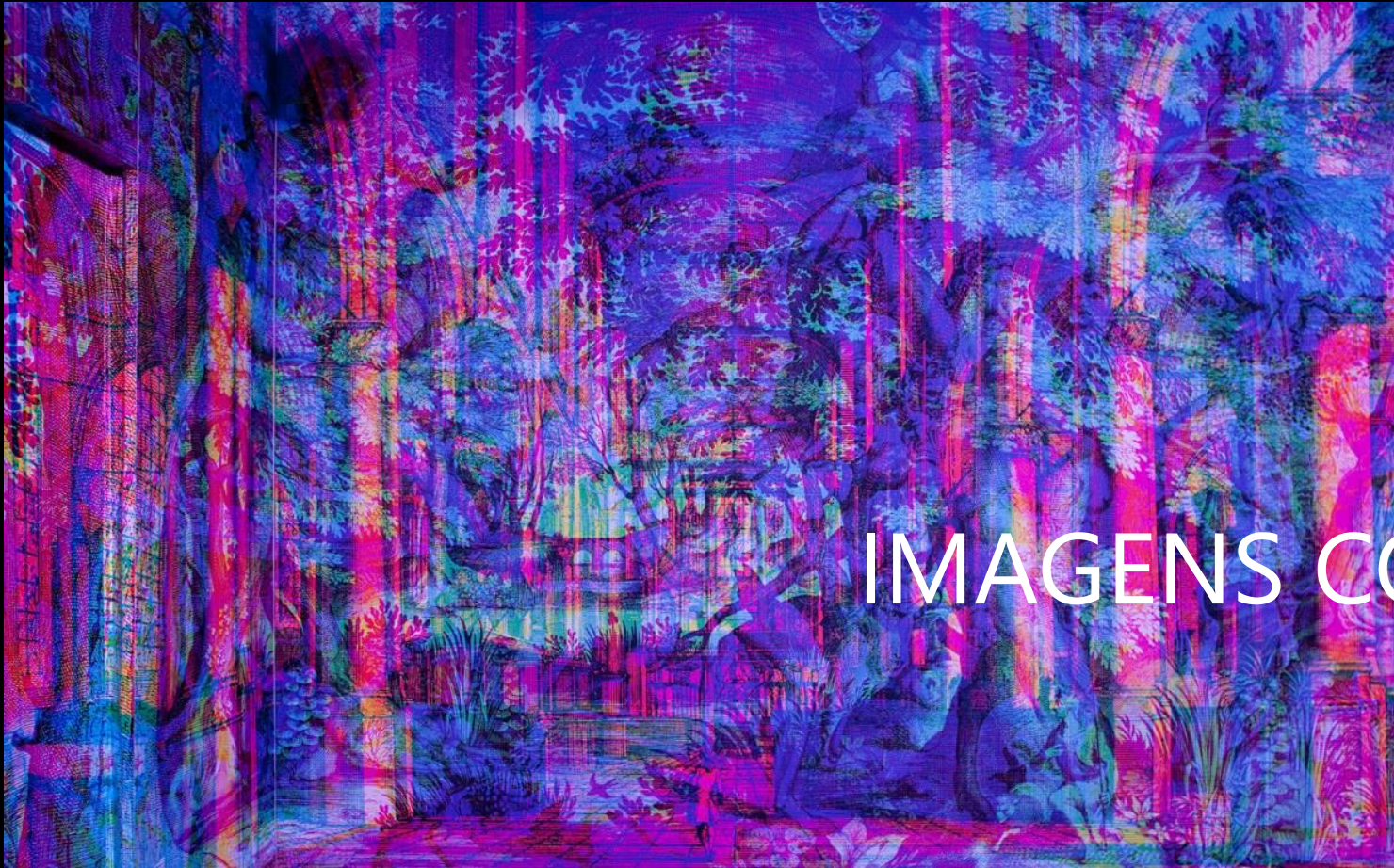
# QUANTIZAÇÃO

- Todo pixel de uma imagem é uma função de brilho da cena. Estes valores são contínuos porém devem ser quantizados em valores discretos. Tipicamente o valor de níveis de brilho por canal é  $k = 2^b$ , onde  $b$  é o número de bits (tipicamente 8).



