SOUTH UNIVERSITY OF SCIENCE AND TECHNOLOGY OF CHINA

IMAGE AND VIDEO PROCESSING

CLASS PROJECT 4

Frequency Domain Filtering

Author:
Jie YANG

Supervisor:
JIANHONG SHI

October 16, 2018

CONTENTS

Frequency Domain Filtering

Introduction

Image filtering is a common used method to separate different component from original image. To do this work, it is difficult in spatial domain. In spatial domain, threshold and gradient is common used method, however, these method only consider pixel individually or several pixels around it. When we transfer image into frequency domain, we can divide it into low frequency and high frequency part, so we can easy to do filter and then reconstruct what we want. In frequency domain, multiply equals to convolution in spatial domain, so that this filtering has global influence.

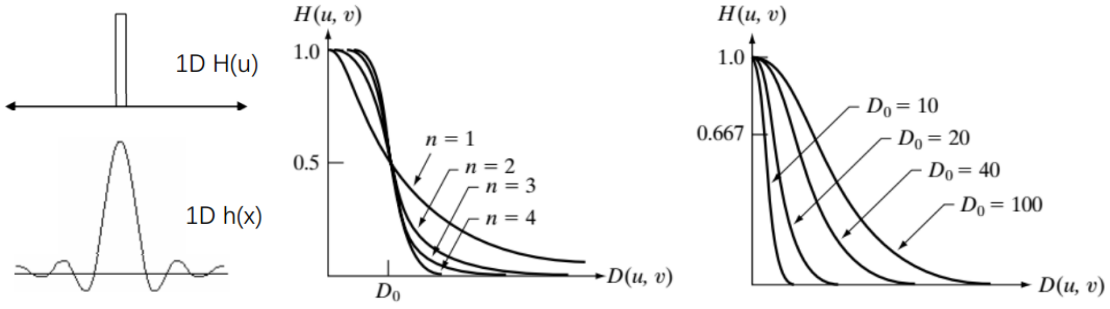
In frequency domain, we should select proper filters, base on cut off frequency, we can divide filers into low pass filter, high pass filter and band pass filter. For low pass filter we want it can let low frequency pass while block higher frequency. However, this ideal filter does not exist in nature, so we design use some mathematical expressions to approach it,like Butterworth filter.

In this experiment, we will construct different filter and learn how to use then to filter image in frequency domain. Finally, high pass filter will be used to sharp image.

Method

Filter is the transfer function in frequency domain. Image can be transform to frequency domain, then multiply with the filter select what we want. For ideal low pass filter.**Eq.1** D_0 is the cut off frequency, it only pass frequency component which lower that D_0. In this report, we set $\frac{D_0}{D}$ as cut off frequency. But for Gibson effect, when we do reverse FFT to reconstruction, the ideal low pass filter (ILPF)whill induce ringing artifacts in image.**Fig.1**

$$H(u,v) = \begin{cases} 1 & if D(u,v) \le D_0 \\ 0 & if D(u,v) > D_0 \end{cases} \tag{1}$$



(a) Ideal low pass filter (b) Butterworth filter (c) Gaussian filter

Figure 1: Low pass filters

Butterworth filter (GLPF) is a wildly used low pass filter, when D_0 increase, it scut off frequency also increase, n is the order of it.**Eq.2** This polynomial filter can recede the ringing artifacts, for it has tails after cut off frequency.**Fig.1** But as n increase, the ringing artifacts will increase.

$$H(u,v) = \frac{1}{1+[\frac{D(u,v)}{D_0}]^{2n}} \tag{2}$$

Gaussian filter (gLPF) is another common used low pass filter, it use Guassian function as transfer function,**Eq.3**, it can remove ringing artifacts perfectly in reconstruct image. **Fig.1**

$$H(u,v) = e^{-D^2(u,v)/2D_0^2} \tag{3}$$

High frequency filter is opposite to low frequency filter, it will block low frequency component. So that we can revers low pass filter to construct high frequency filter.**Eq.4**

$$H(u,v) = a - H_{LPF} \tag{4}$$

To do frequency filtering, we can multiply the filter to get what we want in frequency,**Eq.5** it is more easy to do in frequency domain. Because, in spatial domain, we should do convolution.

