

**UNIVERSIDADE FEDERAL DO RIO
GRANDE DO SUL
INSTITUTO DE INFORMÁTICA**
INF01046 - Fundamentos de Processamento de
Imagens - 2017_2
Professor Manuel Menezes de Oliveira Neto

Relatório

Laboratório 5
Fundamentos de Processamento de Imagens em
Cores

1. Fazer download das imagens: ok

2. Procedimento para MATLAB/Octave que faça as operações:

```
function separate(image)
% question 2 starts:
I = imread(image);

[Rows, Cols, Channels] = size(I);

Red = uint8(zeros(size(I)));
Green = uint8(zeros(size(I)));
Blue = uint8(zeros(size(I)));

for i = 1:Rows
    for j = 1:Cols
        Red(i, j, 1) = I(i, j, 1);
        Green(i, j, 2) = I(i, j, 2);
        Blue(i, j, 3) = I(i, j, 3);
    end
end

figure
subplot(2,2,1);
imshow(Red);
title('Red');

subplot(2,2,2);
imshow(Green);
title('Green');

subplot(2,2,3);
imshow(Blue);
title('Blue');

subplot(2,2,4);
imshow(I);
title('Original');

% e)

newRed = uint8(zeros(size(I)));
newGreen = uint8(zeros(size(I)));
newBlue = uint8(zeros(size(I)));

newRed = Red;
newGreen = Green;
newBlue = Blue;

% question 2 ends

% question 3 starts:
```

```
red_green = uint8(zeros(size(I)));
red_blue = uint8(zeros(size(I)));
green_blue = uint8(zeros(size(I)));
```

```
red_green = newRed + newGreen;
red_blue = newRed + newBlue;
green_blue = newGreen + newBlue;
```

```
figure
subplot(2,2,1);
imshow(red_green);
title('Red + Green');
```

```
subplot(2,2,2);
imshow(red_blue);
title('Red + Blue');
```

```
subplot(2,2,3);
imshow(green_blue);
title('Green + Blue');
```

```
subplot(2,2,4);
imshow(I);
title('Original');
```

```
% question 3 ends
```

```
% question 4 starts:
```

```
N = uint8(zeros(size(I)));
NR = uint8(zeros(size(I)));
NG = uint8(zeros(size(I)));
NB = uint8(zeros(size(I)));
```

```
N = 255 - I;
```

```
