

Homework 3 for Kun

Introduce to image process

All codes are attached on the last page.

Histogram Equalization

a.

```
1 % Histogram Equalization
2 % a
3 I = im2uint8(rgb2gray(imread("amazon.png")));
4 imshow(I), title("Original Image");
```

Res:

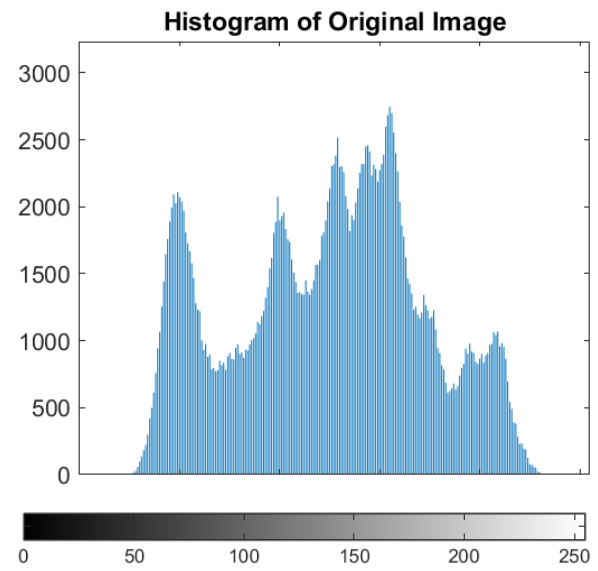
Original Image



b.

```
6 % b
7 imhist(I), title("Histogram of Original Image");
```

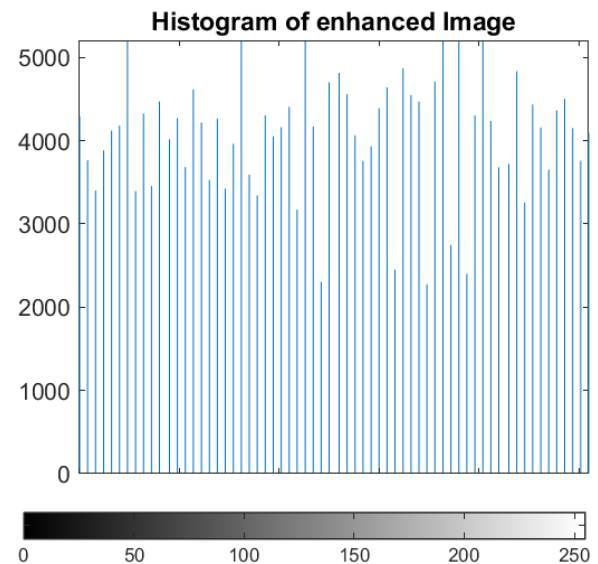
Res:



c.

```
9 % c
10 img_he = histeq(I);
11 imhist(img_he), title("Histogram of enhanced Image");
12 imshow(img_he), title("Enhanced Image");
```

Res:



Enhanced Image



a.

```
20 % Filtering
21 % a
22 h = fspecial('sobel');
23 aft_my_fil = myFilter(I,h);
24 imshow(aft_my_fil), title("After my filter function with Sobel kernel");
```

```
56
57 function res = myFilter(I,h)
58 % We can add zero padding to original image
59 [img_h,img_w] = size(I);
60 hsize = size(h);
61 padsize = [(hsize(1)-1)/2,((hsize(2)-1)/2)];
62 padded_img = zeros(img_h+padsize(1)*2,img_w+padsize(2)*2);
63 padded_img(padsize(1)+1:img_h+padsize(1), ...
64             padsize(2)+1:img_w+padsize(2)) = I;
65 [padded_h,padded_w] = size(padded_img);
66 for i=1:img_h
67     for j=1:img_w
68         copy(i+padsize(1),j+padsize(2)) = sum(sum( ...
69             padded_img(i:i+hsize(1)-1,j:j+hsize(2)-1).*h));
70     end
71 end
72 res = uint8(copy(1+padsize(1):padded_h-padsize(1), ...
73                1+padsize(2):padded_w-padsize(2)));
74 end
```

We can add zero padding to original image

Res:

After my filter function with Sobel kernel



d.

