

Computer Image Processing

Classes 4 - arithmetic operations, histogram

Example 1

Reducing image size 8 x 8 pixels to size 8 x 7 with the use of three different interpolation methods.

```
L = zeros(8);
L(1:2:end,1:2:end) = 1;
L(2:2:end,2:2:end) = 1;
imshow(L, 'InitialMagnification', 'fit');

L2a=imresize(L,[7 8], 'nearest')
pause
figure
imshow(L2a, 'InitialMagnification', 'fit')
pause
L2b=imresize(L,[7 8], 'bilinear')
figure
imshow(L2b, 'InitialMagnification', 'fit')
pause
L2c=imresize(L,[7 8], 'bicubic')
figure
imshow(L2c, 'InitialMagnification', 'fit')
pause
L2d=imresize(L,[7 8], 'lanczos3')
figure
imshow(L2d, 'InitialMagnification', 'fit')
```

Exercise 1

Change the resolution of the checkerboard image with the original size 4 x 4 to 8 x 8 using different interpolation methods.

```
L = zeros(4);
L(1:2:end,1:2:end) = 1;
L(2:2:end,2:2:end) = 1;
imshow(L,'InitialMagnification','fit');

L2a=imresize(L,[8 8],'nearest')
figure
imshow(L2a, 'InitialMagnification','fit')
L2b=imresize(L,[8 8],'bilinear')
figure
imshow(L2b, 'InitialMagnification','fit')
L2c=imresize(L,[8 8],'bicubic')
figure
imshow(L2c, 'InitialMagnification','fit')
L2d=imresize(L,[8 8],'lanczos3')
figure
imshow(L2d, 'InitialMagnification','fit')
```

Exercise 2

Rotate the checkerboard image with the original 8 x 8 size by 45 degrees using different interpolation methods

```
L = zeros(8);
L(1:2:end,1:2:end) = 1;
L(2:2:end,2:2:end) = 1;
imshow(L,'InitialMagnification','fit');

L2a=imrotate(L,45,'nearest')
figure
imshow(L2a, 'InitialMagnification','fit')
L2b=imrotate(L,45,'bilinear')
figure
imshow(L2b, 'InitialMagnification','fit')
L2c= imrotate(L,45,'bicubic')
figure
imshow(L2c, 'InitialMagnification','fit')
```

Arithmetic operations

Example 2

Adding a constant value to the monochrome image (increase brightness):

```
L1=imread('portrait.jpg');
figure;
imshow(L1);
L2=(L1)+50;
figure;
imshow(L2)
```

or

```
L1=imread('portrait.jpg');
L1=double(L1)/255;
figure;
imshow(L1);
L2=L1+0.3;
figure;
imshow(L2)
```

Example 3

Some information lost after brightening the image with the use of saturation method

```
L1=imread('portrait.jpg');
L1=double(L1)/255;
figure;
imshow(L1);
L2=L1+0.5;
figure;
imshow(L2);
L3=(L2>1)-0.5;
figure;
imshow(L3)
```

Example 4

Creating a simple menu to control the brightness of the image

```
L1=imread('portrait.jpg');
figure;
imshow(L1);

while 1
    choice = menu('Menu','Add 10','Subtract 10','Exit');
    switch choice
        case 1
            L1 = L1 + 10;
            imshow(L1);
        case 2
            L1 = L1 - 10;
            imshow(L1);
        case 3
            close
            break
    end
end
```

menu(title, option1, option2) - creating a menu in a new window, with the buttons described as 'options'. Returns the value corresponding to the number of the button that was pressed

Exercise 3

Create and program a menu, in which user will be able to select the following operations on the image:

- adding a random number from the range 0.0 - 1.0
- subtracting a random number from the range 0.0 - 1.0
- logarithm
- square root
- power
- restoring original image

Remember to use element-wise operations if necessary.

```
L1=imread('portrait.jpg');
L1 = double(L1)/255;
figure;
imshow(L1);
original = L1;

while 1
    wyb = menu('Menu','Add random','Subtract random','Logarithm','Square root',...
        'Power','Original image','Exit');
    switch wyb
        case 1
            L1 = L1 + rand(1);
            imshow(L1);
        case 2
            L1 = L1 - rand(1);
            imshow(L1);
        case 3
            L1=log(L1+1);
```

```

    imshow(L1);
case 4
    L1 = sqrt(L1);
    imshow(L1);
case 5
    L1 = L1.^3;
    imshow(L1);
case 6
    L1 = original;
    imshow(L1);
case 7
    close
    break
end
end

```

Image histogram

Example 5

Low-contrast image histogram

```

L1=imread('portrait1.jpg');
figure;
imshow(L1);
figure;
imhist(L1);
[y,x]=imhist(L1);
figure;
bar(x,y);
figure;
stairs(x,y);

```

Exercise 4

Lighten and then darken image *portrait1.jpg* and display histograms of transformed images.

Exercise 5

Display histograms of the images *3.jpg* and *4.jpg*, analyze them and adjust given images basing on information achieved from histograms. Use the *imadjust* function.

```

L1 = imread('3.jpg');
L2 = imread('4.jpg');
figure;
imshow(L1);
figure;
imhist(L1);
figure;
imshow(L2);
figure;
imhist(L2);
L1a = imadjust(L1,[120/255 220/255],[0 1]);
figure;
imshow(L1a);

```

```
figure;  
imhist(L1a);  
L2a = imadjust(L2,[0 120/255],[0 1]);  
figure;  
imshow(L2a);  
figure;  
imhist(L2a);
```