NR = 255 - Red;
NG = 255 - Green;
NB = 255 - Blue;
```

```
figure
subplot(2,2,1);
imshow(NR);
title('Red negative');
```

```
subplot(2,2,2);
imshow(NG);
title('Green negative');
```

```
subplot(2,2,3);
imshow(NB);
title('Blue negative');
```

```

subplot(2,2,4);
imshow(N);
title('Original negative');

% question 4 ends

% question 5 starts:

lumi = uint8(zeros(size(I)));

for i = 1:Rows
    for j = 1:Cols
        lumi(i, j, 1) = (0.299*Red(i,j,1) + 0.587*Green(i,j,2) + 0.114*Blue(i,j,3));
        lumi(i, j, 2) = (0.299*Red(i,j,1) + 0.587*Green(i,j,2) + 0.114*Blue(i,j,3));
        lumi(i, j, 3) = (0.299*Red(i,j,1) + 0.587*Green(i,j,2) + 0.114*Blue(i,j,3));
    end
end

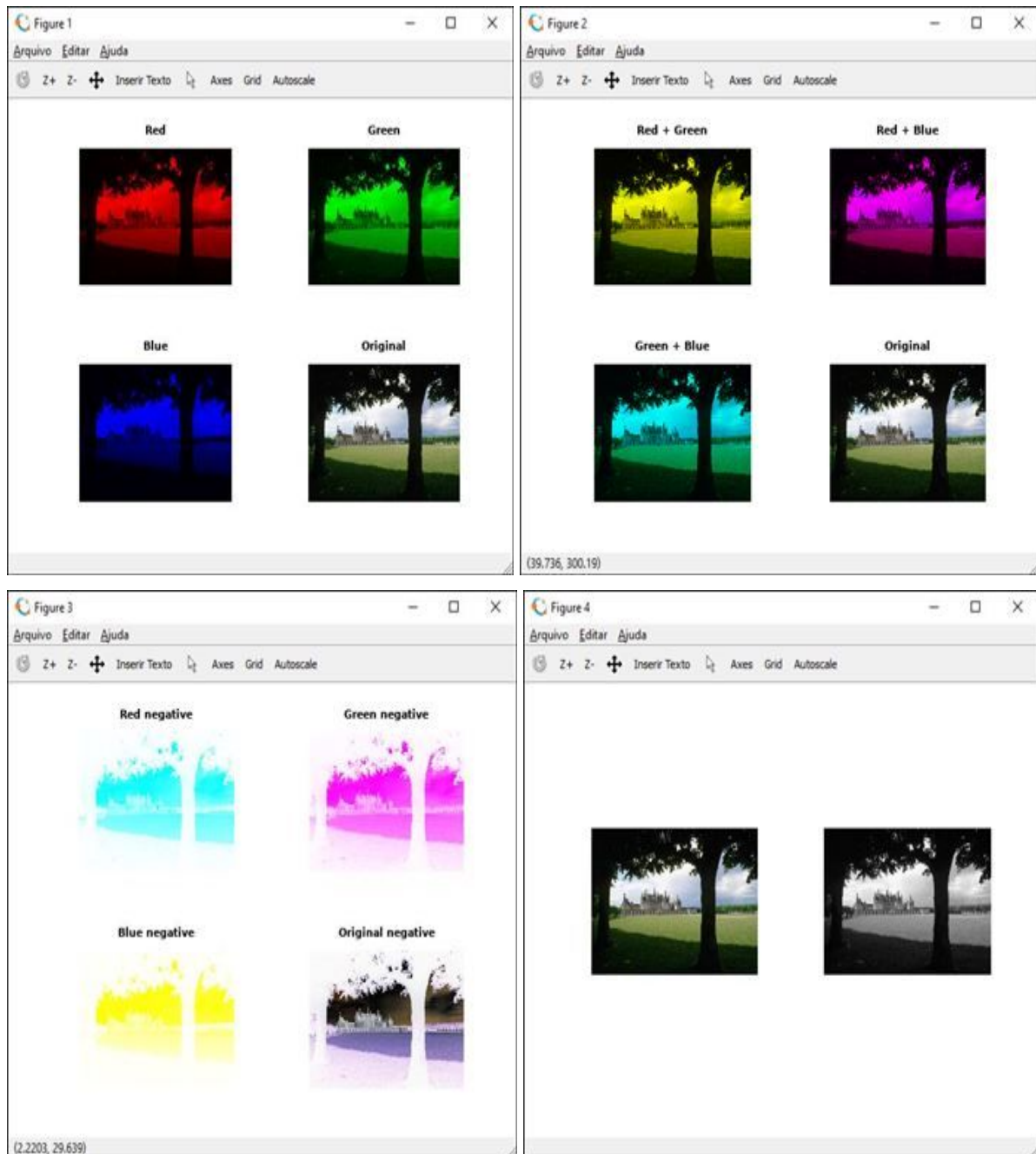
figure
subplot(1,2,1);
title('Original');
imshow(I);

subplot(1,2,2);
title('Luminância');
imshow(lumi);

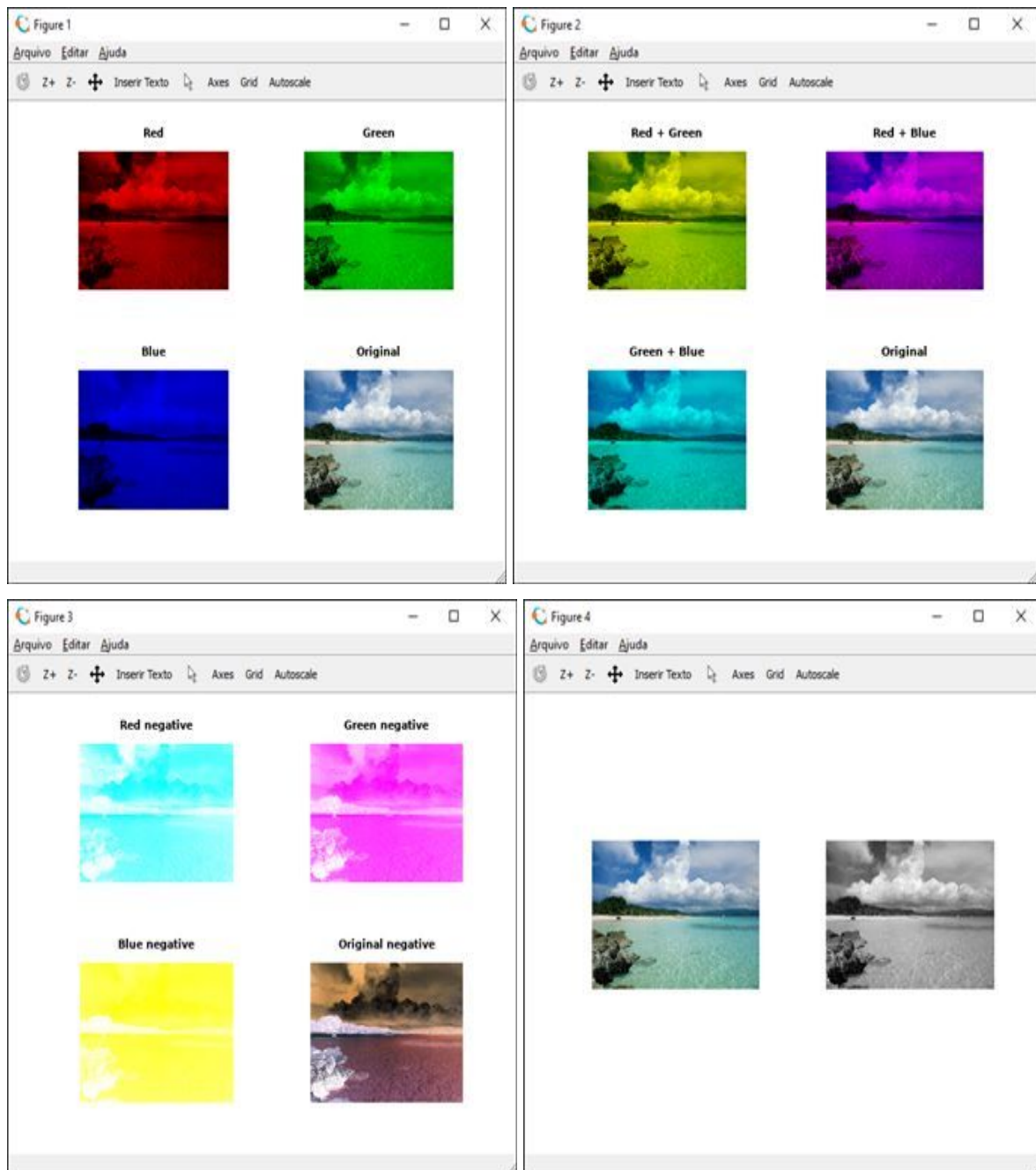
% question 5 ends
end