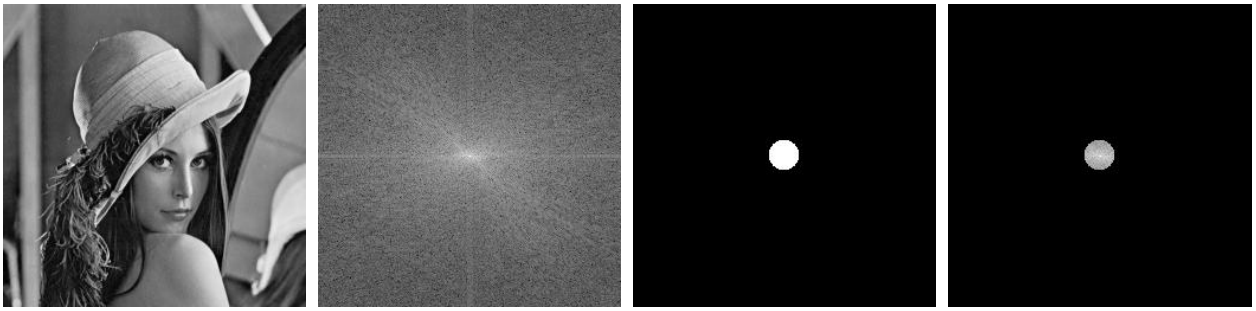
$$G(u,v) = H(u,v)F(u,v) \tag{5}$$

Results

Filter

Lena

We use ILPF, BLPF and GLPF to filter lena image and set $D_0 = 0.1, 0.3, 0.6, 0.9$, respectively. The use filtered to reconstruction. The filter process in indicate in **Fig.2**, we first do FFT to lena image, to transfer it into frequency domain and set $H(x,y)$ base on D_0. Then multiply $F(u,v)$ with $H(x,y)$, finally do IFFT to reconstruct image.



(a) original (b) $F(u,v)$ (c) $H(x,y)$ (d) $H(x,y) * F(u,v)$

Figure 2: ILPF filtering process of lena image with $D_0 = 0.1$

The reconstruct image is indicated in **Fig.3**. It is clear to see that with low cut off frequency, the image is more blur. Compare to BLPF and GLPF, ILPF has clear ringing artifacts in reconstruction,especially in low cut off frequency situation. When $D_0 > 0.6$, there is no remarkable difference for three filters.

