UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA BIOMÉDICA
PROCESSAMENTO DIGITAL DE IMAGENS

Laboratório 2 - Processamento de imagens coloridas

Alunos: Josiel Patrício Pereira de Oliveira
Julia Apolonio de Amorim

1 - Seleção de cores em uma imagem

É possível, por meio de processamento de imagem, exibir apenas determinadas partes da imagem que são de alguma cor específica. No algoritmo usado para gerar as imagens abaixo, as seguintes etapas foram seguidas:

a) Conversão de RGB para HSV: para simplificar a busca pelos valores corretos de intensidade;
b) Definição do máximo e mínimo para cada valor: determinar o intervalo de cada componente restringe a imagem à cor desejada;
c) Geração da máscara: a máscara consiste em uma imagem possuindo verdadeiro caso o pixel da região da imagem original esteja nos intervalos determinados, e falso caso não;
d) Filtragem da imagem: para filtrar, a imagem original é subtraída da máscara, gerando uma imagem em que só a cor escolhida estará presente.

Para exemplificar, foram utilizadas as seguintes imagens:

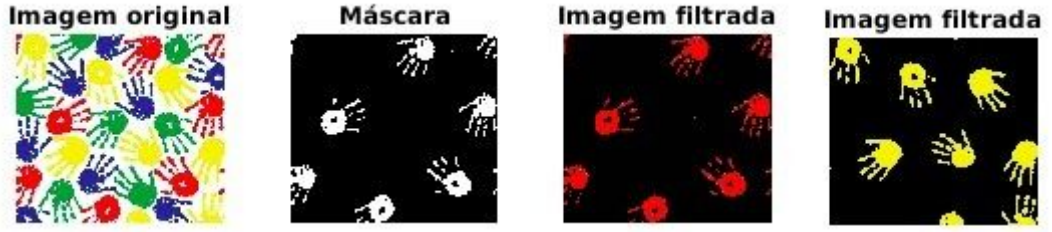


Fig. 1 - Imagem original, máscara de filtragem para a região vermelha, resultado da operação com a máscara vermelha e resultado de operação com a máscara amarela

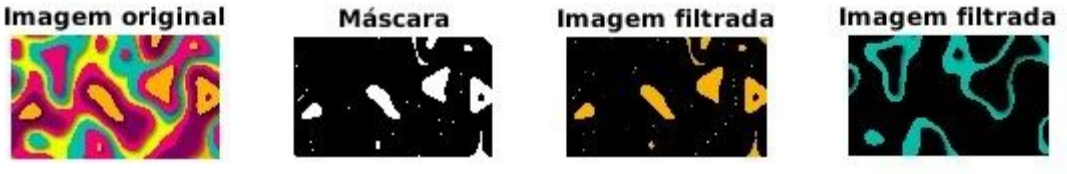


Fig.2 - Outro exemplo de imagem original, máscara de filtragem para a região laranja, resultado da operação com a máscara laranja e resultado da operação com a máscara azul turquesa

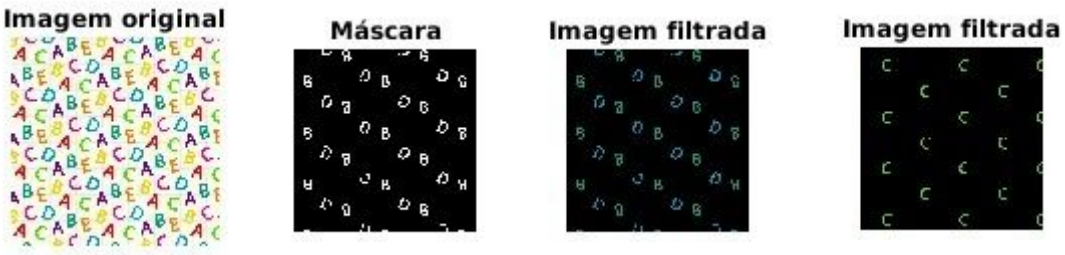


Fig.3 - Outro exemplo de imagem original, máscara de filtragem para a região azul, resultado da operação com a máscara azul e resultado da operação com a máscara verde

Com essa técnica também é possível associar mais de uma cor, basta adicionar o operador “OU” para dois intervalos de valores diferentes para cada canal, na construção da máscara. Como exemplo as imagens abaixo:

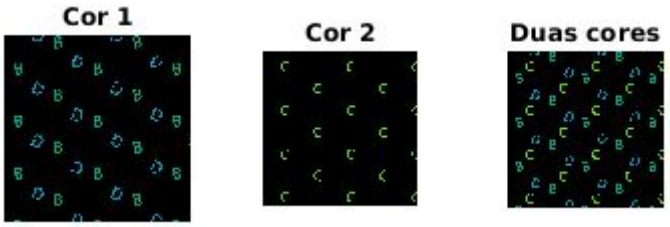


Fig.4 - Resultado da filtragem para azul, resultado da filtragem para verde e soma das duas filtragens

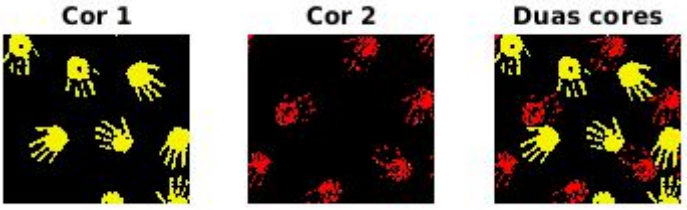


Fig.5 - Resultado da filtragem para amarelo, resultado da filtragem para vermelho e soma das duas filtragens

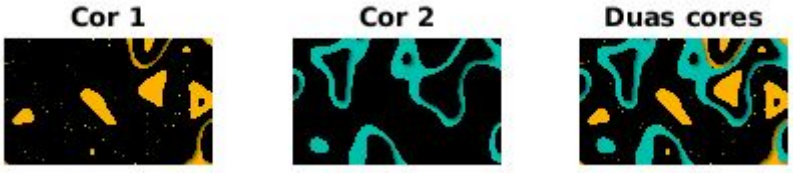


Fig.6 - Resultado da filtragem para laranja, resultado da filtragem para azul turqueza e soma das duas filtragens

2 - Comandos básicos para imagens coloridas

a) utilizando a função imfinfo foi apresentada a seguinte saída:

```
ans =

        Filename: 'C:\Program Files\MATLAB\MATLAB Production Server\R...'
```

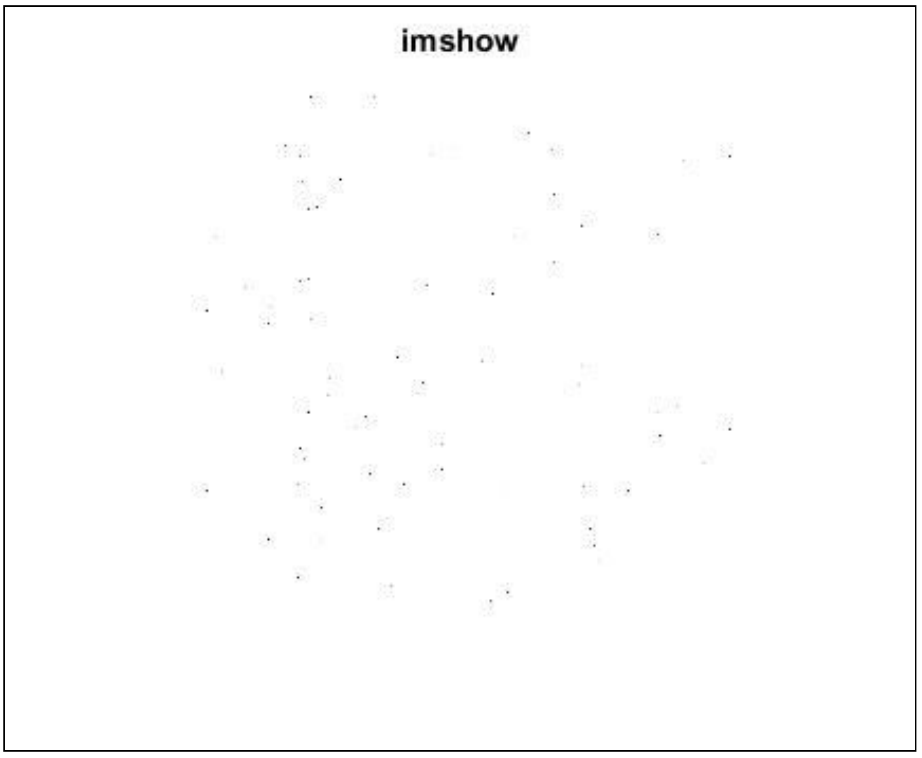
```
          FileModDate: '01-Mar-2001 09:52:38'
             FileSize: 74948
               Format: 'jpg'
        FormatVersion: ''
                Width: 500
               Height: 300
             BitDepth: 24
            ColorType: 'truecolor'
    FormatSignature: ''
NumberOfSamples: 3
       CodingMethod: 'Huffman'
      CodingProcess: 'Sequential'
            Comment: {}
```

