

Lista 2 – PDI

Gabriel Bicalho Ferreira 18.1.4024

Questão 1:

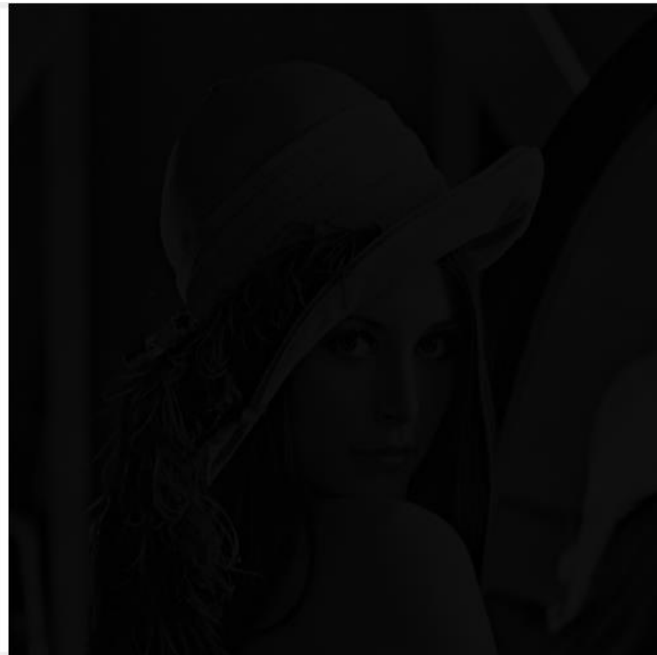
Código para implementação:

```
function img2 = transformacaoCinza(n)
    img = imread('C:\Users\gabri\Desktop\6 periodo\PDI\Lista 2\lennaRGB.tif');
    img = double(img);
    i = 0;
    R = img(:, :, 1);
    G = img(:, :, 2);
    B = img(:, :, 3);
    while i < n
        R = R*0.299;
        G = G*0.587;
        B = B*0.114;
        img2 = R + G + B;
        %img2 = cat(3, R ,G, B);
        i = i+1;
    end
    img2 = uint8(img2);
    imshow(img2)
end
```

Imagem com apenas uma interação do algoritmo, n=1



Imagem com 5 interações, $n = 5$:



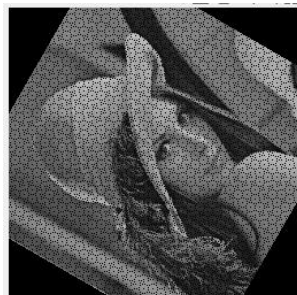
A cada interação a imagem fica mais escura.

Questão 2:

Código para **rotação**:

```
function nimg = minhaRotacao(img, ang)
    [lin, col, ~] = size(img);
    nimg = zeros(lin, col);
    matT1 = [1 0 -lin/2; 0 1 -col/2; 0 0 1];
    matR = [sind(ang) -cosd(ang) 0; cosd(ang) sind(ang) 0; ...
            0 0 1];
    matT2 = [1 0 lin/2; 0 1 col/2; 0 0 1];
    maT = matT2 * matR * matT1;
    for i = 1 : lin
        for j = 1 : col
            ncoord = maT * [i; j; 1];
            ni = floor(ncoord(1));
            nj = floor(ncoord(2));
            if ni > 0 && ni <= lin && nj > 0 && nj <= col
                nimg(ni, nj) = img(i,j);
            end
        end
    end
    nimg = uint8(nimg);
end
```

Rotação de uma imagem em 30°:



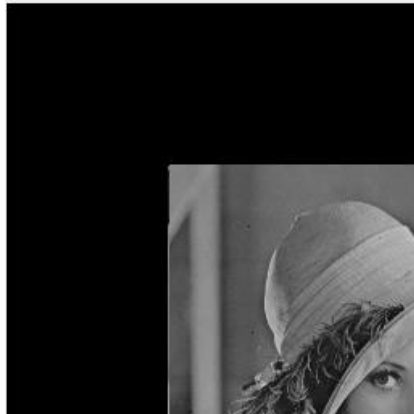
Translação

Código para translação:

```
function nimg = minhaTranslacao(img,dx, dy)
    [lin, col, ~] = size(img);
    nimg = zeros(lin, col);
    maT = [1 0 dx; 0 1 dy; 0 0 1];

    for i = 1 : lin
        for j = 1 : col
            ncoord = maT * [i; j; 1];
            ni = floor(ncoord(1));
            nj = floor(ncoord(2));
            if ni > 0 && ni <= lin && nj > 0 && nj <= col
                nimg(ni, nj) = img(i,j);
            end
        end
    end
    nimg = uint8(nimg);
end
```