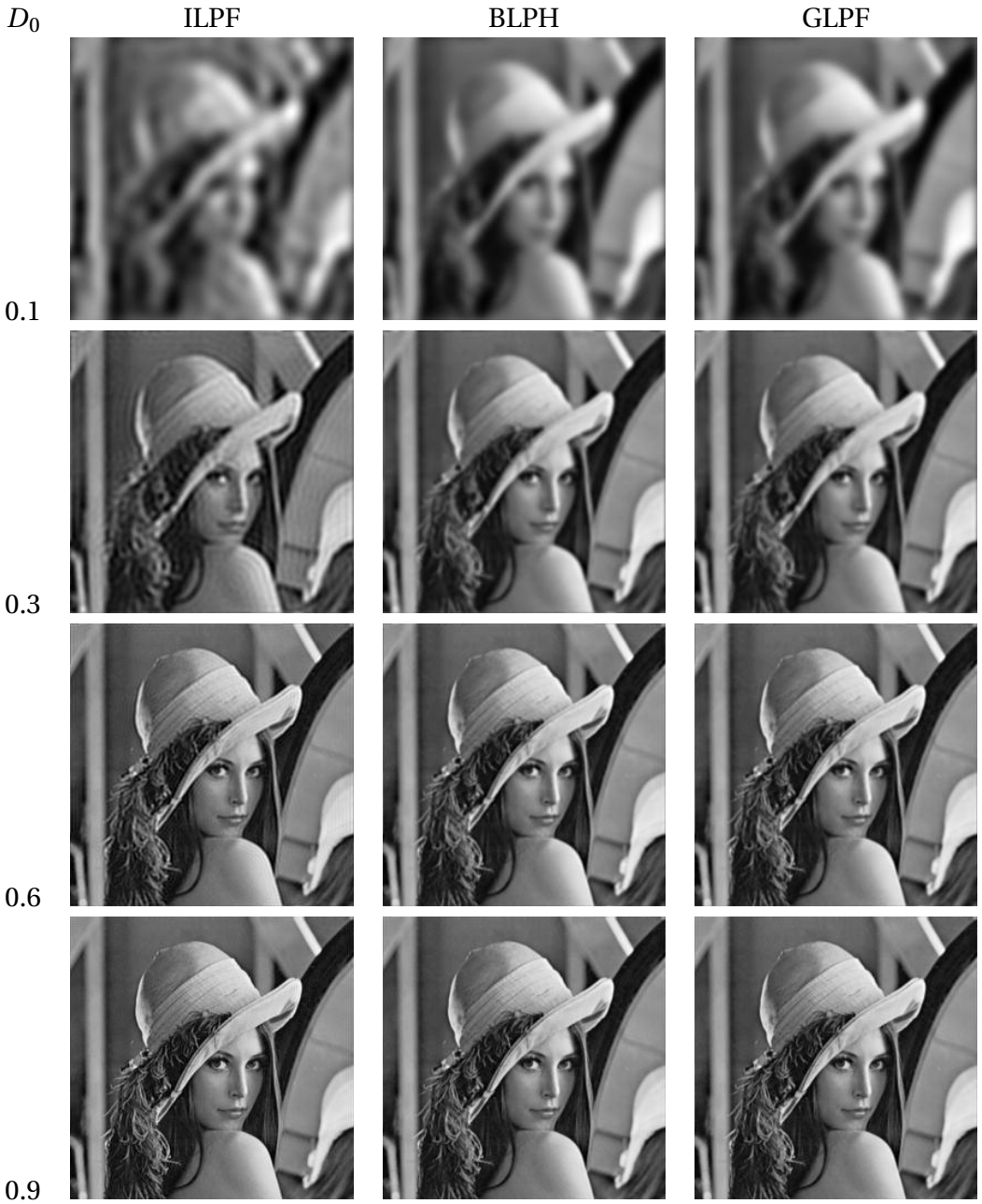
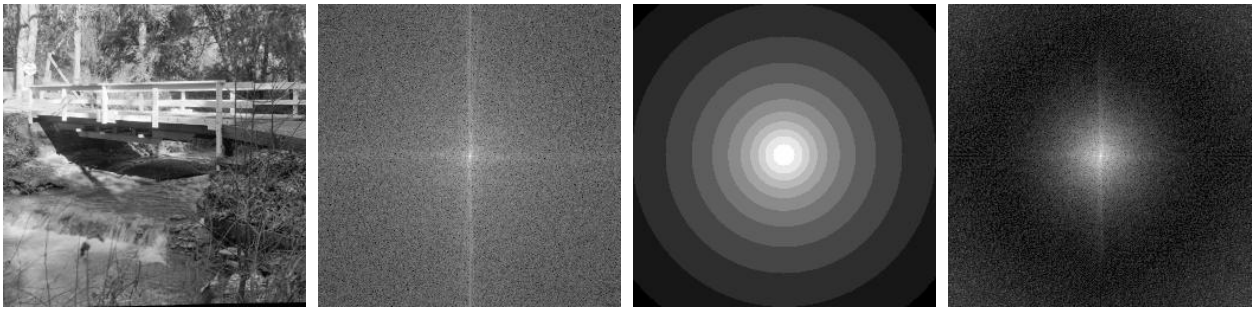


Figure 3: filtered lena image with different low pass filters and D_0

Bridge

We also test these filters in bridge image.The filter process in indicate in **Fig.4**. Compare to ILPF, GLPF also has weight in higher frequency than D_0, so that with same D_0, GLPF has high