The most significant target(background) has been moved to the medium position.

481 Students.

```
14 % local enhancement approach
15 ROI = [100,300,100,300];
16 loc_he = I;
17 loc_he(ROI(1):ROI(2),ROI(3):ROI(4)) = histeq(I(ROI(1):ROI(2),ROI(3):ROI(4)));
18 imshow(loc_he), title("Local Enhanced Image");
```

Res:

Local Enhanced Image



Filtering

b.

```
26 % b
27 % Prewitt filter
28 aft_pre = imfilter(I,fspecial('prewitt'));
29 imshow(aft_pre), title("After build in prewitt filter");
30 aft_pre_my = myFilter(I,fspecial('prewitt'));
31 imshow(aft_pre_my), title("After my prewitt filter");
32 % Sobel filter
33 aft_sob = imfilter(I,fspecial('sobel'));
34 imshow(aft_sob), title("After build in Sobel filter");
35 aft_sob_my = myFilter(I,fspecial('sobel'));
36 imshow(aft_sob_my), title("After my Sobel filter");
37 % Point filter
38 point_f = [1 1 1;-8 1 1;1 1 1];
39 aft_poi = imfilter(I,point_f);
40 imshow(aft_poi), title("After build in Point filter");
41 aft_poi_my = myFilter(I,point_f);
42 imshow(aft_poi_my), title("After my Point filter");
43 % Blurring filter
44 aft_blu = imfilter(I,fspecial('gaussian',5,1));
45 imshow(aft_blu), title("After build in Blurring filter");
46 aft_blu_my = myFilter(I,fspecial('gaussian',5,1));
47 imshow(aft_blu_my), title("After my Blurring filter");
48
```

Res:

After build in prewitt filter



After my Sobel filter



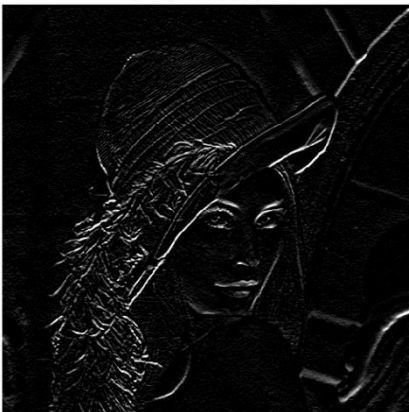
After my prewitt filter



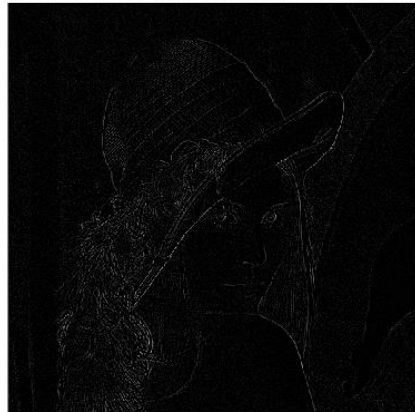
After build in Point filter



After build in Sobel filter



After my Point filter



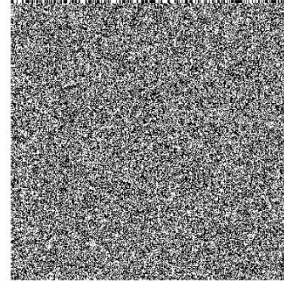
After build in Blurring filter



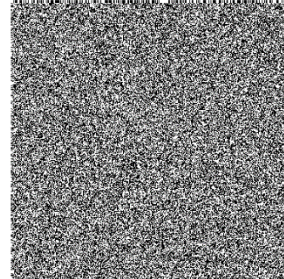
After my Blurring filter



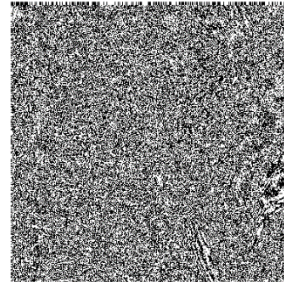
Bit plane 1



Bit plane 2



Bit plane 3



Bit plane 4



Bit plane 5



Those are almost same

Bit Plane Splicing

```
48 % Bit Plane Splicing
49 I = im2uint8(rgb2gray(imread("lena_std.tif")));
50 for i=1:8
51     B=logical(bitget(I,i));
52     imshow(B);title("Bit plane "+i);
53     drawnow;
54     pause(0.5);
55 end
```

Res:

Bit plane 6



Bit plane 8



Bit plane 7



```

% Histogram Equalization
% a
I = im2uint8(rgb2gray(imread("lena_std.tif")));
imshow(I), title("Original Image");

% b
imhist(I), title("Histogram of Original Image");

% c
img_he = histeq(I);
imhist(img_he), title("Histogram of enhanced Image");
imshow(img_he), title("Enhanced Image");

% local enhancement approach
ROI = [100,300,100,300];
loc_he = I;
loc_he(ROI(1):ROI(2),ROI(3):ROI(4)) = histeq(I(ROI(1):ROI(2),ROI(3):ROI(4)));
imshow(loc_he), title("Local Enhanced Image");

% Filtering
% a
h = fspecial('sobel');
aft_my_fil = myFilter(I,h);
imshow(aft_my_fil), title("After my filter function with Sobel kernel");

% b
% Prewitt filter
aft_pre = imfilter(I,fspecial('prewitt'));
imshow(aft_pre), title("After build in prewitt filter");
aft_pre_my = myFilter(I,fspecial('prewitt'));
imshow(aft_pre_my), title("After my prewitt filter");
% Sobel filter
aft_sob = imfilter(I,fspecial('sobel'));
imshow(aft_sob), title("After build in Sobel filter");
aft_sob_my = myFilter(I,fspecial('sobel'));
imshow(aft_sob_my), title("After my Sobel filter");
% Point filter
point_f = [1 1 1;1 -8 1;1 1 1];
aft_poi = imfilter(I,point_f);
imshow(aft_poi), title("After build in Point filter");
aft_poi_my = myFilter(I,point_f);
imshow(aft_poi_my), title("After my Point filter");
% Blurring filter

```

```

aft_blu = imfilter(I,fspecial('gaussian',5,1));
imshow(aft_blu), title("After build in Blurring filter");
aft_blu_my = myFilter(I,fspecial('gaussian',5,1));
imshow(aft_blu_my), title("After my Blurring filter");

```

% Bit Plane Splicing

```

I = im2uint8(rgb2gray(imread("lena_std.tif")));
for i=1:8
    B=logical(bitget(I,i));
    imshow(B);title("Bit plane "+i);
    drawnow;
    pause(0.5);
end

```

```

function res = myFilter(I,h)
    % We can add zero padding to original image
    [img_h,img_w] = size(I);
    hsize = size(h);
    padsize = [((hsize(1)-1)/2),((hsize(2)-1)/2)];
    padded_img = zeros(img_h+padsize(1)*2,img_w+padsize(2)*2);
    padded_img(padsize(1)+1:img_h+padsize(1), ...
        padsize(2)+1:img_w+padsize(2)) = I;
    [padded_h,padded_w] = size(padded_img);
    for i=1:img_h
        for j=1:img_w
            copy(i+padsize(1),j+padsize(2)) = sum(sum( ...
                padded_img(i:i+hsize(1)-1,j:j+hsize(2)-1).*h));
        end
    end
    res = uint8(copy(1+padsize(1):padded_h-padsize(1), ...
        1+padsize(2):padded_w-padsize(2)));
end

```