UNIVERSIDADE FEDERAL DE OURO PRETO - UFOP
DEPARTAMENTO DE COMPUTAÇÃO - DECOM

Disciplina: PROCESSAMENTO DE IMAGENS - BCC 326
Prof.: Guillermo Cámara Chávez
Curso: Ciência da Computação

LISTA 02

Gustavo Lucas Moreira[1]
gustavo.lucas@aluno.ufop.edu.br

17.2.4289

1. Converta uma imagem colorida para tons de cinza (luminancia).

```
function nimg = luminancia (img)
  img = double (img);
  R = img(:,:,1);
  G = img(:,:,2);
  B = img(:,:,3);
  for i = 1:100
    nimg = 0.299*R+0.587*G+0.114*B;
    nimg = uint8(nimg);
  endfor;
 endfunction;
```

2. Implemente as funções de transformação geométrica
   a. Rotação

```
function nimg2 = rotacao(img)
  [lin,col,~] = size(img);
  nimg = zeros(floor(lin/2),floor(col/2));
  for i = 1:2:lin
    for j = 1:2:col
      nimg(i,j) = img(i,j);
    endfor
  endfor
  nimg2 = zeros(lin, col);
  for i = 1:2:lin
    for j = 1:2:col
      nimg2(i,j) = nimg(i,j);
    endfor
```

[1] Graduando do Curso de Ciência da Computação da Universidade Federal de Ouro Preto

```
  endfor
  nimg2 = uint8(nimg2);
endfunction;
```

b. Escalonamento

```
function nimg = escalamento(img , x, y)
  [col, lin, ~] = size(img);
  nCol = col * x;
  nLin = lin * y;

  lInt = ceil([1:(size(img,1)*y)]./(y));
  cInt = ceil([1:(size(img,2)*x)]./(x));
  d = double(img);

  aux = d(:,:,1);
  R = aux(lInt, :);
  R = R(:, cInt);

  aux = d(:, :, 2);
  G = aux(lInt, :);
  G = G(:, cInt);
  aux = d(:, :, 3);

  B = aux(lInt, :);
  B = B(:, cInt);

  nimg = zeros([nLin, nCol, 3]);
  nimg(:, :, 1) = R;
  nimg(:, :, 2) = G;
  nimg(:, :, 3) = B;

  nimg = uint8(nimg);
 endfunction;
```

c. Translação

```
function nimg = translacao(img, v, h)
  [lin, col, ~] = size(img);
  nimg = zeros(lin, col);
  sLin = 1; eLin = lin; iLin = 1;
  sCol = 1; eCol = col; iCol = 1;
  if v > 0
```

```
    sCol = col; eCol = 1; iCol = -1;
  endif
  if h < 0
    sLin = lin; eLin = 1; iLin = -1;
  endif
  for i = sLin : iLin : eLin
    for j = sCol : iCol : eCol
      if i-h >= 1 && i-h <= lin && j+v >= 1 && j+v <= col
          nimg(i-h, j+v) = img(i, j);
      endif
    endfor
  endfor
  nimg = uint8(nimg);
 endfunction;
```

d. Cisalhamento

```
function nimg = cisalhamento(img, x, y)
  [lin, col, ~] = size(img);
  nimg = zeros(lin*2, col*2);

  for i = 1:lin
    for j = 1:col
      if j+y > 0
        nimg(i, j+y) = img (i,j);
      endif
      y = y + 1;
    endfor
  endfor

  nimg = uint8(nimg);
  imshow(nimg);
endfunction;
```