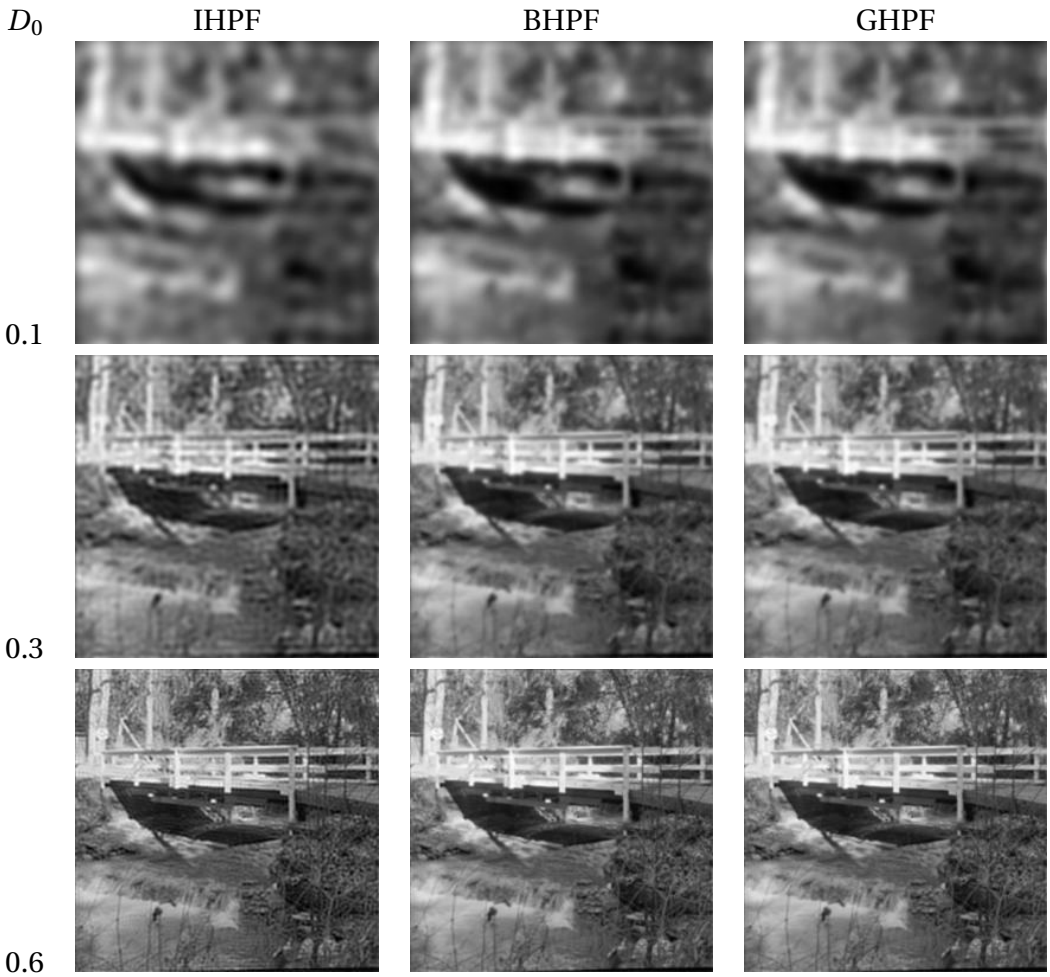
frequency component.



(a) original (b) $F(u,v)$ (c) $H(x,y)$ (d) $H(x,y) * F(u,v)$

Figure 4: GLPF filtering process of bridge image with $D_0 = 0.1$

The reconstruct image is indicated in **Fig.5**. We can still find low D_0 cause blur image, for low cut cut off frequency loss too much detail information of image. The ringing artifacts effect in bridge is not clear that lena image, the reason may be that bridge image has more complex background, which let ringing artifacts is difficult to be seen.



