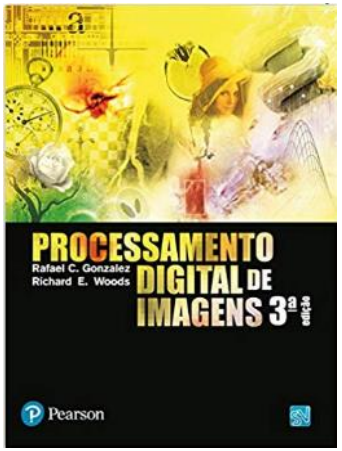
PROCESSAMENTO DE IMAGENS

SISTEMAS DE CORES

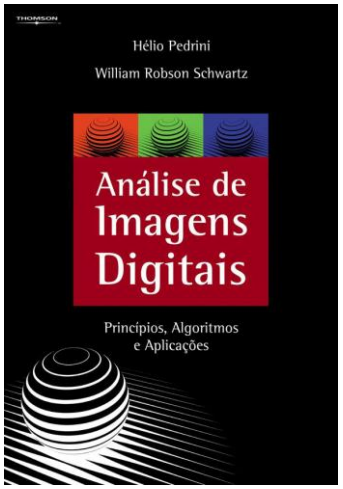
Prof. Msc. Giovanni Lucca França da Silva
E-mail: giovanni-lucca@live.com

SOBRE A DISCIPLINA

- Bibliografia principal:
  - GONZALEZ, Rafael C.; WOODS, Richard C. **Processamento digital de imagens**. Pearson, 2011.



- Bibliografia complementar:
  - PEDRINI, Hélio; SCHWARTZ, William Robson. **Análise de imagens digitais: princípios, algoritmos e aplicações**. Thomson Learning, 2008.



NA AULA PASSADA...

- Transformações geométricas.

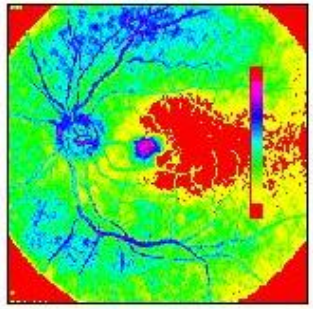
ROTEIRO

- Introdução.
- Fundamentos de cores.
- Modelos de cores.

INTRODUÇÃO

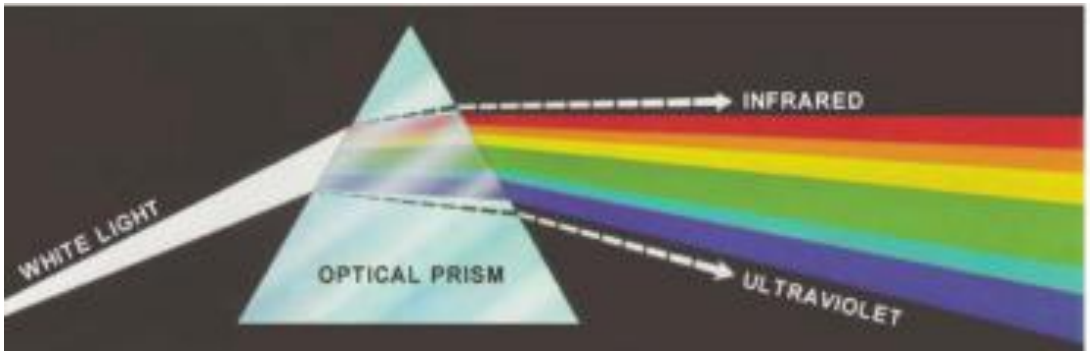
- O uso de cores em PI é motivado por dois fatores principais:
  - A cor é um descritor poderoso para a identificação e extração de objetos.
  - O olho humano pode discernir milhares de tons e intensidades de cores.

- O processamento de imagens coloridas é dividido em duas áreas principais:
  - Processamento de cores reais.
  - Processamento de pseudo-cores.



