

Nome: Lucas de Brito Silva

E_1.

a) O recurso de "Data Cursor" que é utilizado para realizar as análises de coordenadas e níveis de cores RGB não é disponível no Octave, todavia pode ser realizado o comando *imshow* para se obter as imagens de intensidade a partir da matriz E.

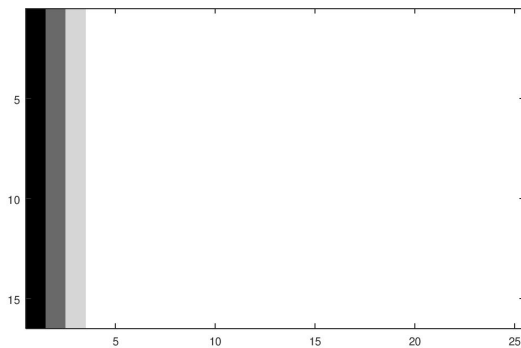
Outro ponto que difere o Matlab do Octave é a função *imshow*, em que no Matlab podemos utilizar o parâmetro *n* (ou *LIMITS*) para definir os níveis de cinza, todavia no Octave precisamos de um elemento com no mínimo duas posições, que seria o *low* e *high*, para definir os *LIMITS*, como é possível ver na captura de tela abaixo:

```
If LIMITS is a 2-element vector '[LOW, HIGH]', the image is shown using a display range between LOW and HIGH. If an empty matrix is passed for LIMITS, the display range is computed as the range between the minimal and the maximal value in the image.
```

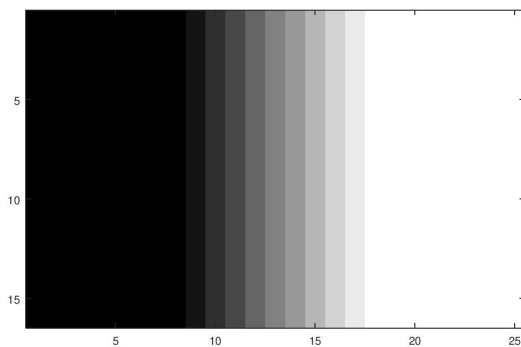
```
If MAP is a valid color map, the image will be shown as an indexed image using the supplied color map.
```

```
If a filename is given instead of an image, the file will be read and shown.
```

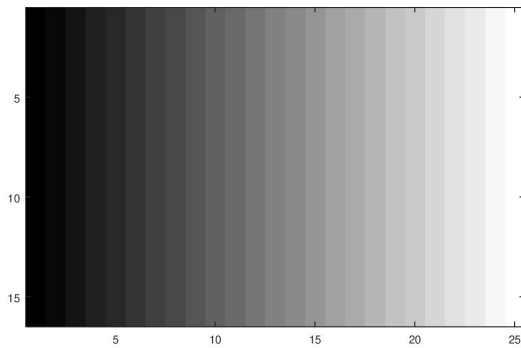
As figuras obtidas foram a partir de seus respectivos comandos:



```
figure(1); imshow(IE,[0,0.1]);
```



```
figure(2); imshow(IE,[0.3,0.7]);
```



`figure(3);imshow(IE,[]);`

Como pôde ser observado, como tentativa de suprir a falta da funcionalidade para definir os níveis de cinza do Octave, foi realizado um teste utilizando um limite de $\text{low} = 0$ e $\text{high} = 0.1$, o que resultou em uma imagem com apenas 4 cores.

A segunda imagem demonstrou 9 tons que vão do preto ao branco, sendo que todos os tons demonstrados estão entre 0.3 e 0.7, o que pode ser visto entre as colunas 9 a 17.

Já o último comando também pode ser representado por `imshow(IE)`, em que todas as cores são representadas com um limite de $\text{low} = 0$ e $\text{high} = 1$, ou seja, esse comando representa a imagem original.

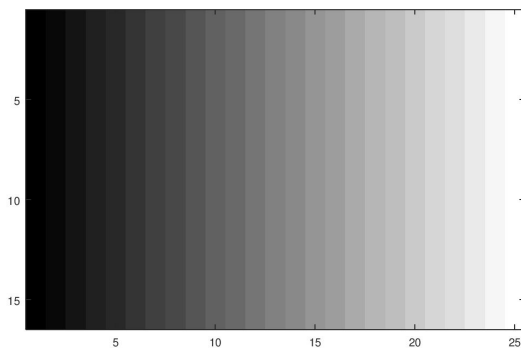


Figura F

b) Visto que não passaram por nenhuma conversão, os pixels da imagem IE recebem a classe de dados padrão, que é `double`.

c) Os pixels encontram-se no intervalo de 0 à 1.

d) Na imagem F o intervalo é de 0 à 255 e podem ser visualizados a partir dos comandos `min(F(:))` e `max(F(:))`, respectivamente.