Imagem com uma translação em 100 unidades em x e 100 unidades em y:



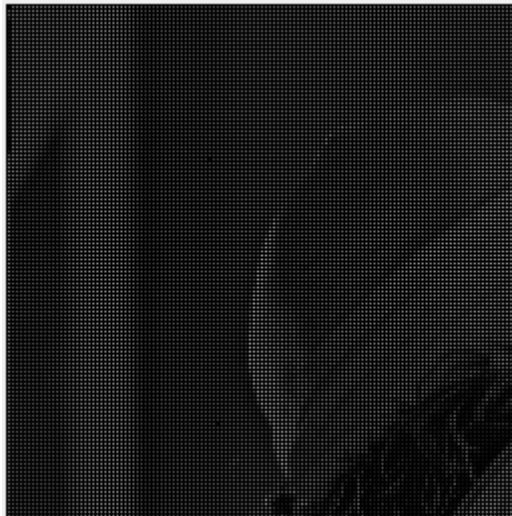
Escala:

Código para a realização do escalamento:

```
function nimg = minhaEscala(img,sx, sy)
    [lin, col, ~] = size(img);
    nimg = zeros(lin, col);
    maT = [sx 0 0; 0 sy 0; 0 0 1];

    for i = 1 : lin
        for j = 1 : col
            ncoord = maT * [i; j; 1];
            ni = floor(ncoord(1));
            nj = floor(ncoord(2));
            if ni > 0 && ni <= lin && nj > 0 && nj <= col
                nimg(ni, nj) = img(i,j);
            end
        end
    end
    nimg = uint8(nimg);
end
```

Imagem de uma transformação de escala utilizando 2 em x e 2 em y. temos uma ampliação do chapéu da Lenna.



Cisalhamento

Código para realização do cisalhamento:

```
function nimg = meuCisalhamento(img,shx)
    [lin, col, ~] = size(img);
    nimg = zeros(lin, col);
    maT = [1 shx 0; 0 1 0; 0 0 1];

    for i = 1 : lin
        for j = 1 : col
            ncoord = maT * [i; j; 1];
            ni = floor(ncoord(1));
            nj = floor(ncoord(2));
            if ni > 0 && ni <= lin && nj > 0 && nj <= col
                nimg(ni, nj) = img(i,j);
            end
        end
    end
    nimg = uint8(nimg);
end
```

Imagem com uma distorção de 0.8:



Questão 3:

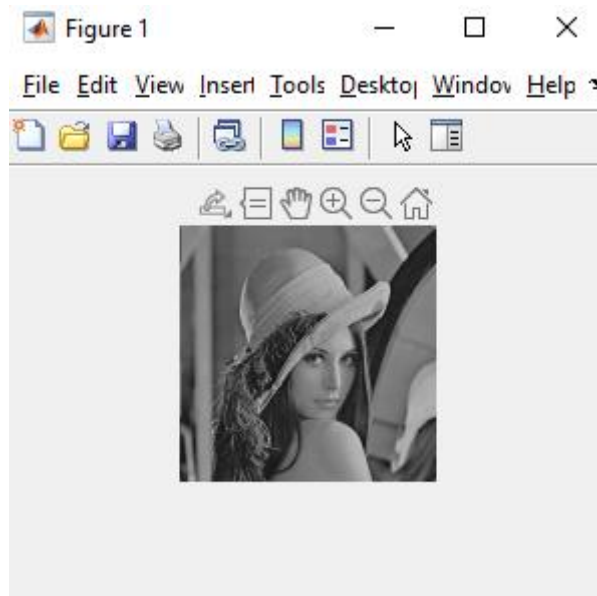
Para essa questão, foi criada duas funções, uma para diminuir a resolução da imagem para a metade, e outra função, para pegar a imagem reduzida e aumentar sua resolução ao dobro.

