

# Computação Gráfica

## *Sistemas de Cores*

### (Aula 3)

Leonardo Medeiros

Instituto Federal de Alagoas

9 de Abril de 2017

# Roteiro

1 Sistemas de Cores

2 Práticas

3 Bibliografia

# Cores em Computação Gráfica

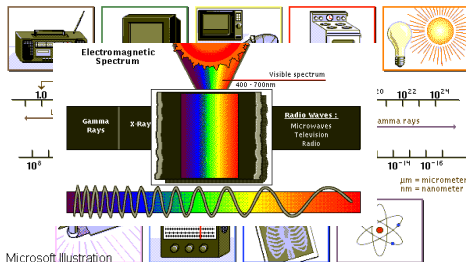
- Uso de cores permite melhorar a legibilidade da informação, possibilita gerar imagens realistas, focar atenção do observador, passar emoções e muito mais.
- Colorimetria – Conjunto de técnicas que permitem definir e comparar cores. Estuda como o olho humano percebe cada cor.
- Qualquer cor pode ser definida por três parâmetros: intensidade, tonalidade cromática e saturação.

## Cores em Computação Gráfica

- Intensidade – indica o grau de intensidade luminosa da superfície examinada, normalmente associada ao brilho ou claridade do material.
- Tonalidade cromática – Caracteriza o comprimento da onda dominante da cor, também chamado de matiz.
- Saturação – mede a pureza da cor, ou seja o quanto ela é saturada em um só tom.

# Processo de formação de cores

- Percepção de cores
  - As diferentes cores, ou espectros luminosos, que podem ser percebidos pelo sistema visual humano correspondem a uma pequena faixa de frequências do espectro eletromagnético, que inclui as ondas de rádio, microondas, os raios infravermelhos e os raios X, como mostrado na figura abaixo:



## Processo de formação de cores

- Percepção visual do mundo baseada nas cores dos objetos
  - Alguns animais só enxergam em preto e branco
  - Outros conseguem ver cores para nós invisíveis
- Conseguimos distinguir algumas dezenas de tons de cinza
- Discernimos vários milhões de cores diferentes
- Só percebemos as cores na presença de luz

## O que são Modelos de Cores ?




- Você precisa de um método para definir cores.
- Os modelos de cores fornecem diversos métodos para definir cores, cada modelo definindo usando componentes de cores específicos.
- Há vários modelos de cores que podem ser escolhidos ao se criar um gráfico.

# Sistemas de cores aditivas

- Usados nos monitores de vídeo e TV.
- A cor é gerada pela mistura de vários comprimentos de onda luminosa, provocando uma sensação de cor quando atinge o olho.
- No processo aditivo, o preto é gerado pela ausência de qualquer cor, indicando que nenhuma luz é transmitida.
- O branco é a mistura de todas elas.

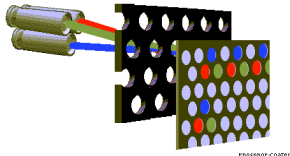


## Modelo de cor RGB

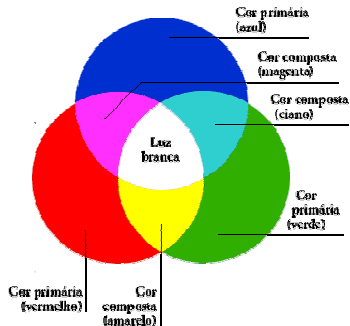
- O modelo de cor RGB define a cor usando os seguintes componentes:
  - vermelho (R) 
  - verde (G) 
  - azul (B) 
- Os componentes R, G e b são as quantidades de luz vermelha, verde e azul que uma cor RGB tem e são medidos em valores que variam de 0 a 255.

## Modelo de cor RGB

- Os monitores usam o modelo de cor RGB. Quando você adiciona luz vermelha, azul e verde juntas, de forma que o valor de cada componente seja 255, aparece a cor branca.
- Quando o valor de cada componente é 0, o resultado é preto puro.

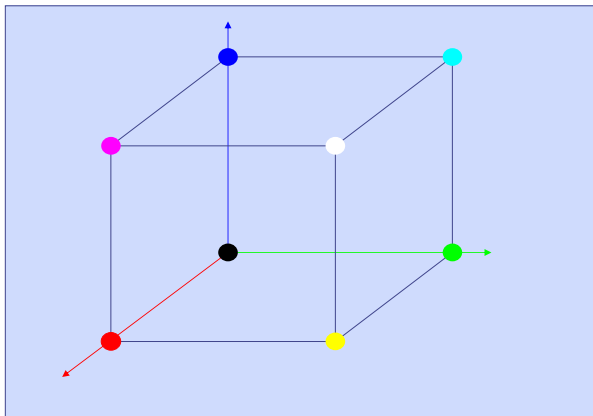


# Modelo de cor RGB

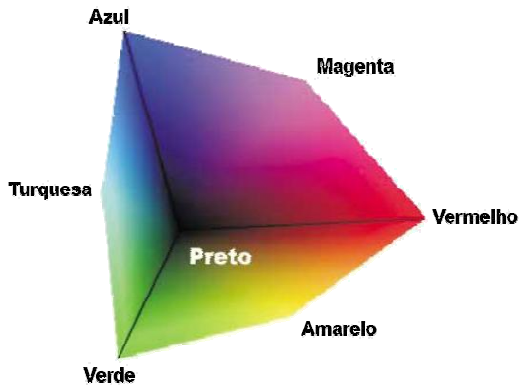


- A representação da cor  $C$  de cada pixel de uma imagem pode ser obtida matematicamente por:
  - $C = r.R + g.G + b.B$