0.9

Figure 5: filtered bridge image with different low pass filters and D_0

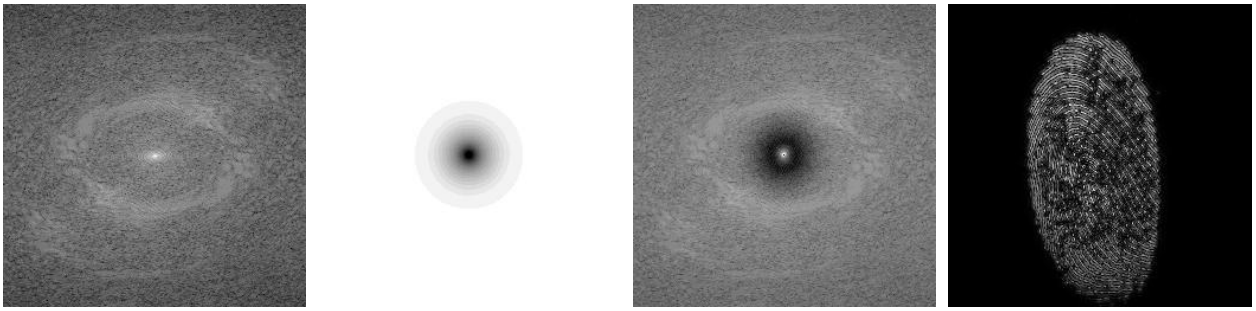
Fingerprint sharpen

After above exercise, we have already know how to filter image in frequency domain, now will use high pass filter to extract high frequency of image.

We choose two fingerprint images.**Fig.6**, then do FFT to it and filter with high pass filter, which generate base on **Eq.4**. Finally use a threshold to binarize itto get sharped image. **Fig.7**

(a) fingerprint 1 (b) fingerprint 2

Figure 6: Original image of fingerprint images



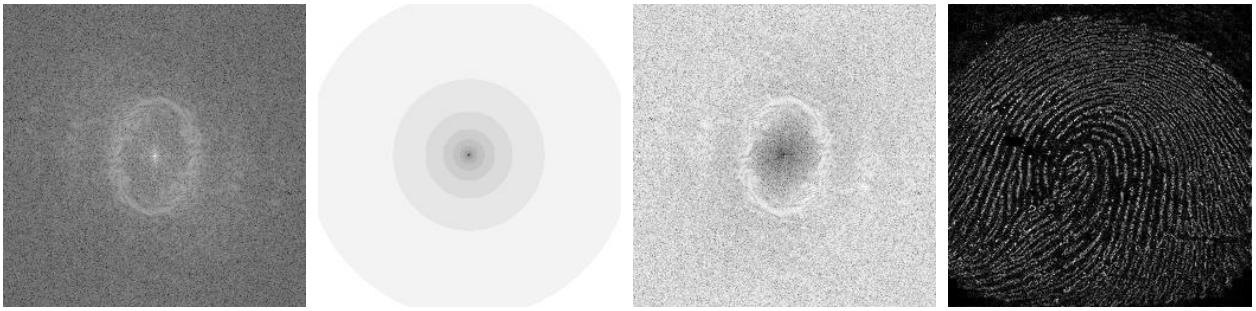
(a) $F(u,v)$ (b) $H(x,y)$ (c) $H(x,y) * F(u,v)$ (d) filtered

Figure 7: BHPF filtered bridge image with $D_0 = 0.3$

Fig.8 is the filtered image of GHPF with $D_0 = 0.6$, it seem to see that the filtered image be clear that original image, because it remove smug in original image which in low frequency part.

But for **Fig.7**, the filtered fingerprint i image, there still are several smug in image, to remove