```

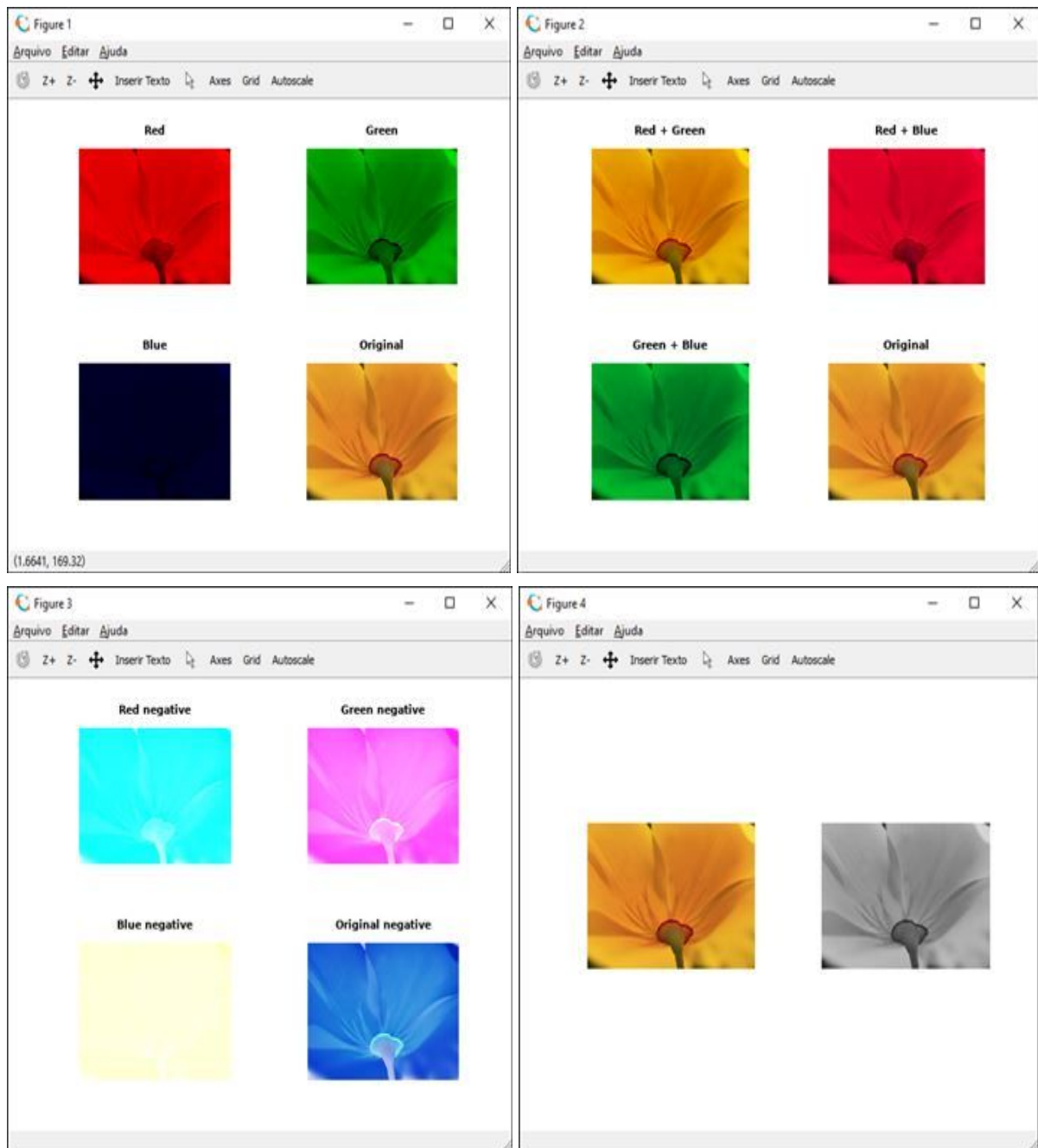
Questão 3, 4 e 5 para Chateau_small.jpg:



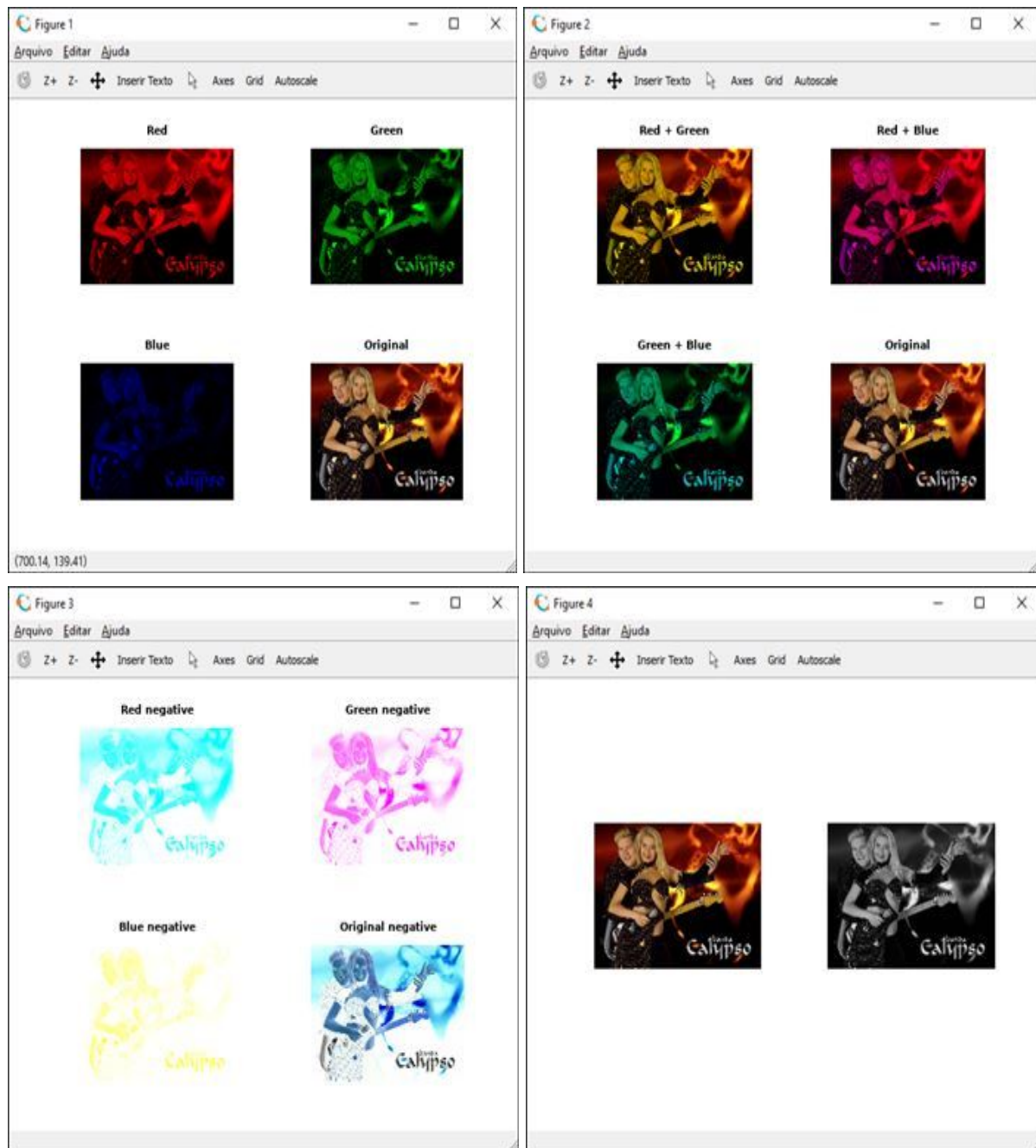
Questões 3, 4 e 5 para Paradise_small.jpg:



Questão 3, 4 e 5 para gold_petals_small.jpg:



Questão 3, 4 e 5 para BC.jpg:



Erros ocasionados pelo octave impossibilitaram a implementação do que foi pedido na questão 6:

6. Defina um kernel Gaussiano $G = \begin{bmatrix} 0.0625 & 0.125 & 0.0625 \\ 0.125 & 0.25 & 0.125 \\ 0.0625 & 0.125 & 0.0625 \end{bmatrix}$ e obtenha o resultado da convolução da imagem colorida com este kernel. Você consegue aplicar a convolução diretamente sobre a imagem? Exiba a imagem original e sua filtrada, em uma janela definida pelo comando subplot(1, 2,...).

Após definir o Kernel G, a convolução não era possível de ser feita.

Para todas as outras, o script separate fazia todos os passos, como separar os canais R, G, B, misturar as cores, fazer o negativo de cada canal, e o negativo.