## O cubo RGB



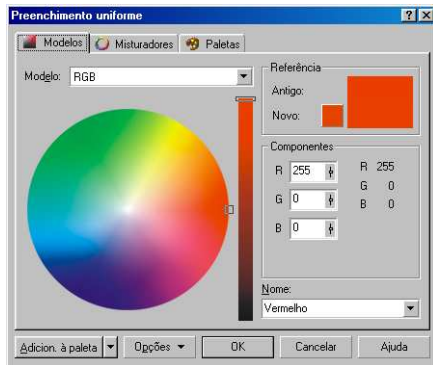
## O cubo RGB



# Canais RGB



# Canais RGB







## Sistemas de cores subtrativas

- Usados nas impressões e pinturas. Possui como cores primárias o Azul Ciano, o Magenta e Amarelo (CMY).
- No processo subtrativo, cores da luz branca são absorvidas.
- A luz branca ao atingir um objeto tem parte absorvida e parte refletida.
- O branco corresponde a ausência de qualquer cor e o preto a mistura de todas.

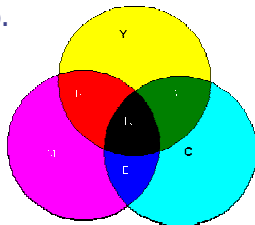


## Modelo de cor CMYK

- O modelo de cor CMYK define a cor usando os seguintes componentes:
  - ciano (C) 
  - magenta (M) 
  - amarelo (Y) 
  - preto (K) 
- Os componentes C M Y K são quantidades de tinta ciano, magenta, amarelo e preto que uma cor CMYK contém e são medidos em porcentagem de 0 a 100.

## Modelo de cor CMYK

- Os materiais impressos são reproduzidos usando o modelo de cor CMYK.
- Ao combinar ciano, magenta, amarelo e preto, de forma que o valor de cada componente seja 100, o resultado é preto.
- Quando o valor de cada componente é 0, o resultado é branco puro.



# Modelo de cor CMYK



# Combinações de cores

Aditivas

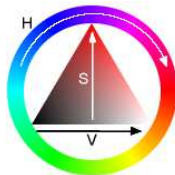


Subtrativas



## Modelo de cor HSV (HSB)

- Um modelo de cor que define três componentes: matiz (H), saturação (S) e brilho (V - Value).
- O matiz determina a cor ou tonalidade (amarelo, laranja, vermelho, etc.);
- O brilho determina a intensidade percebida (cor mais clara ou mais escura);
- A saturação determina a profundidade ou “pureza” da cor (de esmaecida a intensa).

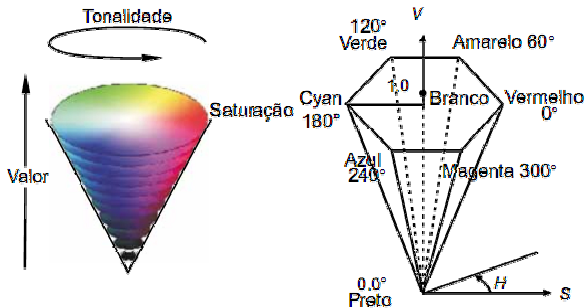


## Modelo de cor HSV

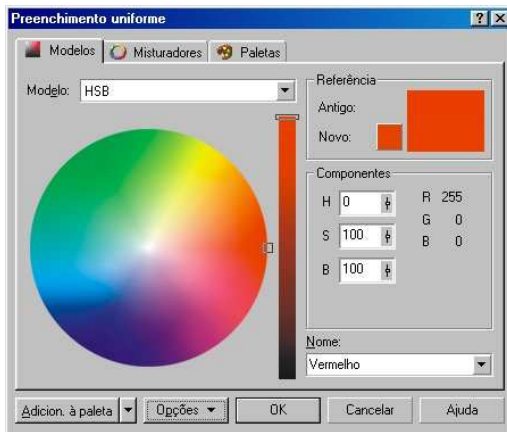
- O matiz descreve o pigmento de uma cor e é medido em graus de 0 a 359.
  - Por exemplo: 0 grau é vermelho; 60 graus, amarelo; 120 graus, verde; 180 graus, ciano; 240 graus, azul e 300 graus, magenta.
  - A saturação descreve a vivacidade ou o esmaecimento de uma cor e é medida em porcentagem de 0 a 100 (quanto maior a porcentagem, maior a vivacidade da cor).
  - O brilho descreve a quantidade de branco que uma cor contém e é medido em porcentagem de 0 a 100 (quanto maior a porcentagem, maior o brilho da cor).

## Modelo de cor HSV

- Por utilizar um sistema de cores que são mais intuitivas do que combinações de cores primárias, é mais adequada para ser usada na especificação de cores em nível de interface com o usuário.



# Modelo de cor HSV





# HSV Color Space

**HSV** (Hue, Saturation & Value/Brightness) is a color space that attempts to represent colors the way humans perceive it. It stores color information in a cylindrical representation of RGB color points.

**Hue** – Color Value (0 – 179)

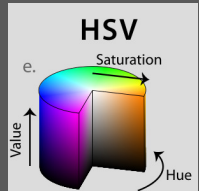
**Saturation** – Vibrancy of color (0-255)

**Value** – Brightness or intensity (0-255)

It's useful in computer vision for color segmentation. In RGB, filtering specific colors isn't easy, however, HSV makes it much easier to set color ranges to filter specific colors as we perceive them.

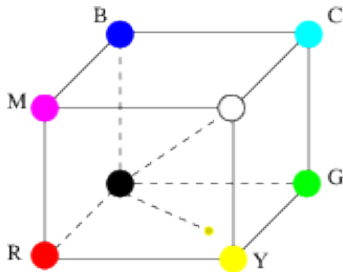
Visit these links to learn more:

- [wikipedia.org/wiki/HSL\\_and\\_HSV](http://wikipedia.org/wiki/HSL_and_HSV)
- [http://coecsl.ece.illinois.edu/ge423/spring05/group8/FinalProject/HSV\\_writeup.pdf](http://coecsl.ece.illinois.edu/ge423/spring05/group8/FinalProject/HSV_writeup.pdf)

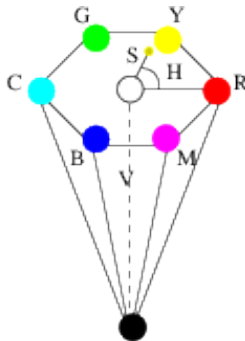


# Espaços de cor

- Uma imagem colorida pode ser armazenada em diferentes espaços de cor (RGB, YCbCr, CMY, HSV, Lab, etc.).
- A conversão da imagem de um espaço de cor para outro é uma operação matemática pixel a pixel (e.g., multiplicação matricial envolvendo  $\vec{I}(p)$ ).



RGB and CMY



HSV

# Baixar Imagem no SIGAA

Antes de prosseguirmos, baixem a imagem 'babuino2.jpg' do SIGAA e coloquem no subdiretório (**img** superior ao seu código fonte).

# Convertendo a Imagem Para a YCR

```
import cv2
import numpy as np
image = cv2.imread('../img/babuino2.jpg')
cv2.imshow("Original", image)

image_cy = cv2.cvtColor(image, cv2.COLOR_BGR2YCR_CB)
cv2.imshow("Converted Image in New Color SPACE YCR",
           image_cy)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

# Exemplo Prático de HSV - Track da Cor Azul

Sabendo o HSV é um sistema de cores voltado para identificar mais facilmente range de cores, veja este código que identifica objetos das cores azul.

## Exemplo Prático de HSV - Track da Cor Azul

```
import cv2
import numpy as np
cap = cv2.VideoCapture(0)
while(1):
    _, frame = cap.read()
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    lower_blue = np.array([110,50,50])
    upper_blue = np.array([130,255,255])
    mask = cv2.inRange(hsv, lower_blue, upper_blue)
    res = cv2.bitwise_and(frame,frame, mask= mask)
    cv2.imshow('res',res)

# Press ESC to close windows
k = cv2.waitKey(5) & 0xFF
if k == 27:
    break
cv2.destroyAllWindows
```

# Exercício

## Altere o Código

Altere o código para identificar uma bola de tênis :D.

Para identificar os valores BGR que você precisa. Por exemplo, tentar encontrar o Range de HSV para o valor Verde, tente os seguintes comandos no terminal do Python:

```
>>> green = np.uint8([[[0,255,0 ]]])  
>>> hsvgreen = cv2.cvtColor(green,cv2.COLOR_BGR2HSV)  
print(hsvgreen)
```

Agora utilize  $[H-10, 100, 100]$  e  $[H+10, 255, 255]$  como *lower bound* e *upper bound* respectivamente.

# Bibliografia

- Solomon C., Breckon T. **Fundamentals of Digital Image Processing.. A Practical Approach with Examples in Matlab** (Wiley-Blackwell, 2011)(ISBN 9780470844724)(en)(355s)
- A. Conci, E. Azevedo e F.R. Leta - **Computação Gráfica: volume 2 (Processamento e Análise de Imagens Digitais)**, Campus/Elsevier. 2008 - ISBN 85-352-1253-3.
- Open CV Tutorials
- Tutorial de ColorSpace [http://docs.opencv.org/trunk/df/d9d/tutorial\\_py\\_colorspaces.html](http://docs.opencv.org/trunk/df/d9d/tutorial_py_colorspaces.html)



DÚVIDAS ?