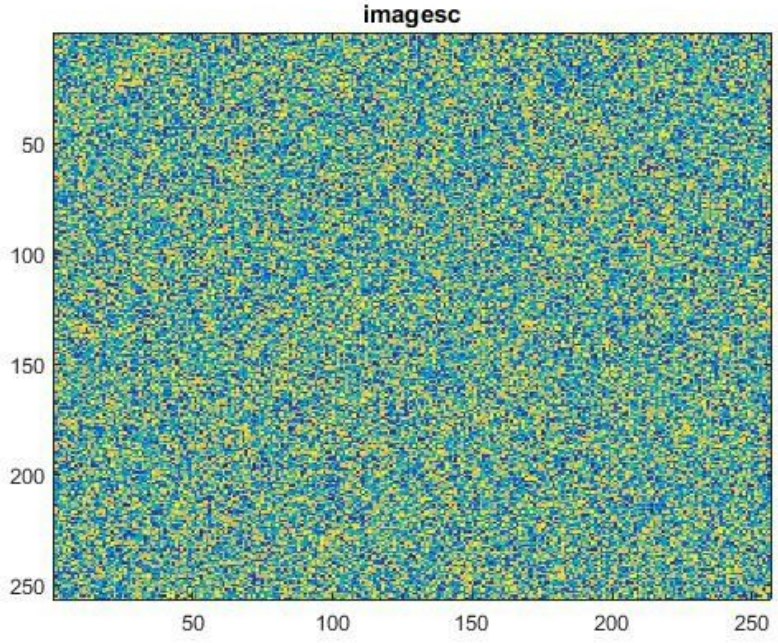
Da saída apresentada podemos destacar a importância do formato no qual a imagem está salva, as dimensões da imagem, a profundidade de bits que informa quantos bits não necessários para representar cada pixel da imagem, neste caso R, G e B com 8 bits cada cor. Também é importante a forma de codificação da imagem que neste caso é usado “Huffman” que explora apenas a redundância de codificação para compactar imagens.

b) Exibição de imagens coloridas:

Diferença entre as funções imshow e imagesc:
Como dito no enunciado da questão a função imshow requer uma matriz 2D cujos valores devem estar no intervalo [0, 1] ou [0, 255].
imagesc: aceita qualquer tipo de dado (uint8, uint16 ou duplo) e qualquer faixa de valor numérico. Ela ajusta a faixa de valores dos dados de entrada e exibe a imagem usando o mapa de cor atual ou default. O uso de ambas as funções é apresentado no trecho de código abaixo com as respectivas saídas.

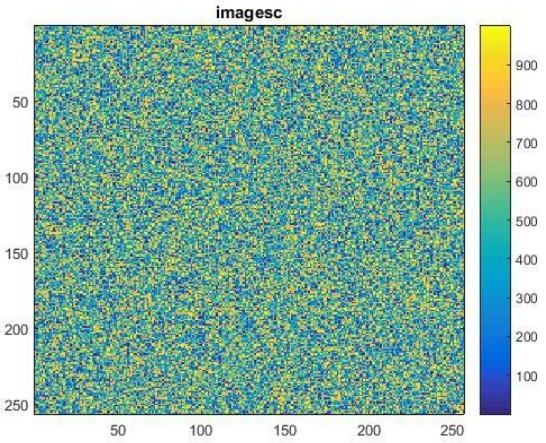
```
I = rand(256);
Imil = I * 1000;
figure();
imshow(Imil);
title('imshow');
figure();
imagesc(Imil);
title('imagesc');
```





De forma simplificada podemos interpretar a imshow como a representação da matriz 2D feita de forma que seus valores representam os tons de cinza de uma imagem e a imagesc como uma função que plota a representação dos valores passados utilizando pseudocores (isso pode ser demonstrado utilizando o comando ‘colorbar’ após o comando imagesc para apresentar a barra de pseudocor atribuída para cada intensidade), que consiste na atribuição cores para os níveis de intensidade. Importante salientar que quando são passados para a imshow valores maiores que 255 ocorre um overflow e todos os valores maiores que 255 são limitados a 255 fazendo com que todo valor a partir de 255 aparece com nível máximo de intensidade. Outro