E_2.

a) Levando em conta que não é possível fazer a análise com o Data Cursor no Octave, a diferença entre uma imagem indexada e uma imagem de intensidades está na divisão da imagem entre um Map

e os index da imagem, ou seja, na matriz da variável X estão apenas as posições das intensidades e na variável map encontram-se as intensidades de RGB. Por exemplo, na posição 0 da imagem, as intensidades serão 0,0 e 0.

b) Pois as intensidades de RGB remetem às cores em cinza. Por exemplo, se for feita uma alteração em `map(map(end,2:end) = 0)`, alterando as intensidades de verde e azul para zero e deixando a de vermelho = 1, a cor plotada na última coluna da imagem será vermelha.

c) A quantidade de cores definidas em Map e o número dos index, visto que por padrão a função `gray2ind` utiliza $n = 64$.

d) Na visualização da imagem não houve alterações, embora foram obtidas 64 cores, os index demonstrados em X não formam uma matriz de (64,) ou (64,1), mas uma matriz (16,25), em que seleciona as cores de Map de forma não sequencial como anteriormente (1,2,3...). Por exemplo, na posição (1,2) de X visualiza-se o index 3 e não 1, como anteriormente. O index 3 equivale às intensidades 0.031746, 0.031746 e 0.031746.

e) O parâmetro MAP na função `imshow`, do Octave, possui a função de ser um mapa de cores.

f) Visualmente, para o olho humano, as três imagens não apresentam diferença. Quanto à execução para a visualização, através das funções `gray2ind` e `imshow`, a diferença está na quantidade de cores de map, além da última execução não carregar um mapa de cor, passando para a função o valor do index como valor dos pixels.

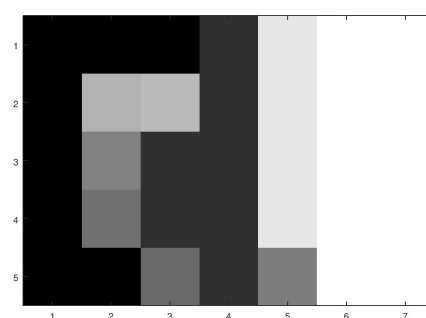
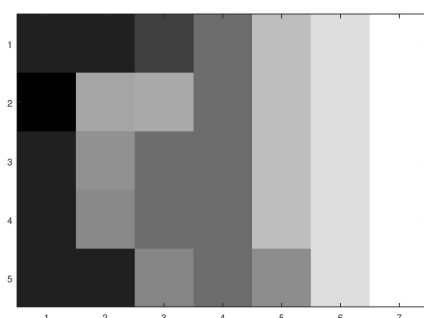
E_3.

a) `im2uint8(F);`

b) `mat2gray(F);`

c) `im2bw(F, 0.3);`

d) É possível observar que o G2 passa por duas transformações, sendo que a primeira delas é a transformação para `uint8`, e depois para `im2double`. O que merece destaque é que na primeira transformação todos os valores que são abaixo de 0, recebem 0. E quando são transformados para a classe `double` estão todos zerados, diferente da transformação de `mat2gray` que diferencia esses valores negativos, consequentemente, a imagem reproduzida. Abaixo observa-se respectivamente G1 e G2.



E_4.

a) `imwrite(h,'bubbles5.jpg','Quality',5)`

b)

```
imread('bubbles5.jpg')
```

```
imfinfo bubbles5.png
```

c)

```
h = imread('bubbles.jpg');
```

```
i = imread('bubbles5.jpg');
```

```
imshow(h), figure, imshow(i)
```

```
imfinfo Figuras1/bubbles.jpg
```

```
image_bytes_h = size(h)(1,1) * size(h)(1,2) * 8 / 8
```

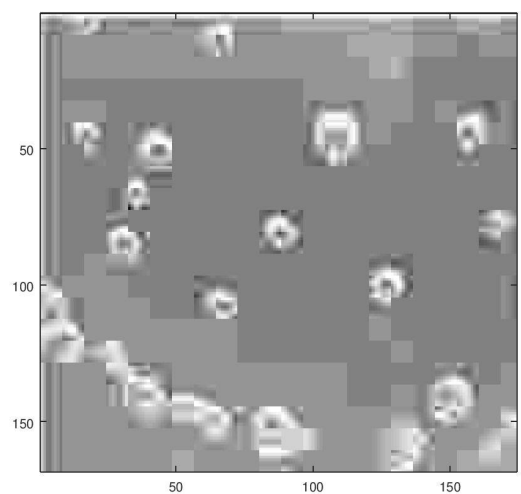
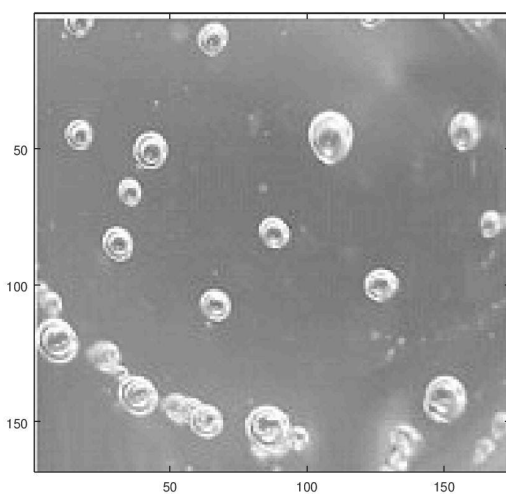
```
compress_ratio_h = image_bytes_h / 5928
```

```
imfinfo bubbles5.jpg
```

```
image_bytes_i = size(i)(1,1) * size(i)(1,2) * 8 / 8
```

```
compress_ratio_i = image_bytes_i / 607
```

