407 Comp Lab 4

Dr. Hewayda Lotfy

Pixelwise Transform Functions

Note that MATLAB have the pointwise operation instead of using the for loop for accessing each pixel location

Q1: Try Gamma transformation with different values of gamma and c using the equation $s = cr^{\gamma}$,

fix the value of c to one for example and then use p = 0.5 and 2,

fix the value of γ to 0.4 for example and then use c=0.5 and c=2,

show the image and the histogram before and after applying the gamma transform and explain the resulting images.

```
clear
clc

image=imread('pout.tif');
image_double=im2double(image);

c=input('enter c ');
gama=input('enter gama ');

imout = c * power(image_double,gama) * 255 ;
imout = uint8(imout);

subplot (2,2,1); imshow(image);
subplot (2,2,2); imhist(image);
subplot (2,2,3); imshow(imout);
subplot (2,2,4); imhist(imout);
```

Q2. Perform log transform of the cameraman.tif image. The formula is s = c * log(1 + r). Change c and show image and histogram (before and after) each time you apply log transform

```
clear
clc

image=imread('cameraman.tif');
image_double=im2double(image);

c=input('enter c ');

imout = c * log(1+image_double) * 255 ;
imout = uint8(imout);

subplot (2,2,1); imshow(image);
subplot (2,2,2); imhist(image);
subplot (2,2,3); imshow(imout);
subplot (2,2,4); imhist(imout);
```

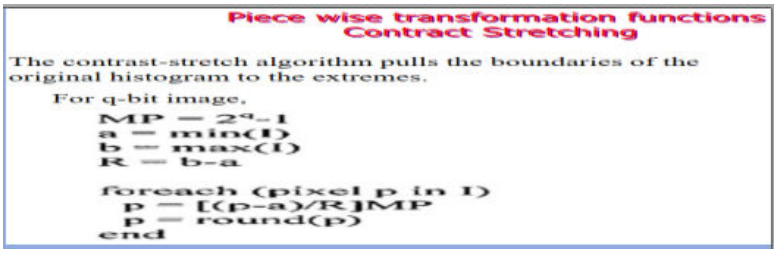
Q3- Convert the input image into its negative by point transformation.

```
clear
clc

image=imread('pout.tif');

imout = (255-image) ;
imout = uint8(imout);

subplot (2,2,1); imshow(image);
subplot (2,2,2); imhist(image);
subplot (2,2,3); imshow(imout);
subplot (2,2,4); imhist(imout);
```

Q4- Apply contrast/histogram algorithm stretching in the lecture to pout.tif image

```
clear
clc
img=imread('pout.tif');
[r,c]=size(img);
minPix=min(min(double(img)));
maxPix=max(max(double(img)));
R=maxPix-minPix;
L=255;
img=double(img);

newImg=zeros(r,c);
for i=1:r
   for j=1:c
      newImg(i,j)=L*((img(i,j)-minPix)/R);
      newImg(i,j)=round(newImg(i,j));
   end
end
img=uint8(img);
newImg=uint8(newImg);
subplot(2,2,1);imshow(img)
subplot(2,2,2);imhist(img)
subplot(2,2,3),imshow(newImg,[]);
subplot(2,2,4),imhist(newImg);
```

```
clear
clc
image=imread('pout.tif');
imout = imadjust(image , stretchlim(image,[0,1]) , [] ) ;
subplot (2,2,1); imshow(image);
subplot (2,2,2); imhist(image);
subplot (2,2,3); imshow(imout);
subplot (2,2,4); imhist(imout);
```



Q5- Apply average filtering:

1. Add noise to image using the MATLAB function imnoise (choose salt&pepper and speckle noises), now you have two noisy images
2. Define the average filters
    a. h1=1/9*ones(3,3);
    b. h2=1/25*ones(5,5);
3. Apply these two filters to each of the noisy images using function conv2
4. Show the images before and after filtering

Simple example:

```
b=imnoise(a,'salt & pepper');
h1=1/9*ones(3,3);
b1=conv2(b,h1,'same');
and so on
```

```
clear
clc
a=imread('pout.tif');
b=imnoise(a,'salt & pepper');
h1=1/9*ones(3,3);
b1=conv2(b,h1);
h2=1/25*ones(5,5);
b2=conv2(b,h2);
subplot(2,2,1);imshow(a)
subplot(2,2,2);imshow(b);
subplot(2,2,3);imshow(uint8(b1))
subplot(2,2,4);imshow(uint8(b2))
```