FUNDAMENTOS DE CORES

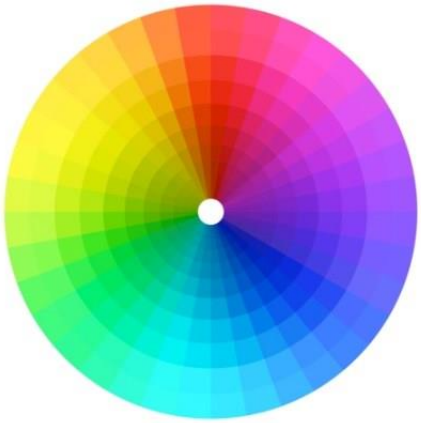
- Em 1666, Isaac Newton descobriu que um feixe de luz solar é decomposta em um espectro contínuo de cores ao passar no prisma.



- As cores que os seres humanos percebem num objeto são determinadas pela natureza da luz refletida do objeto.

FUNDAMENTOS DE CORES

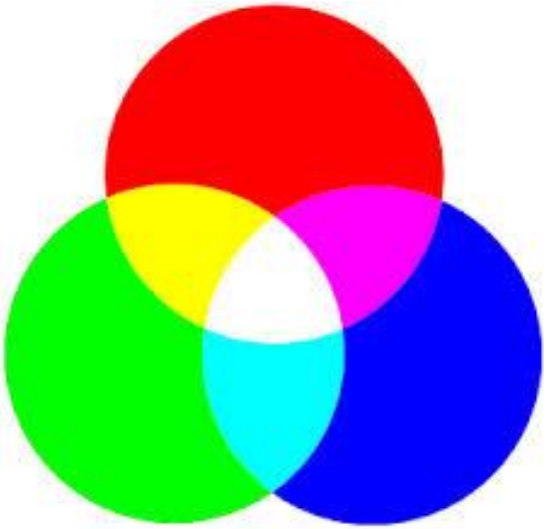
- O que é cor?
  - Propriedade que os corpos têm de absorver ou refletir a luz.
  - Impressão variável que a luz refletida pelos corpos produz no órgão da visão.
  - Sensação produzida pelos diferentes comprimentos de onda atingindo os olhos.



FUNDAMENTOS DE CORES

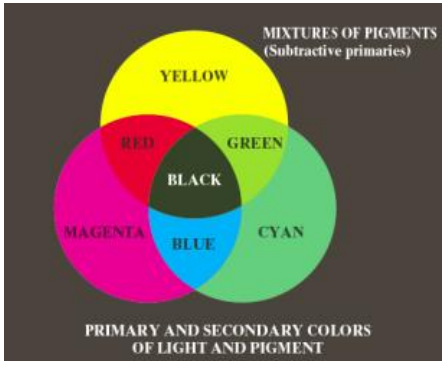
- Devido à estrutura do olho humano, todas as cores são vistas como combinações variáveis das três chamadas cores primárias.
  - Vermelho, verde e azul.

- As cores primárias podem ser adicionadas para produzir as cores secundárias da luz.
  - Magenta (vermelho e azul).
  - Ciano (verde e azul).
  - Amarelo (vermelho e verde).



FUNDAMENTOS DE CORES

- Formação das cores:
  - Processo aditivo: as cores primárias podem ser somadas para produzir as cores secundárias de luz.
    - Misturando as três cores primárias temos o branco.
  - Processo de pigmentação ou coloração: neste processo partículas chamadas pigmentos absorvem ou subtraem uma cor primária da luz e reflete as outras duas.
    - Cores primárias: Magenta, ciano e amarelo.
    - Ex.: Magenta – absorveu verde e refletiu azul e vermelho.



FUNDAMENTOS DE CORES

- As características normalmente usadas para distinguir uma cor da outra são brilho, matiz e saturação.
  - Brilho incorpora a noção cromática da intensidade.
    - Com base em uma escala de preto para branco.
  - Matiz representa a cor dominante percebida pelo observador.
  - Saturação refere-se à pureza relativa ou quantidade de luz branca misturada com um matiz.
    - Ex.: Cor de rosa (vermelho e branco).