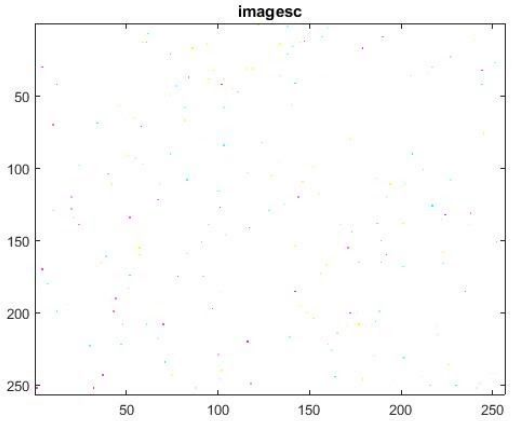
detalhe é que a função imshow pode receber como parâmetro um vetor na forma [low high] em que low indica o limite inferior que por padrão é 0 e high indica o limite superior que pode ser 1 ou 255 isso evita o overflow e deixa a imagem apresentada com os níveis de intensidades melhor representados.
A utilização dos parâmetros opcionais de imshow [low high] são demonstrados na imagem abaixo juntamente com a apresentação da barra de cores da função imagesc.



Já utilizando uma matriz I do tipo 256*256*3 que representa uma matriz com imagem colorida os resultados são os seguintes:
imshow e imagesc usando a imagem RGB não há muita diferença pois por se tratar de uma matriz 3D de ponto flutuante ambas as funções interpretam os limites no intervalo [0, 1]. Isso será apresentado nas imagens a seguir e ocorre porque o parâmetro recebido com os limites inferior e superior na forma [low high] na função imshow só é aplicado quando se trata de escalas de cinza, ou seja, usando uma matriz 2D, conforme apresentado acima.