**Figure 2.3** Four different quantizations of the same grey-scale image; top left 8 bits, top right 6 bits, bottom left 4 bits and bottom right 2 bits





# IMAGENS COLORIDAS



# IMAGENS COLORIDAS

- Imagens Multiespectrais ou coloridas tem múltiplos canais, diferente das imagens em escala de cinza (monocromática) que tem apenas um canal, e são chamadas de forma errôneas de imagens em preto e branco.
- As imagens em escala de cinza representam à Luminância (Y) em cada ponto de uma cena, já as imagens coloridas representam à Luminância e a Crominância (informação de cor) em uma cena.
- Essa informação pode ser representada de inúmeras formas, mas todas as formas envolvem a necessidade de múltiplos canais.
- Portanto as imagens coloridas são maiores e mais complexas que as imagens em escala de cinza. Deve ser definido como se processa cada canal.
- Grande parte das aplicações de processamento de imagens foram desenvolvidas para escala de cinza e sua aplicação em imagens coloridas não é bem definida.



# IMAGENS COLORIDAS

- Por muitos anos a visão computacional foi baseada em imagens de escala de cinza, principalmente pelos dois motivos a seguir:
  - Humanos conseguem compreender imagens em escala de cinza, então porque se preocupar com cor?
  - Imagens em escala de cinza são menores e menos complexas (um valor por ponto).
- Todavia a informação de cor é útil para várias tarefas, como segmentação de uma imagem em objetos fisicamente separáveis (objetos, superfícies e etc.).
- Os humanos são sensíveis a luz com comprimentos de onda entre 400nm e 700nm e os sensores são feitos para serem sensíveis a estes comprimentos.
- Geralmente as imagens coloridas são representadas usando um espaço de 3 dimensões (canais).

# RGB – RED-BLUE-GREEN

- É a representação mais usual de imagens coloridas.
- Vermelho (RED) 700nm
- Verde (GREEN) 546.1nm
- Azul (BLUE) 435.8nm
- Os elementos fotossensíveis das câmeras são espectralmente sensíveis a comprimentos de onda que são centrados nessas três cores.



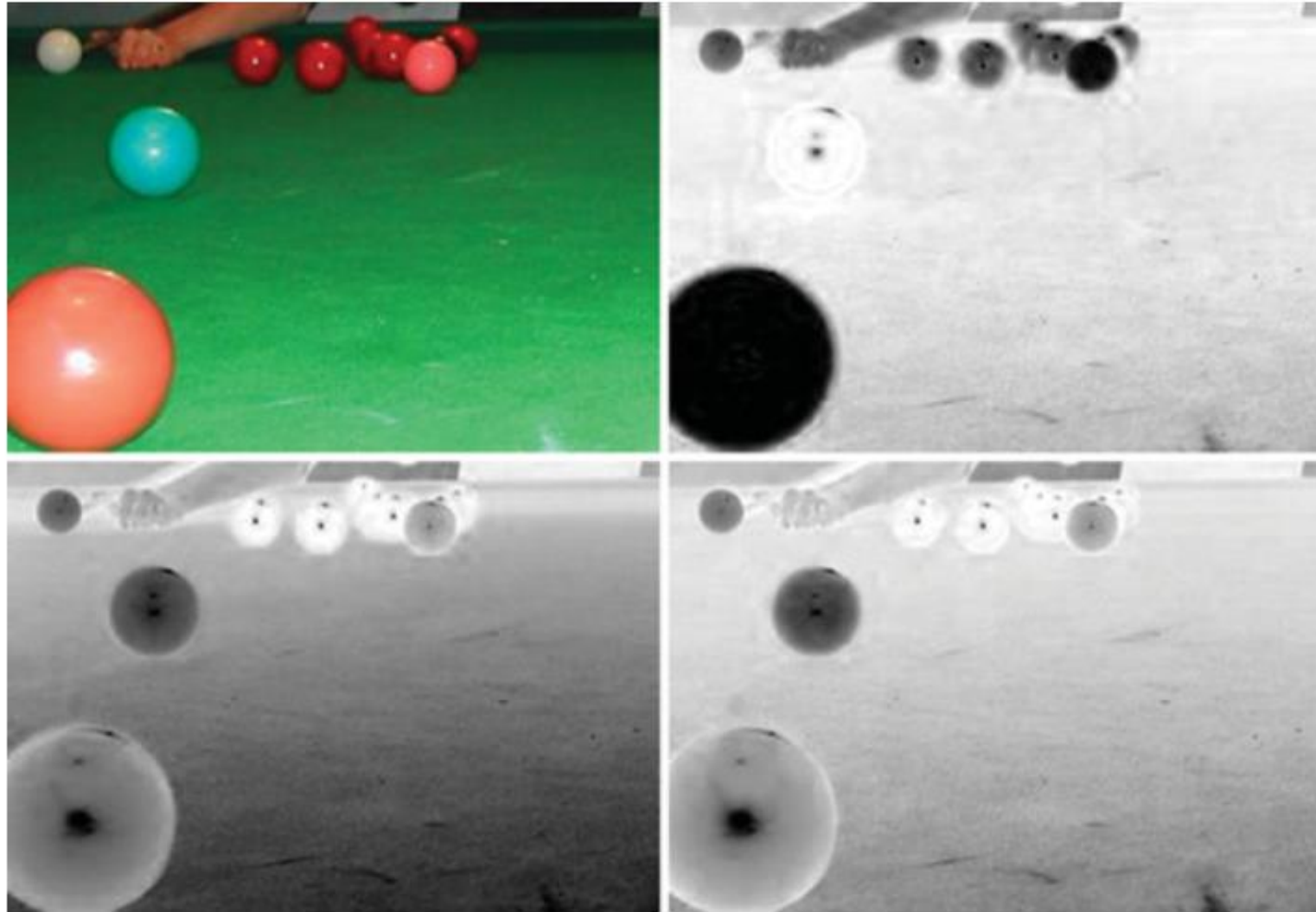
**Figure 2.6** Spectral sensitivity curves for blue (left), green (middle) and red (right) photosensitive elements



**Figure 2.5** RGB Image (top left) shown with red channel (top right), green channel (bottom left) and blue channel (bottom right)

# CMY- CYAN-MAGENTA-YELLOW

- É baseado nas cores secundárias (RGB são as cores primárias), e é um esquema de cores subtrativo, isso quer dizer que os valores de C,M e Y são subtraídos do branco puro para se obter a cor desejada.
- Por esta razão é empregado como padrão de cor em impressoras, onde a superfície de impressão geralmente é branca (CMYK\*)
- Não é suportado diretamente pela OpenCV
- Contudo converter para CMY é trivial tendo em vista que este é o inverso do padrão RGB.
- $C = 255 - R.$
- $M = 255 - G.$
- $Y = 255 - B.$



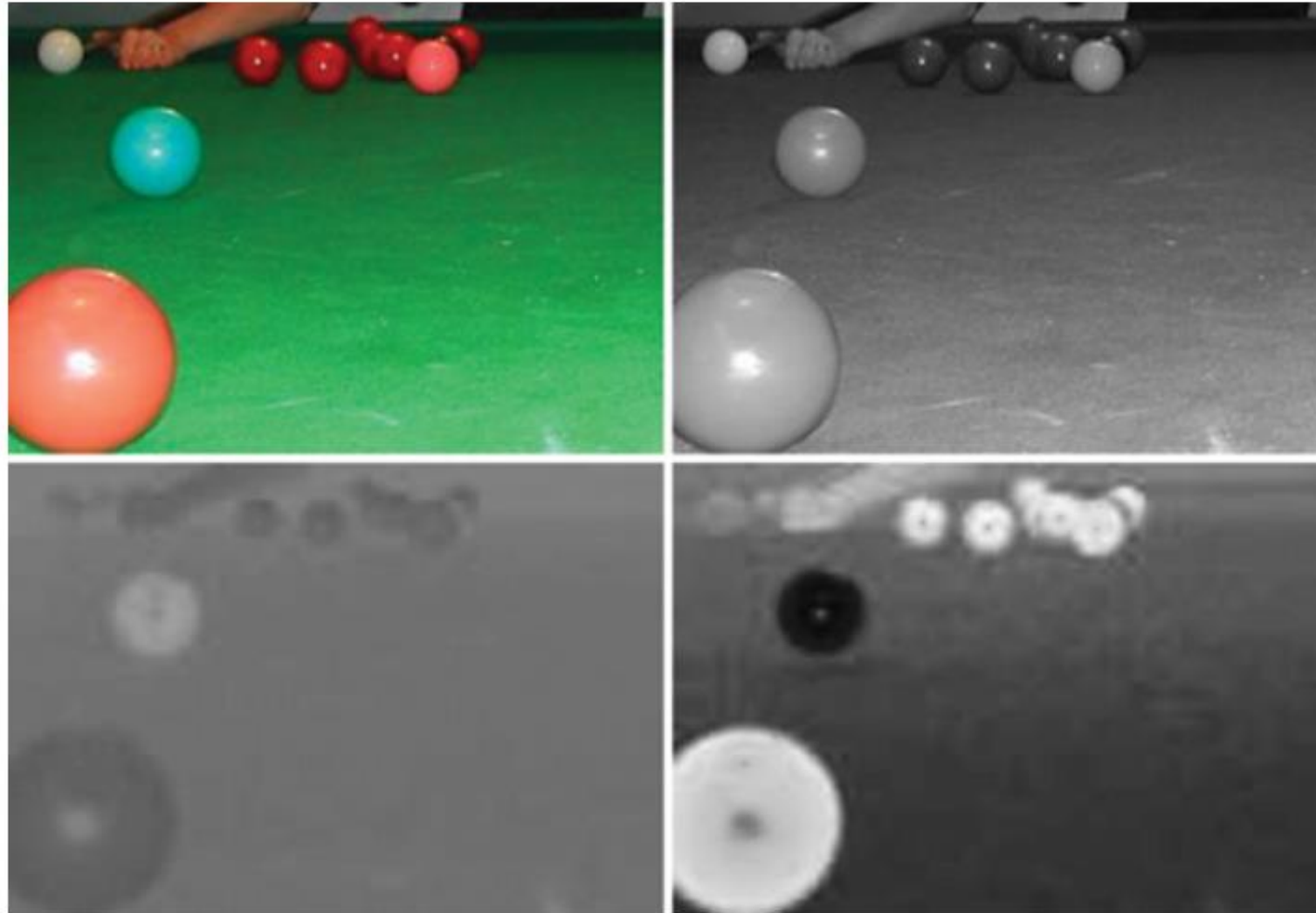
**Figure 2.8** CMY Image (top left) shown with yellow channel (top right), magenta channel (bottom left) and cyan channel (bottom right)