- Cromaticidade = matiz + saturação.

FUNDAMENTOS DE CORES

- As quantidades de vermelho, verde e azul necessários para formar qualquer cor são denominadas valores triestímulo e são denotadas X, Y, Z, respectivamente.

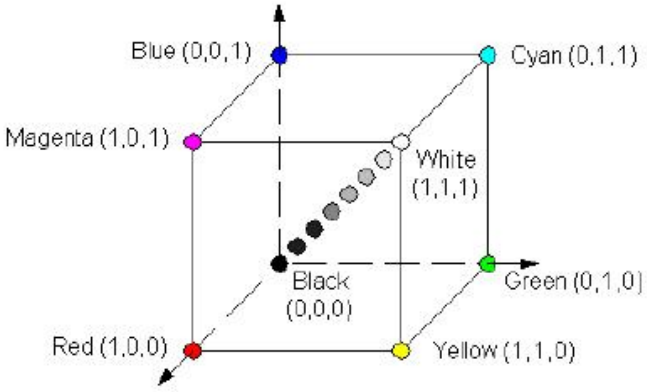
$$x = \frac{X}{X+Y+Z} \qquad y = \frac{Y}{X+Y+Z} \qquad z = \frac{Z}{X+Y+Z}$$

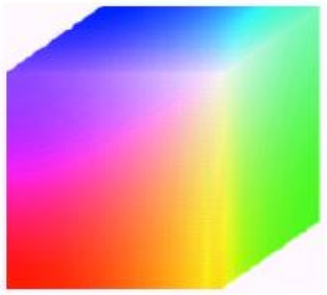
$$x + y + z = 1$$

MODELOS DE CORES

- Facilitar a especificação das cores em alguma forma padrão e de aceite geral.

- É uma especificação de um sistema de coordenadas tridimensionais e um subespaço dentro desse sistema onde cada cor é representada por um único ponto.





MODELOS DE CORES

- A maioria dos modelos de cores atualmente em uso são orientados ou em direção do hardware ou em direção a aplicações envolvendo manipulação de cores.
  - Para hardware:
    - RGB para monitores coloridos e câmeras.
    - CMY para impressoras.
    - YIQ para transmissão de TV colorida.
  - Para aplicações:
    - HSI (matiz, saturação e intensidade).
    - HSV (matiz, saturação e valor).

MODELOS DE CORES

- O OpenCV possui várias conversões de modelos de cores.

- Código.

C++: void **cvtColor**(InputArray **src**, OutputArray **dst**, int **code**, int **dstCn**=0)

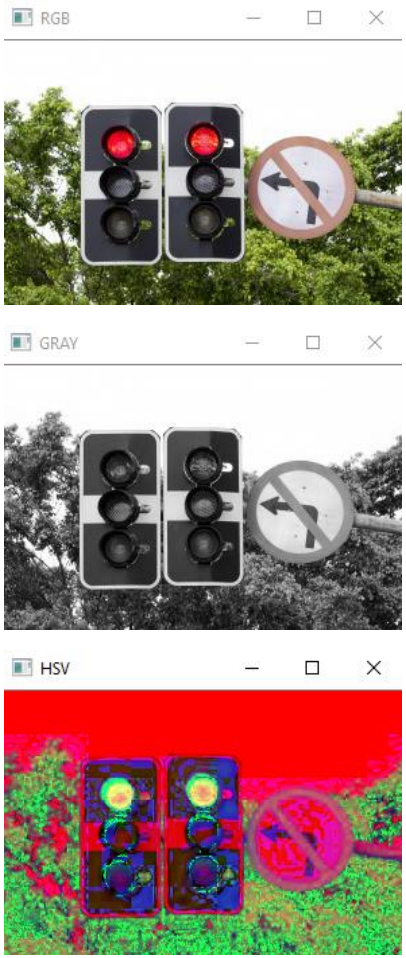
Python: cv2.**cvtColor**(src, code[, dst[, dstCn]]) → dst