Função para diminuir:

```
function nimg = resolucaoReduzida(img)
    [lin, col, ~] = size(img);
    img2 = zeros(lin/2, col/2);
    aux = (1:2:lin);
    for i = (1 : lin/2)
        for j = (1: lin/2)
            img2(i, j) = mean(mean(img(aux(i):aux(i) + 1, aux(j) : aux(j) + 1)));
        end
    end

    nimg = uint8(img2);
end
```

Imagem retornada:



Função para aumentar a resolução, que vai pegar a imagem anterior e aumentar a resolução:

```

function nimg = resolucaoAumentada(img)
    [lin, col, ~] = size(img);
    img2 = zeros(lin*2, col*2);

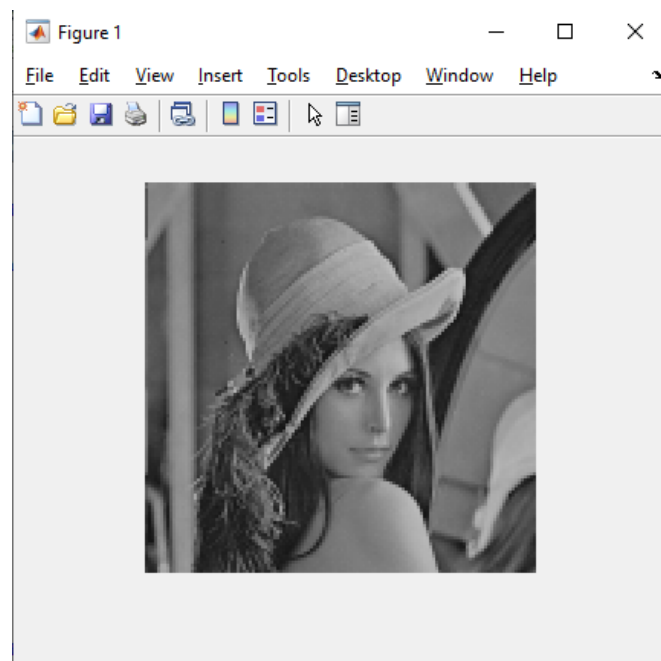
    aux = (1: 2: lin*2);
    for i = (1 : lin)
        for j = (1: lin)
            img2((aux(i):aux(i) + 1), (aux(j) : aux(j) +1)) = img(i, j);

        end

    end
    nimg = uint8(img2);
end

```

Imagem retornada:



Como é possível perceber, a qualidade de imagem é prejudicada um pouco.

Questão 4

rgb2gray -> converte em uma imagem colorida em uma imagem em escala de cinzas.

rgb2ind -> converte uma imagem colorida em uma imagem indexada.

im2double -> Converte uma imagem em uma precisão dupla.

Im2bw -> converte uma imagem em binária, com base no limite.

Código utilizado para teste das funções:

```
function nimg = testeFuncoes()
    img = imread('C:\Users\gabri\Desktop\6 periodo\PDI\Lista 2\lennaRGB.tif');
    nimg = rgb2gray(img);
    imwrite(nimg, 'Cinza.png');
    [nimg2, x] = rgb2ind(img, 32);
    imwrite(nimg2, 'ind.png');
    nimg3 = im2double(img);
    imwrite(nimg3, 'double.png');
    nimg4 = im2bw(img);
    imwrite(nimg4, 'binary.png')
end
```

Questão 5:

Para espelhar de forma vertical:

```
function nimg = espelhoVertical(img)
    [lin, col, ~] = size(img);
    aux = [col:-1:1];
    img2 = zeros(lin, col);
    img2 = img(:, aux);
    imshow(img2);
    nimg = img2;
end
```

Imagem resultado dessa função:



Agora para espelhar na horizontal, é utilizado o seguinte código:

```
function nimg = espelhoHorizontal(img)
    [lin, col, ~] = size(img);
    aux = [lin:-1:1];
    img2= zeros(lin, col);
    img2 = img(aux, :);
    imshow(img2);
    nimg = img2;
end
```

Espelhando na horizontal, a imagem já espelhada na vertical, temos o seguinte resultado:

