

Sistemas de Visão e Percepção Industrial

6-Complementos sobre Percepção e Representação

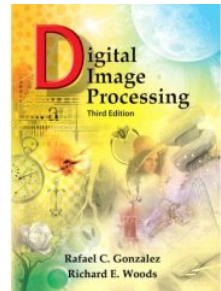
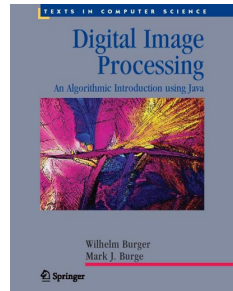
Percepção com Cor

Sumário

- 1 Introdução e origem da cor
- 2 Percepção humana da cor
- 3 Espaços de cor
- 4 Conversão entre espaços de cor
- 5 Exemplos de aplicação

Referências

- Burger, Cap. 12
- Gonzalez, Cap. 6



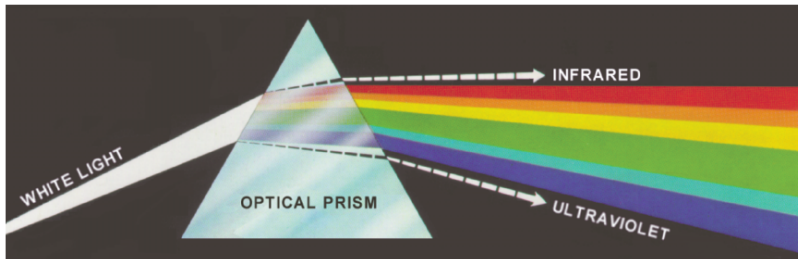
Introdução e origem da cor

Relevância e aplicações da percepção da cor

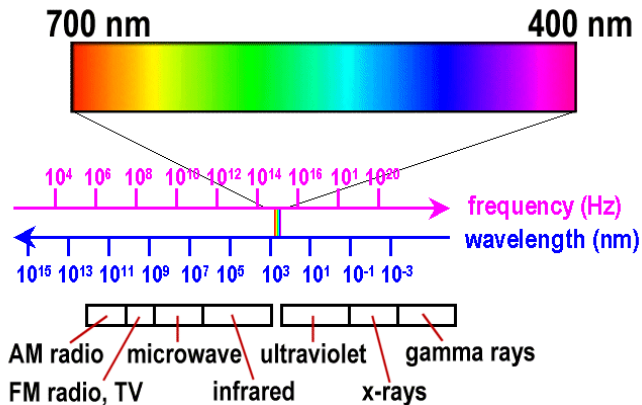
- A informação de cor é muito rica e permite sistemas de percepção mais sofisticados e robustos, desde que asseguradas as condições de iluminação!
- Industrialmente, a visão com cor aplica-se para resolver três grandes categorias de aplicações:
 - Verificação de cor
 - Verificar se a cor de um determinado componente coincide com a que o sistema de visão está programado para encontrar.
 - Ordenação por cor
 - Ordenar componentes em função da sua cor.
 - Inspeção de cor
 - Inspecionar componentes coloridos procurando defeitos que as técnicas baseadas em níveis de cinzento não conseguem detetar.

Origem da Cor

- Associada, sobretudo, ao comprimento de onda da radiação electromagnética.
- Primeiros estudos formais por Newton (1672)



As cores no espectro visível

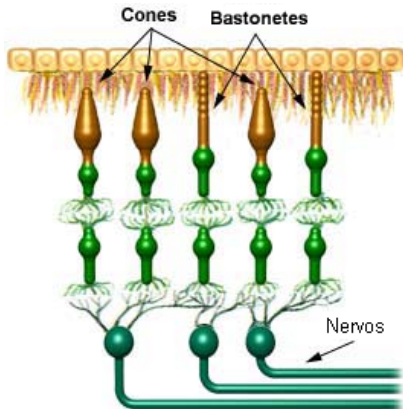


color	wavelength interval	frequency interval
red	~ 700-635 nm	~ 430-480 THz
orange	~ 635-590 nm	~ 480-510 THz
yellow	~ 590-560 nm	~ 510-540 THz
green	~ 560-490 nm	~ 540-610 THz
blue	~ 490-450 nm	~ 610-670 THz
violet	~ 450-400 nm	~ 670-750 THz

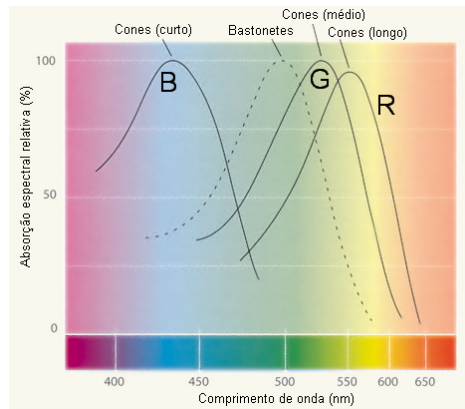
Percepção humana da cor

Percepção da cor pelo olho humano

- O olho humano possui dois tipos de células sensíveis à luz:
 - Cones - de três tipos e sensíveis à cor.
 - Bastonetes – detetam contrastes.

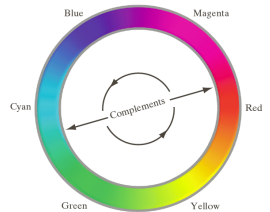
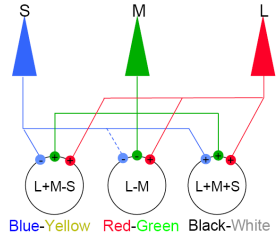


- Os cones apresentam sensibilidades diferentes ao comprimento de onda da radiação incidente:
 - R(ed), G(reen), B(lue)



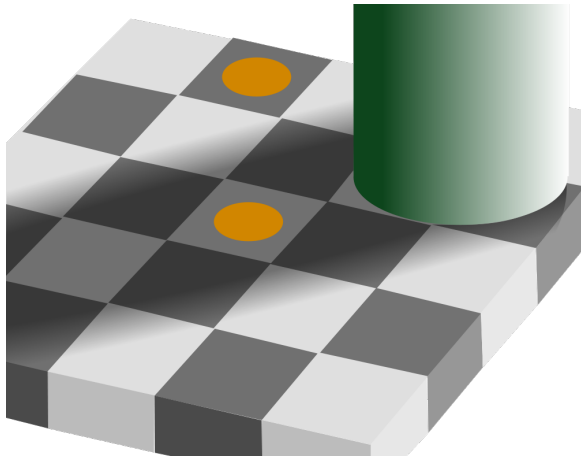
Teoria das cores opostas no olho humano

- Os sinais dos 3 tipos de cones (BGR ou SML, se se pensar nos comprimentos de onda – *Short, Medium, Long*) são recodificados em componentes antagónicas (ditas opostas):
 - Vermelho vs. verde
 - Amarelo vs. azul
 - Intensidade geral da luz: claro vs. escuro
- São estes três sinais recodificados que são enviados ao cérebro
 - Esta recodificação é robusta e eficiente



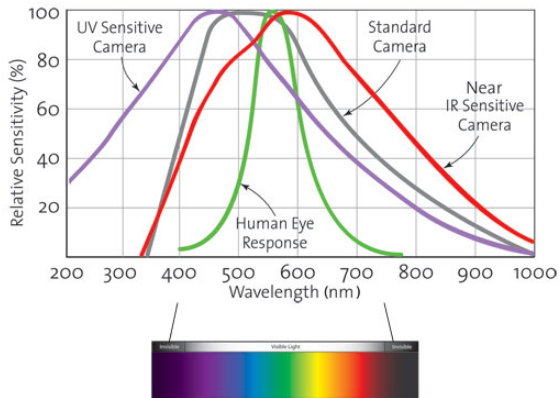
Subjetividade da percepção humana

- A percepção da cor está associada a um conjunto de fenômenos e circunstâncias que podem ser subjetivos.
- Por exemplo, na figura as cores não são o que parecem.
- Cores iguais podem ter interpretações diferentes pelo cérebro humano:
 - Os discos amarelos são ambos: RGB(209,134,0)
 - As quadrículas que os contêm são ambos: RGB(112,112,112)



A gama da percepção humana

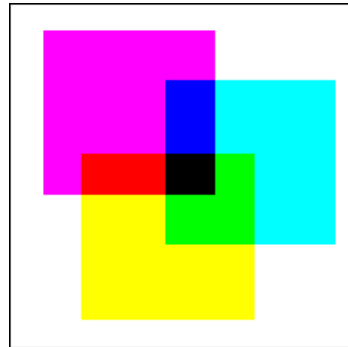
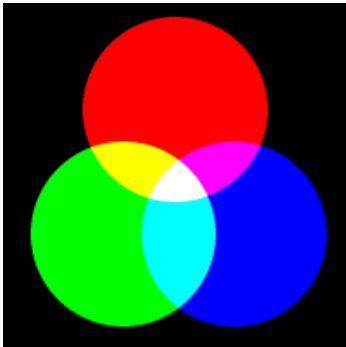
- A sensibilidade do olho humano é diferente da maioria das câmaras modernas.
- A grande sensibilidade das câmaras no infravermelho é bem conhecida:
 - Nalguns casos até se usam filtros para limitar a entrada de luz nessa gama que já é invisível para o olho humano.