  - Exemplos de code:
    - cv2.COLOR_RGB2GRAY
    - cv2.COLOR_RGB2HSV

MODELOS DE CORES

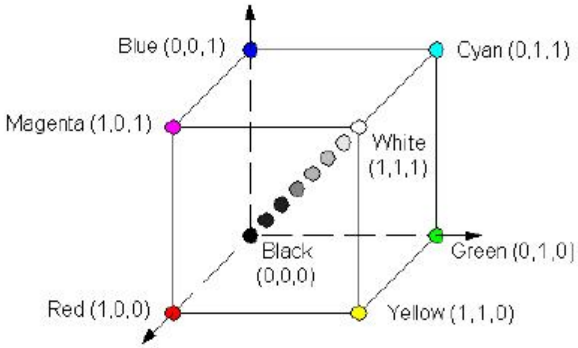
- Exemplo:

```
import cv2

img = cv2.imread("semaforo.png", 1)

imgGRAY = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
imgHSV = cv2.cvtColor(img, cv2.COLOR_RGB2HSV)

cv2.imshow("RGB", img)
cv2.imshow("GRAY", imgGRAY)
cv2.imshow("HSV", imgHSV)
cv2.waitKey()
```



MODELOS DE CORES

- O modelo RGB de cores:
  - Mais utilizado.
  - O branco é a presença de todas as cores.
  - O preto é a ausência de cor.



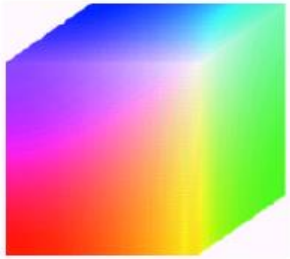


imagem RGB

canal RED

canal GREEN

canal BLUE

MODELOS DE CORES

- O modelo RGB de cores:
  - Exemplo: Separando cada canal da imagem RGB.

```
import cv2

img = cv2.imread("semaforo.png", 1)

channels = cv2.split(img)

cv2.imshow("RGB", img)
cv2.imshow("Blue", channels[0]) # blue channel
cv2.imshow("Green", channels[1]) # green channel
cv2.imshow("Red", channels[2]) # red channel
cv2.waitKey()
```

MODELOS DE CORES

- O modelo RGB de cores:
  - RGB para nível de cinza.
    - Média dos três canais.
    - Y = 0,299R + 0,587G + 0,144B.

```
for(int i = 0; i < width; i++){
    for(int j = 0; j < height; j++){

        ImageTypeRGB::IndexType pixelIndex={{i,j}};
        PixelTypeRGB pixel = image->GetPixel(pixelIndex);

        PixelTypeRGB::ValueType red = pixel.GetRed();
        PixelTypeRGB::ValueType green = pixel.GetGreen();
        PixelTypeRGB::ValueType blue = pixel.GetBlue();

        red = pixel[0];
        green = pixel[1];
        blue = pixel[2];

        float newPixel = ((red + green + blue)/3.0);

        grayLevelImage->SetPixel(pixelIndex, round(newPixel));
    }
}
```

MODELOS DE CORES

- O modelo CMY de cores:
  - Cores secundárias baseadas na distância do branco.
  - Toma como base o sistema subtrativo.
    - C = 255 – R
    - M = 255 – G
    - Y = 255 – B
  - Usado em impressoras.

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix}$$

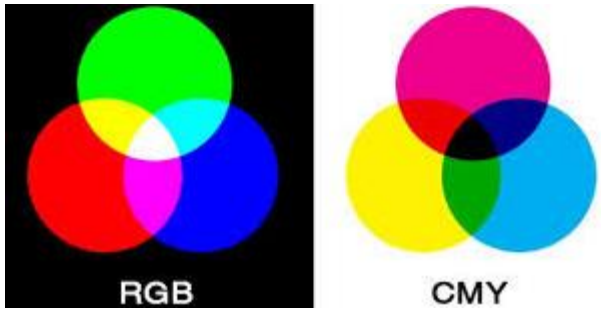
canal CYAN

canal YELLOW

canal MAGENTA

MODELOS DE CORES

- O modelo CMY de cores:
  - RGB para CMY.