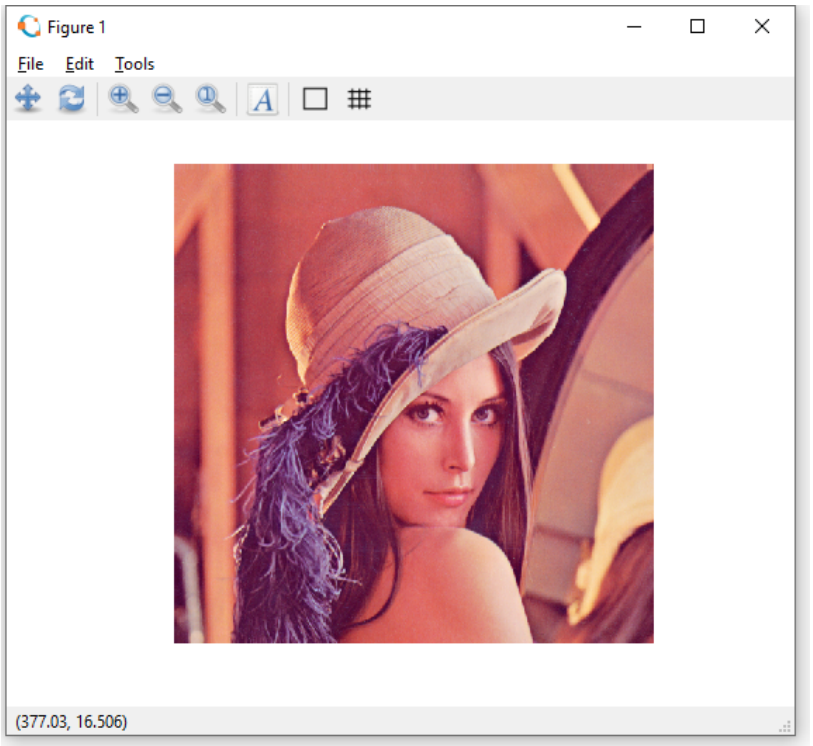
3. Modificar a resolução de uma imagem à metade. E logo multiplicar o tamanho da nova imagem, de tal forma que tenha novamente a resolução original.

```
>> reduzida = escalamento(img, 0.5, 0.5)
>> imwrite (reduzida, 'reduzida.tif')
>> duplicada = escalamento (reduzida, 2, 2)
>> imwrite(duplicada, 'duplicada.tif')
```

4. Testar as funções *rgb2gray(), rgb2ind(), im2double(), im2bw() e salvar as images com imwrite()*
   Funções testadas e imagens resultado estão salvas na pasta.

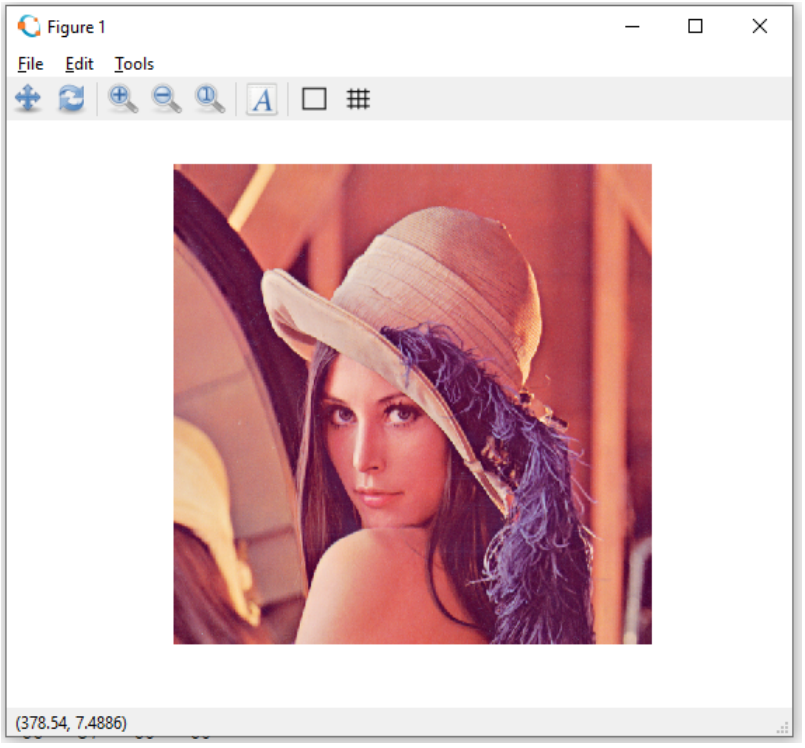
5. Dada uma imagem, primiero espelhar a imagem na vertical, depois espelhar novamente a mesma imagem na horizontal.

```
>> imshow (img)
```




```
>> vertical = img(:,end:-1:1,:)
```

```
>> imshow(vertical)
```



```
>> horizontal = vertical(end:-1:1,:,:)
>> imshow (horizontal)
```