Espaços de cor

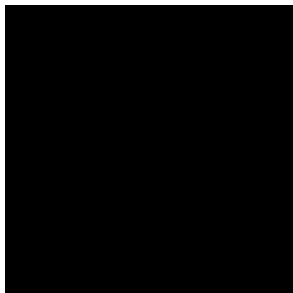
Formação das cores do espectro por combinação de cores

- O processo de formação de cores por "mistura" de cores ditas primárias tem base biológica e relaciona-se com a sensibilidade do olho às diversas frequências do espectro.
- Há dois sistemas mais comuns para a formação de cores por "mistura" de cores primárias:
 - O sistema "RGB" que é dito "aditivo"
 - O sistema "CMYK" que é dito "subtrativo"

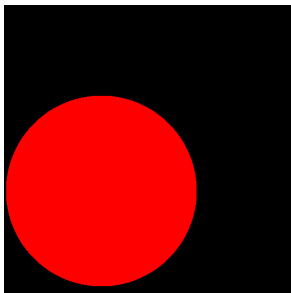


Formação das cores do espectro – Aditivo

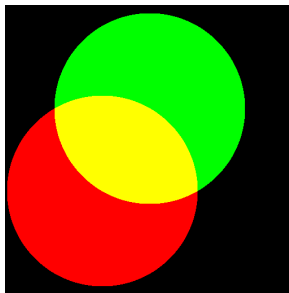
- O sistema "RGB" começa com preto e depois adiciona proporções de R, G e B.
 - Sistema aditivo que começa em "preto" (ausência de luz) e vai mudando a cor até chegar ao branco quando se adicionam as proporções máximas de R, G e B.
 - Intensidades variáveis das cores primárias resultariam noutras cores além destas "puras"



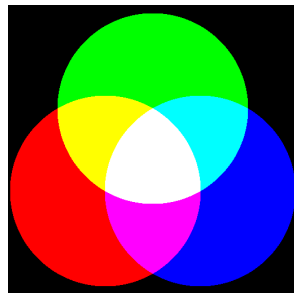
Preto



Vermelho



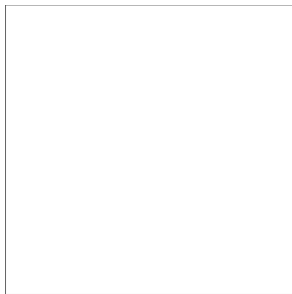
Verde



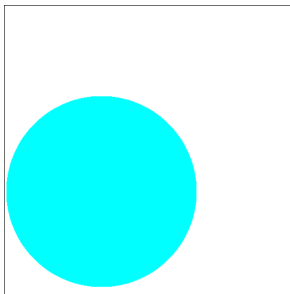
Azul

Formação das cores do espectro – Subtrativo

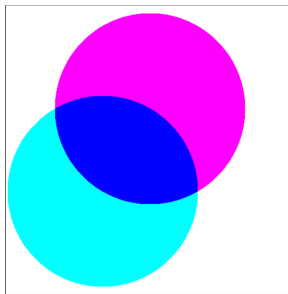
- O sistema "CMYK" começa com branco e "subtrai" partes de Cian, Magenta e Amarelo.
 - A cor branca inicial reflete toda a luz e a adição sucessiva das cores C, M e Y vai impondo a absorção de partes do que era refletido.
 - Em teoria, as contribuições máximas de C, M e Y (por absorção) resultam em preto.
 - Nas impressoras reais isso não é feito por imperfeição das tintas, e para economia de tinteiros;
 - Por isso, usa-se uma quarta cor de tinta para lidar com o preto (K).