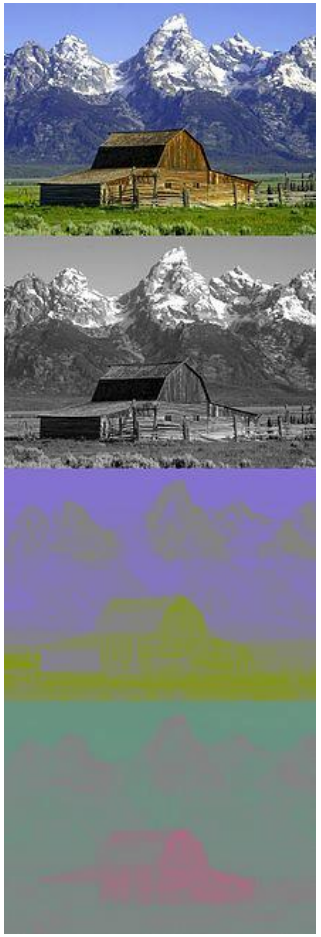

```
for(int i = 0; i < width; i++){
    for(int j = 0; j < height; j++){

        ImageTypeRGB::IndexType pixelIndex={{i,j}};
        PixelTypeRGB pixel = image->GetPixel(pixelIndex);

        PixelTypeRGB::ValueType red = pixel.GetRed();
        PixelTypeRGB::ValueType green = pixel.GetGreen();
        PixelTypeRGB::ValueType blue = pixel.GetBlue();

        red = pixel[0];
        green = pixel[1];
        blue = pixel[2];

        int cyano = 255 - red;
        int magenta = 255 - green;
        int yellow = 255 - blue;

        pixel.SetRed(cyano);
        pixel.SetBlue(magenta);
        pixel.SetGreen(blue);

        changedSystemImage->SetPixel(pixelIndex,pixel);
    }
}
```

MODELOS DE CORES

- O modelo YCrCb de cores:
  - Dedicado ao vídeo analógico.
  - Y = Luminância.
  - Cr = Crominância em vermelho em relação a um valor de referência.
  - Cb = Crominância em azul em relação a um valor de referência.

$$Y \leftarrow 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$$
$$Cr \leftarrow (R - Y) \cdot 0.713 + delta$$
$$Cb \leftarrow (B - Y) \cdot 0.564 + delta$$

$$delta = \begin{cases} 128 & \text{for 8-bit images} \\ 32768 & \text{for 16-bit images} \\ 0.5 & \text{for floating-point images} \end{cases}$$



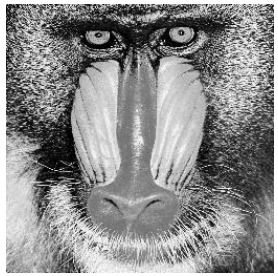
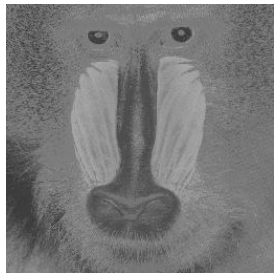
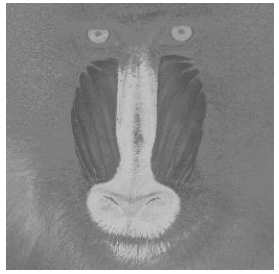
MODELOS DE CORES

- O modelo YUV de cores:
  - Geralmente usado para processamento de vídeos.

$$Y = 0.299\mathrm{R} + 0.587\mathrm{G} + 0.114\mathrm{B}$$

$$U = B - Y$$

$$V = R - Y$$

Y **U** **V**

MODELOS DE CORES

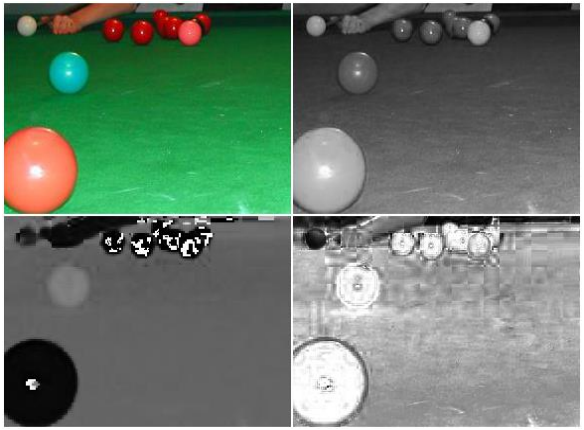
- O modelo YIQ de cores:
  - Luminância (Y), componentes cromáticos: em fase (I) e quadratura (Q).
  - Usado na transmissão comercial de TV colorida.
    - Padrão americano NTSC.
  - A principal vantagem do modelo YIQ é que a luminância (Y) e a informação de cores (I e Q) são desacopladas.

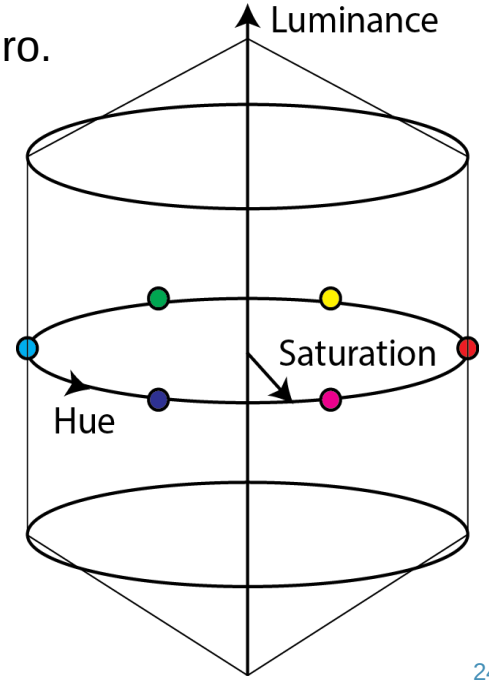
$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.144 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Em que 0 <=R,G,B<=1

MODELOS DE CORES

- Modelo HSL de cores:
  - *Hue* (matiz), *Saturation* (saturação) e *Lightness* (intensidade).
  - Separa o que é cor do que é luminância em um cilindro.
  - Hue [0 , 360].
  - Saturation [0, 1].
  - Lightness [0, 1].





MODELOS DE CORES

- Modelo HSL de cores:
  - RGB para HSL.
    - Normalize os valores de RGB [0.0, 1.0]

$$V_{max} \leftarrow max(R, G, B)$$

$$V_{min} \leftarrow min(R, G, B)$$

$$L \leftarrow \frac{V_{max} + V_{min}}{2}$$

$$S \leftarrow \begin{cases} \frac{V_{max} - V_{min}}{V_{max} + V_{min}} & \text{if } L < 0.5 \\ \frac{V_{max} - V_{min}}{2 - (V_{max} + V_{min})} & \text{if } L \geq 0.5 \end{cases}$$

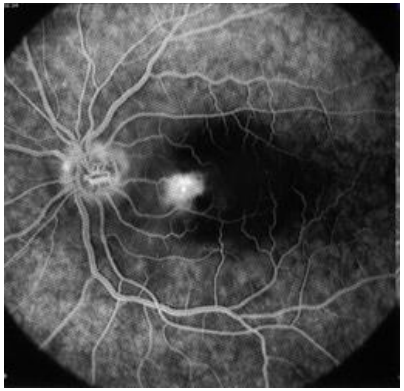
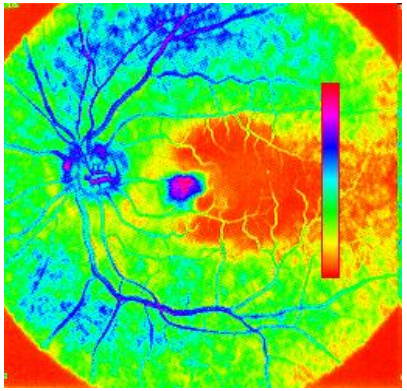
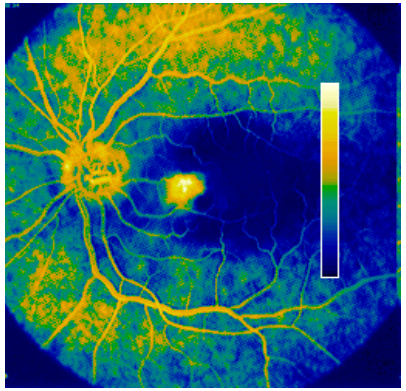
$$H \leftarrow \begin{cases} 60(G - B)/S & \text{if } V_{max} = R \\ 120 + 60(B - R)/S & \text{if } V_{max} = G \\ 240 + 60(R - G)/S & \text{if } V_{max} = B \end{cases}$$

If $H < 0$ then $H \leftarrow H + 360$. On output $0 \leq L \leq 1, 0 \leq S \leq 1, 0 \leq H \leq 360$.

$$V \leftarrow 255V, S \leftarrow 255S, H \leftarrow H/2 (\text{to fit to 0 to 255})$$

MODELOS DE CORES

- Pseudo-cor:
  - Atribuição de um tom de cor para uma intensidade monocromática particular
  - Gray level para RGB.
    - Grupos (cluster) de pixels com determinada cor.

MODELOS DE CORES

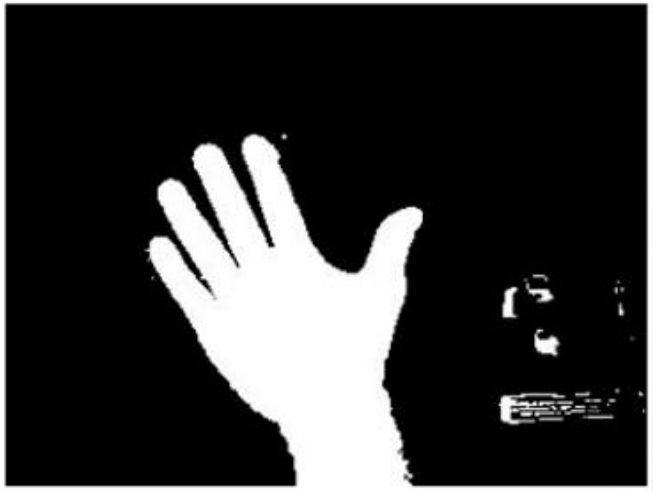
- Outros modelos de cores:
  - HSV.
  - CIE lab.
  - CIE luv.

MODELOS DE CORES

- Aplicações:
  - Detecção de pele com HSL.
    - (S >= 0.2) AND (0.5 < L/S <3.0) AND (H <= 28 OR H >= 330).

MODELOS DE CORES

- Aplicações:
  - Segmentação de pele com YCbCr.
    - $85 \leq Cb \leq 135$ AND $135 \leq Cr \leq 180$ AND $Y \geq 80$.



REFERÊNCIAS

- GONZALEZ, Rafael C.; WOODS, Richard C. **Processamento digital de imagens**. Pearson, 2011.

- PEDRINI, Hélio; SCHWARTZ, William Robson. **Análise de imagens digitais: princípios, algoritmos e aplicações**. Thomson Learning, 2008.

- SILVA, Aristófanes. **Notas de aula da disciplina Processamento de Imagens da Universidade Federal do Maranhão**. 2018.

- BRAZ Jr, Geraldo. **Notas de aula da disciplina Visão Computacional da Universidade Federal do Maranhão**. 2018.