 

**School of Information Science and Engineering**

**2019－2020 school year**

Digital Image Processing Experiment Report

Course Name： Digital Image Processing

Title of Experiment： Image Display and Format Transformation

Major and Class Class 3, Communication Engineering

Student ID 201600121116

Studeng Name 张子超

Date 2019/4/7

# 1． Objectives:

1. Get start with some filters in the MATLAB tool box.

2. Handle the algorithm that eliminates image noise with filter.

3. Learn the spatial transformation of image.

# 2． Experiment Content:

Practice Matlab commands for image enhancement and get familiar with module functions listed below:

**Image enhancement:**imadjust - Adjust image intensity values or colormap  
Image noising:  
imnoise - Add noise to an image  
**Noise estimation:**Roipoly, histroi, statmoments  
**Image filtering:**medfilt2 - Perform 2-D median filtering  
ordfilt2 - Perform 2-D order-statistic filtering  
spfilt  
wiener2 - Perform 2-D adaptive noise-removal filtering  
**Test image making:**maketform， imtransform

# 3． Experiment Principle:

## 1. Adjust the visual effects of the image.

Using function imadjust to improve the visual effect of the image. The function requires input of maximum and minimum of the image, so I use max(max(image)) to find them. After this, the method returns a value in uint8, it needs to be converted to double, so plus the command im2double is OK.



## 2. Generate an image with noise.

|  |  |  |
| --- | --- | --- |
|  | Imnoise | Imnoise2 |
| Var-  iance  =0.1 |  |  |
| Var-  iance  =0.3 |  |  |
| Var-  iance  =0.5 |  |  |
| Var-  iance  =0.9 |  |  |

The imnoise2 creates pure periodic noise without adding it into the image, and it seems rather weak in low variance.

## 3. Noise estimation

From the region of interest, matlab extract pixel imformation about pixel distribution. The shape of its distribution seems alike the gaussian noise, so we can use filter specific for gaussian noise to restore the image.





## 4. Load image cameraman.tif, add gaussian, salt&pepper, speckle noise separately and compare the influences caused by mean and variance.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Gaussian | Salt & Pepper | Speckle |
| Var-iance  =0.1 |  |  |  |
| Var-iance  =0.3 |  |  |  |
| Var-iance  =0.5 |  |  |  |
| Var-iance  =0.9 |  |  |  |

## 5. Use function medfilt2, ordfilt2 and wiener2 to remove noise add into the image separately and compare the processed image with origin image, figure out the effect of each method.

Medfilt2

 

Ordfilt2

 

Wiener2



The higher variance is, the more unclear image will be. We can see that Wiener filtering can restore the figure best and ordfilt2 just made the image very dizzy while the medfilt2 has the worst result.

6. Design a notch filter



With reference to the mathematical equation and some content from the text book, I finally got a figure like this.

## 7. Create a transformed image with the checkerboard pic in specified shape.



The affine transformation is done by matrix multiplication.

## 8. Study on the relationship between spatial filtering methods and the characters of noise, and demonstrate your conclusion with examples.

Firstly, load the image and add some noise into it:

imnoise(y,'gaussian',0,0.3);

we choose gaussian noise here and we know the noise follows a normal distribution. In this case, arithmetical mean filtering can’t be truly helpful. Set the window size to 3×3 and use the function spfilt to accomplish a filtering. In order to compare between results of different filtering methods, also use min and max filter.



Figure added gaussian noise After the arithmetic filtering

 

After the ‘max’ filtering After the ‘min’ filtering

The result is rather contrasting, the aritmetic filtering seems blurred the figure instead of reconstrructing it. And the min and max filtering messed up the whole figure.

 

Figure added salt & pepper noise After the arithmetic filtering



After the ‘max’ filtering After the ‘min’ filtering

In order to see the advantages of arithmetic mean filtering, add ‘salt & pepper’ noise into the figure. The outputs are convincing, that mean filtering almost recovered the picture, with a little sign of noise, but, still better than the others.

# 4． Conclusions and Experiences:

I learned much from this experiment, and I also know without a firm math base, there will be no chance to get deeper in this subject. Almost all important rules in digital signal processing have been applied in this experiment, and with two dimensions, which made it harder to learn, we got to analyse both x and y. Noises can be removed if it’s type is well and processed with proper filter. Affine transformation can be easily explained as the multiplication of matrix. In fact, adjusting several parameters can cause a great difference in output.