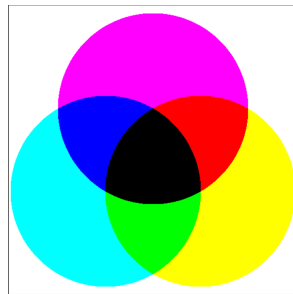
Branco



Cian



Magenta

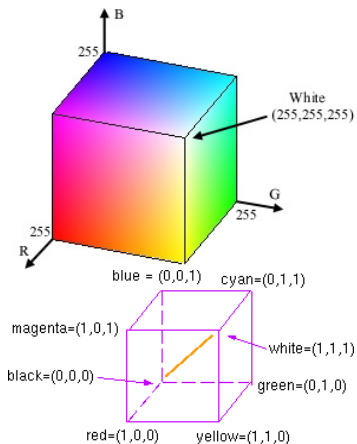


Amarelo

- RGB (Red, Green, Blue)
 - sistema base do olho humano
 - usado em monitores e televisores – sistemas emissores de luz (ou aditivos)
- CMYK (Cyan, Magenta, Yellow, Black)
 - para impressão e impressoras – sistemas absorvedores de luz (ou subtractivos)
- HSV (Hue (tom), Saturation (saturação), Value)
 - Adequado para segmentação de cor (segue a sequência do espectro eletromagnético)
- YCbCr (Em video Digital)
 - Luminância, Crominância (2 sinais):
 - Cb = (componente azul – valor de referência)
 - Cr = (componente vermelho – valor de referência)
- NTSC (Luminância, tom (Hue), Saturação)
 - sistema USA para video digital
- YUV Luminância, (blue-yellow), (red-green)
 - PAL – video
- CIE (Commission Internationale d'Éclairage)
 - Intensidade (O "Value" do HSV) + 2 coordenadas ortogonais para cor
 - Tem diversos standards, entre eles o conhecido $L-\alpha\beta$, ou Lab ou CIELAB

Representação dos espaços de cor

- Os espaços de cor são representados normalmente por entidades tridimensionais dado que são 3 as componentes em qualquer um deles.
- Exemplo RGB – cubo de cores

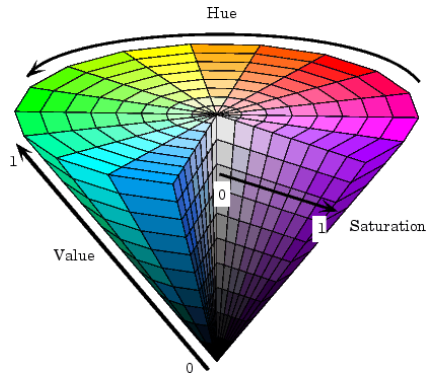


Conversão de RGB para cinzentos

- Na conversão de RGB para níveis de cinzento...
 - ... poder-se-ia pensar numa média igual das três componentes...
 - ... mas o olho humano é mais sensível ao verde.
 - A proporção aproximada é:
 - $I = 0.30 \times R + 0.59 \times G + 0.11 \times B$
 - Em Matlab pode-se usar a função:
 - `rgb2gray()`

O modelo HSV (ou HSI)

- H(ue)
 - Tom, tez, ou matiz
 - Exprime-se num valor angular ($0 - 360^\circ$) [mas em Matlab vai de 0 a 1]
 - Posição no arco-iris (circular)
- S(aturação)
 - "pureza" da cor
 - Qualquer cor pouco "pura" aproxima-se de cinza
 - Varia de 0 a 1
 - Mais ou menos componentes espectrais simultâneas.
- Valor (intensidade)
 - Varia de 0 a 1
 - Traduz "uma intensidade associada ao brilho"
 - Quantidade de luz ("número de fótons")



Conversão entre espaços de cor

Conversão entre RGB e HSV

- Os valores RGB aparecem em duas variantes principais:
 - Intervalos $[0, 255]$ ou $[0, 1]$
- Os valores de HSV aparecem nas seguintes gamas:
 - H em graus $[0, 360]$
 - 0-60 vermelhos
 - 60-120 amarelos
 - 120-180 verdes
 - 180-240 cians
 - 240-300 azuis
 - 300-360 magentas
 - S e V na gama $[0, 1]$