# YUV

- Modelo utilizado para sinais de televisão analógica (PAL,NTSC...) e é formado por Luminância (Y), Azul menos Luminância (U) e Vermelho menos Luminância (V).
- O sistema de visão do ser humano é mais sensível à Luminância do que à Crominância. Este fato é explorado para a codificação de sinais de televisão para reduzir a quantidade de dados a ser transmitida.
- Por exemplo o YUV420p transmite 4 bytes de Luminância (Y) para cada 2 Bytes de Crominância (1 U e 1 V).
- A conversão por sua vez também é simples:
  - $Y = 0.299R + 0.587G + 0.114B$
  - $U = 0.492 * (B - Y)$
  - $V = 0.877 * (R - Y)$

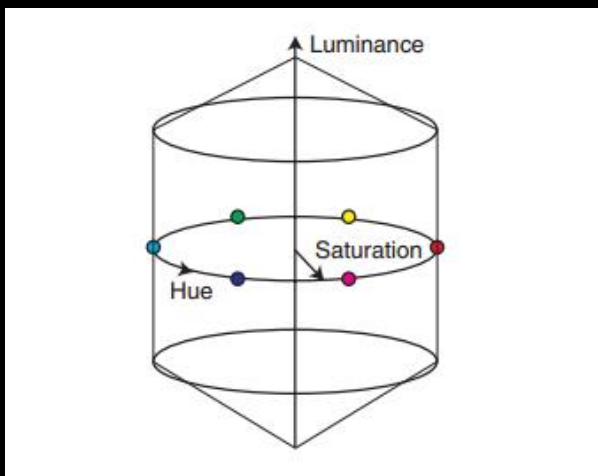


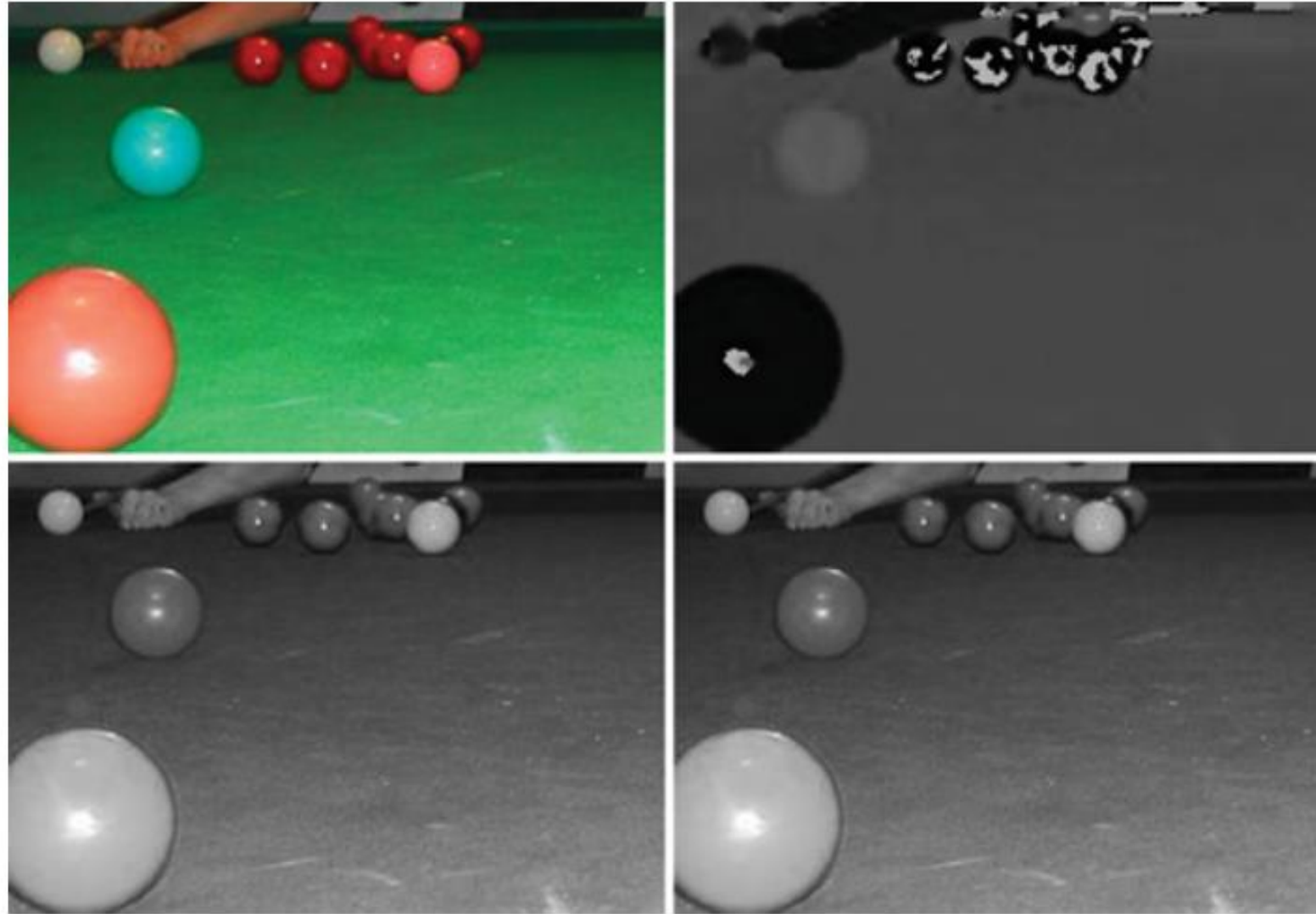


**Figure 2.9** YUV image (top left) shown with luminance (Y) channel (top right), U channel (bottom left) and V channel (bottom right)

# HLS – HUE-LUMINANCE-SATURATION

- O modelo HLS é frequentemente usado na visão computacional pois separa a Luminância e a Crominância (Matiz e Saturação).
- Luminância e Saturação variam de 0 a 1
- A Matiz varia de 0 a 360°
- Tipicamente esses valores são representados de 0 a 255 como os outros modelos, contudo na OpenCV a matiz varia de 0 a 179.





**Figure 2.11** HLS Image (top left) shown with luminance channel (top right), hue channel (bottom left) and saturation channel (bottom right)

## PRATICA 2

- Acrescente à classe image um método que devolva uma imagem em escala de cinza que corresponda à intensidade do canal selecionado pela cena:

```
void extractChannel(cv::Mat& imageIn, cv::Mat& imageOut,int ChannelCode);
```

Os códigos de canal devem ser mapeados como abaixo:

```
enum ChannelTypes {
    ChannelGreen=2,
    ChannelCyan=4,
    ChannelYellow=6,
    ChannelU=8,
    ChannelLuminance=11,
    ChannelSaturation=12,
    ChannelRed=1,
    ChannelBlue=3,
    ChannelMagenta=5,
    ChannelY=7,
    ChannelV=9,
    ChannelHue=10,
};
```

# RECOMENDADO

- Ver mais sobre o modelo HLS e o HSV.
- Olhar a sessão de detecção de pele e de olhos vermelhos presente na sessão de imagens coloridas do livro referência.
- Desenvolver aplicações para implementar as observações lá presentes.