# 5． Appendix（Code）:

## Mission 1:

[y,mapy]=imread('G:\desktop documents\2018´óÈýÏÂ\Êý×ÖÍ¼Ïñ´¦Àí×ÊÁÏFor students\For students\³ÌÐòÓëÍ¼Ïñ\Í¼Ïñ¿â\Fig0704(Vase).tif');

minimum=im2double(min(min(y)))

maximum=im2double(max(max(y)))

j=imadjust(y,[minimum,maximum],[]);

figure

imshow(j)

## Mission 2:

gn1=imnoise(y,'gaussian',0,0.1)

figure

imshow(gn1)

set(gca,'position',[0 0 1 1])

gn2=imnoise(y,'gaussian',0,0.3)

figure

imshow(gn2)

set(gca,'position',[0 0 1 1])

gn3=imnoise(y,'gaussian',0,0.6)

figure

imshow(gn3)

set(gca,'position',[0 0 1 1])

gn4=imnoise(y,'gaussian',0,0.9)

figure

imshow(gn4)

set(gca,'position',[0 0 1 1])

n1=imnoise2('gaussian',length(y(1,:)),length(y(:,1)),0,0.1);

y1=im2double(y);

ypn1=n1+y1;

imshow(ypn1)

set(gca,'position',[0 0 1 1])

## Mission 3:

n1=imnoise2('gaussian',length(y(1,:)),length(y(:,1)),147,20);

n1p=n1/max(n1(:))

y1=im2double(y);

gn1=n1p+y1;

[B, c, r] = roipoly( gn1 );

[p, npix] = histroi(gn1, c, r);

figure, bar(p, 1)

[v , unv] = statmoments( p, 2)

n2=imnoise2('gaussian',npix,1,floor(unv(1)),20);

figure

hist(n2,128)

axis([0 250 0 200])

## Mission 4:

yn1=imnoise(y,'gaussian',0,0.1);

figure

imshow(yn1)

set(gca,'position',[0 0 1 1])

yn2=imnoise(y,'salt & pepper',0.1);

figure

imshow(yn2)

set(gca,'position',[0 0 1 1])

yn3=imnoise(y,'speckle',0.1);

figure

imshow(yn3)

set(gca,'position',[0 0 1 1])

## Mission 5:

yn2=imnoise(y,'salt & pepper',0.5);

figure

imshow(yn2)

set(gca,'position',[0 0 1 1])

f1 = medfilt2(yn2, [7 7], 'symmetric');

figure

imshow(f1)

set(gca,'position',[0 0 1 1])

f2=ordfilt2(y,25,ones(7))

figure

imshow(f2)

set(gca,'position',[0 0 1 1])

f3=wiener2(y,[7 7])

figure

imshow(f3)

set(gca,'position',[0 0 1 1])

## Mission 6:

[M N]=size(y);

C1=[56 77];

u1=single(-C1(1):M-1-C1(1));

v1=single(-C1(2):N-1-C1(2));

idx1=find(u1>M/2);

u1(idx1)=u1(idx1)-M;

idy1=find(v1>N/2);

v1(idy1)=v1(idy1)-N;

[V1, U1]=meshgrid(v1,u1);

D1=hypot(U1,V1);

u2=single(C1(1):M-1+C1(1));

v2=single(C1(2):N-1+C1(2));

idx2=find(u2>M/2);

u2(idx2)=u2(idx2)-M;

idy2=find(v2>N/2);

v2(idy2)=v2(idy2)-N;

[V2, U2]=meshgrid(v2,u2);

D2=hypot(U2,V2);

H=1./(1+(D0\*D0./(D1.\*D2)).^n);

Hsh=fftshift(H);

mesh(Hsh)

## Mission 7:

yn=imnoise(y,'salt & pepper',0.02);

figure

imshow(yn)

set(gca,'position',[0 0 1 1])

ynf1=spfilt(yn,'amean',3,3,1.5);

figure

imshow(ynf1)

set(gca,'position',[0 0 1 1])

ynf2=spfilt(yn,'max',3,3,1.5);

figure

imshow(ynf2)

set(gca,'position',[0 0 1 1])

ynf3=spfilt(yn,'min',3,3,1.5);

figure

imshow(ynf3)

set(gca,'position',[0 0 1 1])

## Mission 8:

C=checkerboard

imshow(C)

T=[2 1 0; 0 2 0; 0 0 1];

tform=maketform('affine', T);

g = imtransform(C, tform, 'FillValue',0.5)

figure

imshow(g)