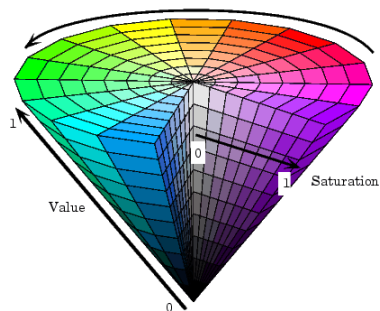
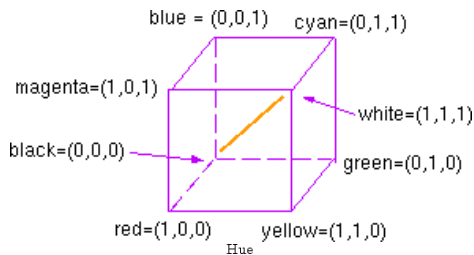
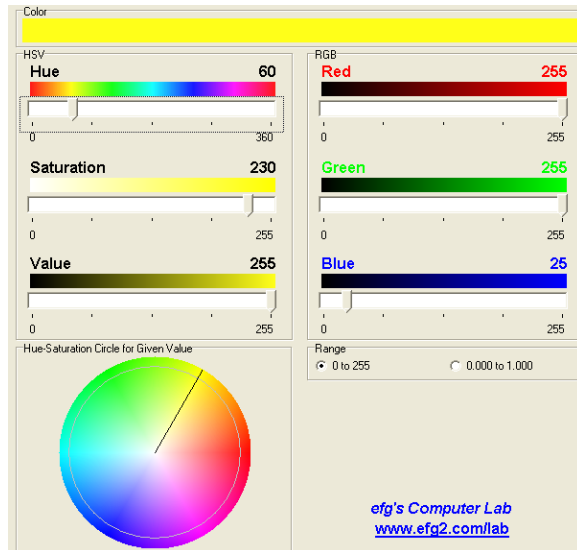


Ilustração da conversão RGB-HSV

- Comando HSV-RGB.exe
- Destaques:
 - Saturação a 0
 - RGB iguais
 - Os cinzentos
 - Variações de um dos R G ou B
 - Verificar as variações não lineares de HSV



Conversão de RGB para HSV

- V – o maior dos valores de RGB
 - $V = \max(R, G, B)$
- S – Gama total de valores RGB em relação a V
 - $S = [\max(R, G, B) - \min(R, G, B)] / \max(R, G, B)$
 - $S = [V - \min(R, G, B)] / V$
 - Mas se $V == 0$ então $S = 0$
- H – depende de RGB do seguinte modo:
 - Se $R == G == B$, H é indefinido (não há cor) — na prática atribui-se 0!
 - Caso contrário, o processo depende de $\max(R, G, B)$
- A literatura dá mais detalhes:
 - <https://cs.stackexchange.com/questions/64549/convert-hsv-to-rgb-colors>
 - Etc.
- O Matlab dispõe da função `rgb2hsv()`

Conversão de HSV para RGB

- Processo em geral menos imediato que de RGB para HSV
- Se $S = 0$ então $R = G = B = V$
- Caso contrário:
 - Obtém-se a cor geral: $C = H/60$
 - Obtém-se o termo $F = (H - 60 \times C)/60$
 - Obter
 - $P = V(1 - S)$
 - $Q = V(1 - S \times F)$
 - $T = V(1 - S(1 - F))$
 - Finalmente
 - $R = V + P$
 - $G = V + Q$
 - $B = V + T$
- Em Matlab há a função:
 - `hsv2rgb()`

- No modelo YUV (sistema PAL de video) há a luminância e duas componentes de cor resultantes da diferença de componentes primárias
- O Matlab não tem funções específicas para a conversão, mas a seguinte relação é comum:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.30 & 0.59 & 0.11 \\ -0.17 & -0.33 & 0.50 \\ 0.50 & -0.42 & -0.08 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Conversão RGB-YIQ (NTSC)

- Também neste modelo há a luminância e duas componentes de cor resultantes da diferença de componentes
- Em Matlab há as funções:
 - `ntsc2rgb()` e `rgb2ntsc()`

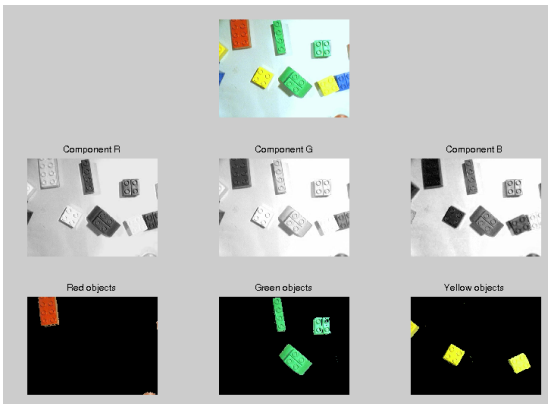
$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.000 & 0.956 & 0.621 \\ 1.000 & -0.272 & -0.647 \\ 1.000 & -1.106 & 1.703 \end{bmatrix} \begin{bmatrix} Y \\ I \\ Q \end{bmatrix}$$