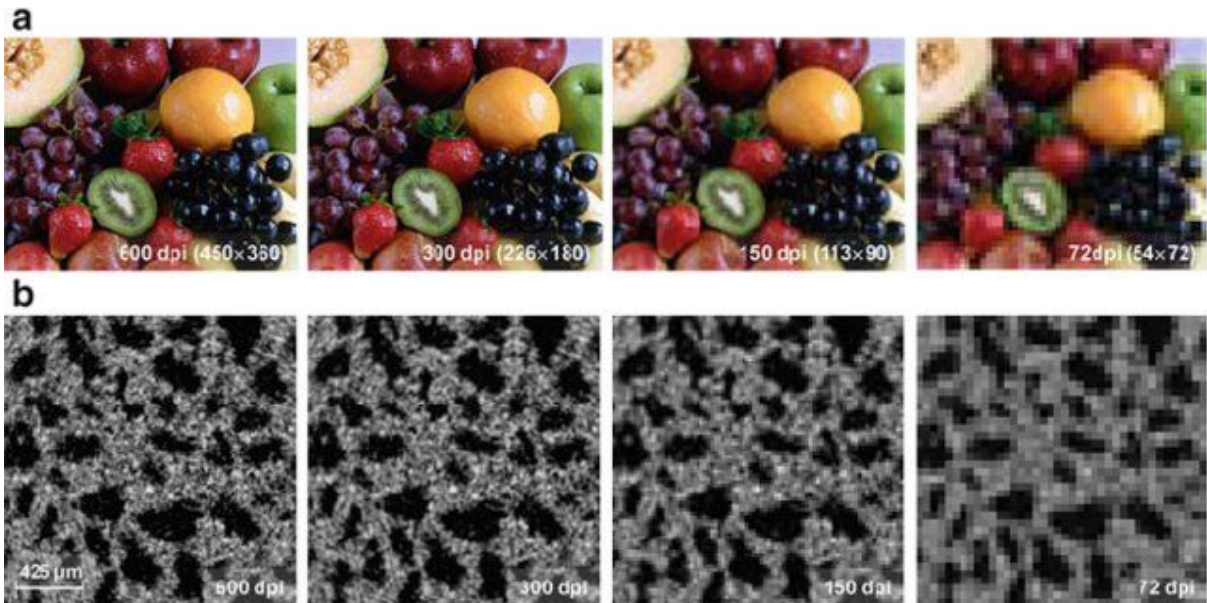
A imagem *h* possui uma taxa de compressão = 4.9312 e a imagem *i* possui uma taxa de compressão = 48.158. Além disso, vale dizer que ambas imagens possuem uma 168 linhas e 174 colunas, além de terem a mesma quantidade de bits, que no caso foi 29232. Por fim, conclui-se que uma imagem com uma compressão maior, gera uma qualidade pior para a visualização, como se observa as imagens abaixo, sendo respectivamente, *h* e *i*.



E_5. Com o software Octave não é possível realizar esse procedimento, visto que dentre os seus parâmetros não consta os mesmos tipos de compressões e não há um parâmetro [colres rowres]. Os parâmetros disponíveis nessa função podem ser visualizados através do comando `help imwrite` ou no seguinte link <https://octave.org/doc/v4.2.1/Loading-and-Saving-Images.html>.

Tentei utilizar o comando `imresize` para realizar essa atividade, mas o tamanho das imagens mudou e não o dpi, dessa forma irei fazer com base na imagem abaixo, retirada de:

https://www.researchgate.net/publication/292354800_Basics_of_Image_Analysis.



a) O tamanho das imagens se mantiveram, mas a dimensão mudou, visto que a quantidade de pixels nas linhas e colunas também foram alterados.

b) Resolução espacial em uma imagem pode ser definido pelo tamanho representado por um pixel, em relação ao mundo real, em que, quanto maior a quantidade de ppi (ou dpi), maior será a quantidade de detalhes expressos nesta imagem.

c) Não foi possível realizar por falta de funcionalidade na ferramenta Octave.