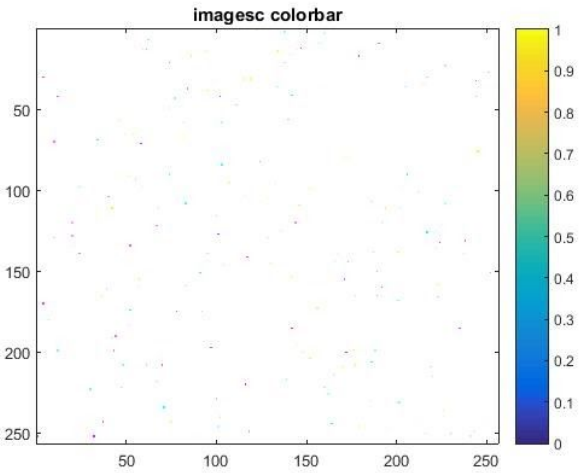
imshow RGB

imagesc RGB

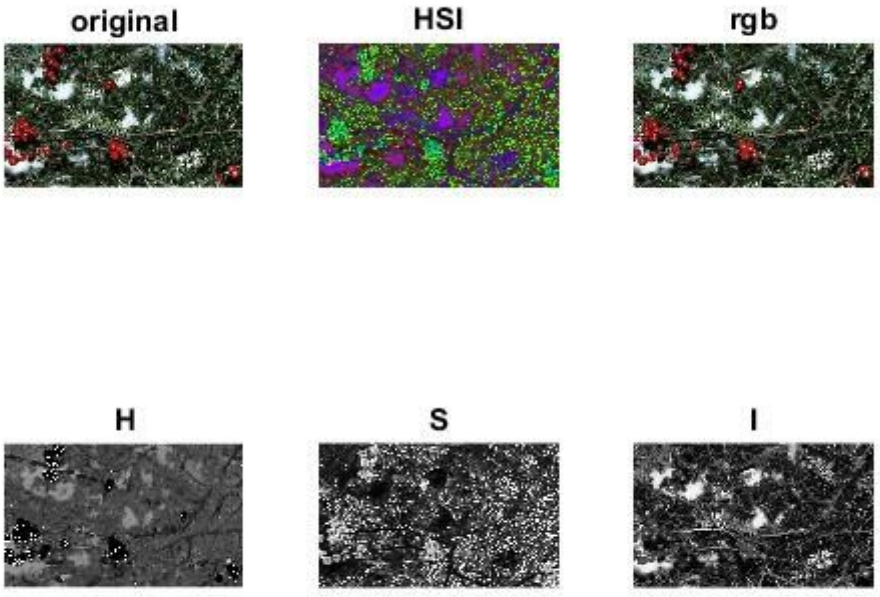


imshow com faixa de dados RGB

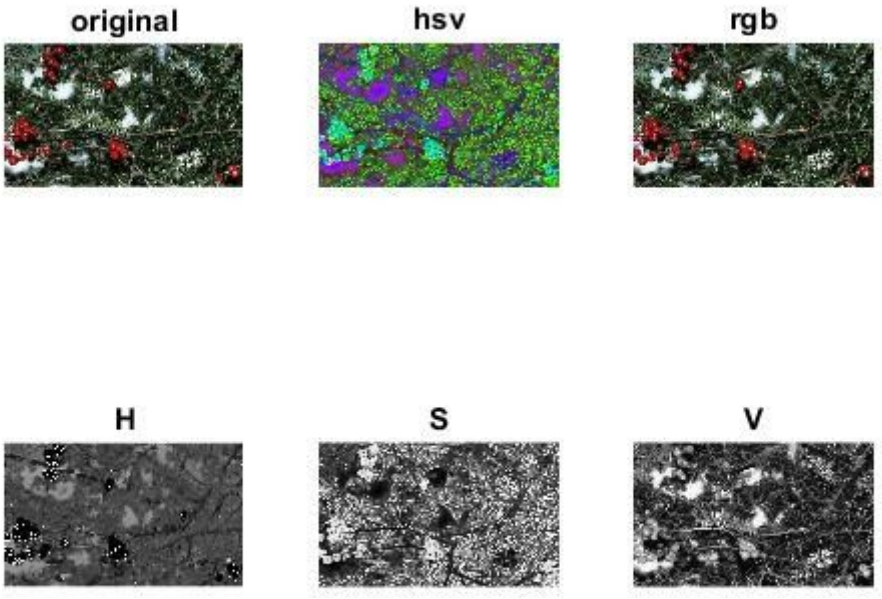
imagesc com colorbar RGB



c) RGB e HSI



d) RGB e HSV



e) Visualmente os resultados são muito parecidos, visualmente imperceptíveis, e as conversões simultâneas em ambos os casos não demonstraram perdas, pelo menos visuais, nas conversões.

3 - Processamento em pseudocores

A partir do fatiamento de intensidade nas cores da imagem, é possível transformar uma imagem em algo mais fácil de ser interpretado pelo olho humano a partir da troca de tons de cinza por cores. No exemplo abaixo, a imagem foi fatiada por intensidade em 4 regiões de cores:

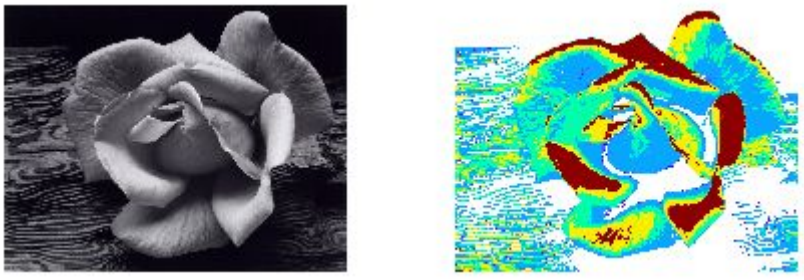


Fig. x - Imagem original e imagem depois de fatiamento por intensidade

A partir da imagem, é possível perceber que nem sempre o fatiamento de intensidade irá facilitar a compreensão da imagem pelo olho humano. Imagens com poucos níveis de fatiamento irão apresentar menos detalhes do que o observável em tons de cinza, além de que

a escolha de cores pode ser decisiva para a compreensão, avaliando se é melhor cores com alto ou baixo contraste entre si.

4 - Balanceamento de cores

O balanceamento de cores pode ser feito no matlab usando uma função chamada *chromadapt.* Como exemplo do balanceamento de cores, a imagem a seguir:

Fig. x - Imagem antes e depois do balanceamento de cores, utilizando como referencial o urso branco da caixa de presente na imagem