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

- Formato comum em muitas webcams
- O Matlab dispõe de duas funções para as conversões:
 - `ycbcr2rgb()`
 - `rgb2ycbcr()`

Exemplos de aplicação

Exemplo em Matlab

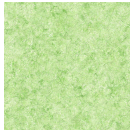


- Representação de imagem a cores: matriz com 3 "folhas", uma para cada componente: $A(:,:,1)$, $A(:,:,2)$, $A(:,:,3)$.
- Conversão para HSV: $HSV=rgb2hsv(A)$;
 - Resulta também em matriz de 3 "folhas" mas com significados diferentes: $HSV(:,:,1)$, $HSV(:,:,2)$, $HSV(:,:,3)$
- Criação de máscara para segmentar verdes com saturação de 0.3 ou mais:

```
HSV=rgb2hsv(A);  
H=HSV(:,:,1);  
S=HSV(:,:,2);  
V=HSV(:,:,3);  
greenM=( H>0.3 & H <0.5 & S>0.3 );
```

Exemplo de separação de fundo – Parte 1

- Dada a primeira imagem, obter as máscaras dos objetos por eliminação dos pixels do fundo.
- Dado que o fundo não é uniforme, binarizar pode não ser suficiente, e pode-se explorar a segmentação do fundo com cor.
- Pode-se obter uma gama de valores (H,S,V) que caracterizem o fundo para o segmentar.
- Tome-se um excerto do fundo, como este:

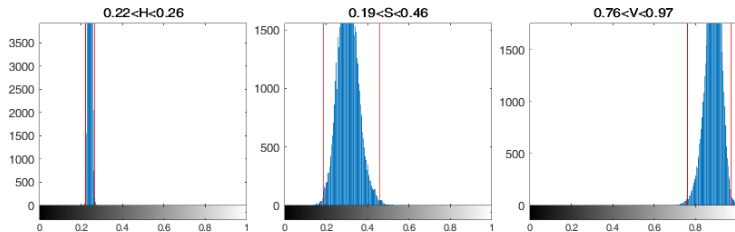


Exemplo de separação de fundo – Parte 2

- Com o excerto do fundo podem obter-se os limites com uma certa tolerância:

```
B=imread('backgd1.jpg');  
HSV=rgb2hsv(B); H=HSV(:, :, 1); S=HSV(:, :, 2); V=HSV(:, :, 3);  
tol=[0.005 0.995]; %tol=0; % use 0 if no tolerance intended  
Hlims=stretchlim(H,tol);  
Slims=stretchlim(S,tol);  
Vlims=stretchlim(V,tol);
```

- Ou seja, os histogramas para H,S e V e os seus limites a usar seriam os seguintes:



Exemplo de separação de fundo – Parte 3

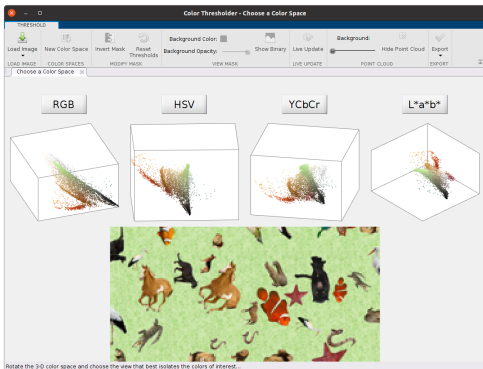
- Com estes limites pode-se obter a máscara pretendida, e os respetivos objetos:

```
A=imread('animals.png');  
HSV=rgb2hsv(A); H=HSV(:,:,1); S=HSV(:,:,2); V=HSV(:,:,3);  
mask=      (H > Hlims(1) & H < Hlims(2)); %select by Hue  
mask=mask & (S > Slims(1) & S < Slims(2)); %add a condition for saturation  
mask=mask & (V > Vlims(1) & V < Vlims(2)); %add a condition for value  
mask=~mask; %mask for objects (negation of background)  
mask=bwareaopen(mask,100); %in case we need some cleaning of "small" areas.  
mask=imclearborder(mask); %in case you want to discard objects on border
```



Ferramenta colorThresholder do Matlab

- Ferramenta interativa que permite criar máscaras em diversos espaços de cor
 - colorThresholder
- Escolher espaço de cor



- Definir os limites e exportar