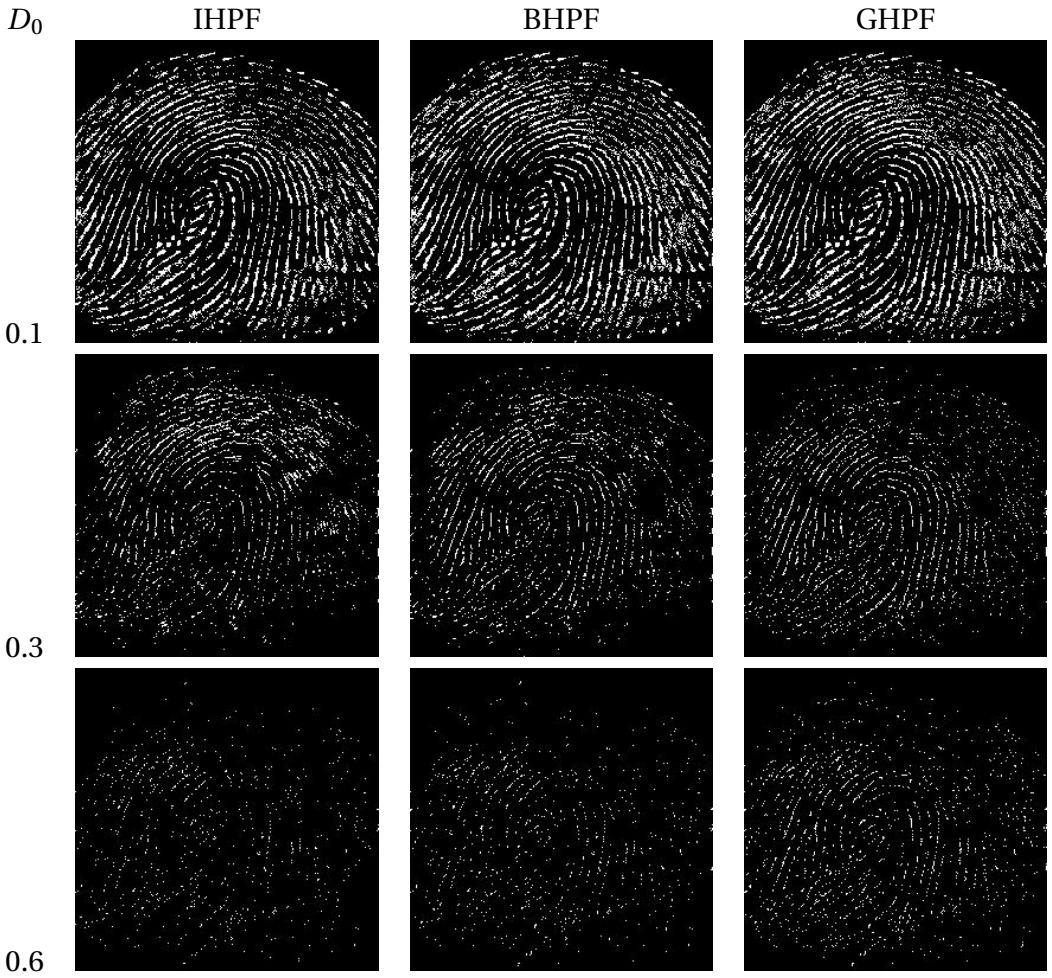
these smug, we can use threshold to ignore it.

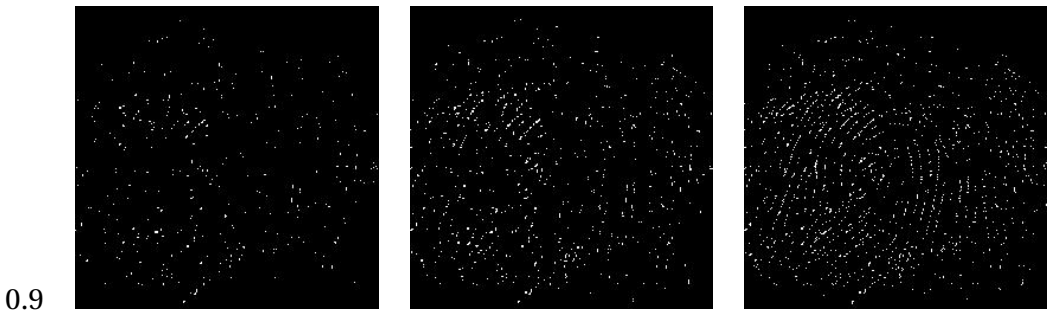


(a) $F(u,v)$ (b) $H(x,y)$ (c) $H(x,y) * F(u,v)$ (d) filtered

Figure 8: GHPF filtered lena image with $D_0 = 0.6$

Fig.9 is the sharpen image of fingerprint 1 image with different image and cut off frequency. It is clear to see that, lower cutoff frequency will make greater performance, for when $D_0 > 0.1$ have already remove mian smug in low frequency, increase cut off frequency will remove more information which let reconstruction be more difficult.

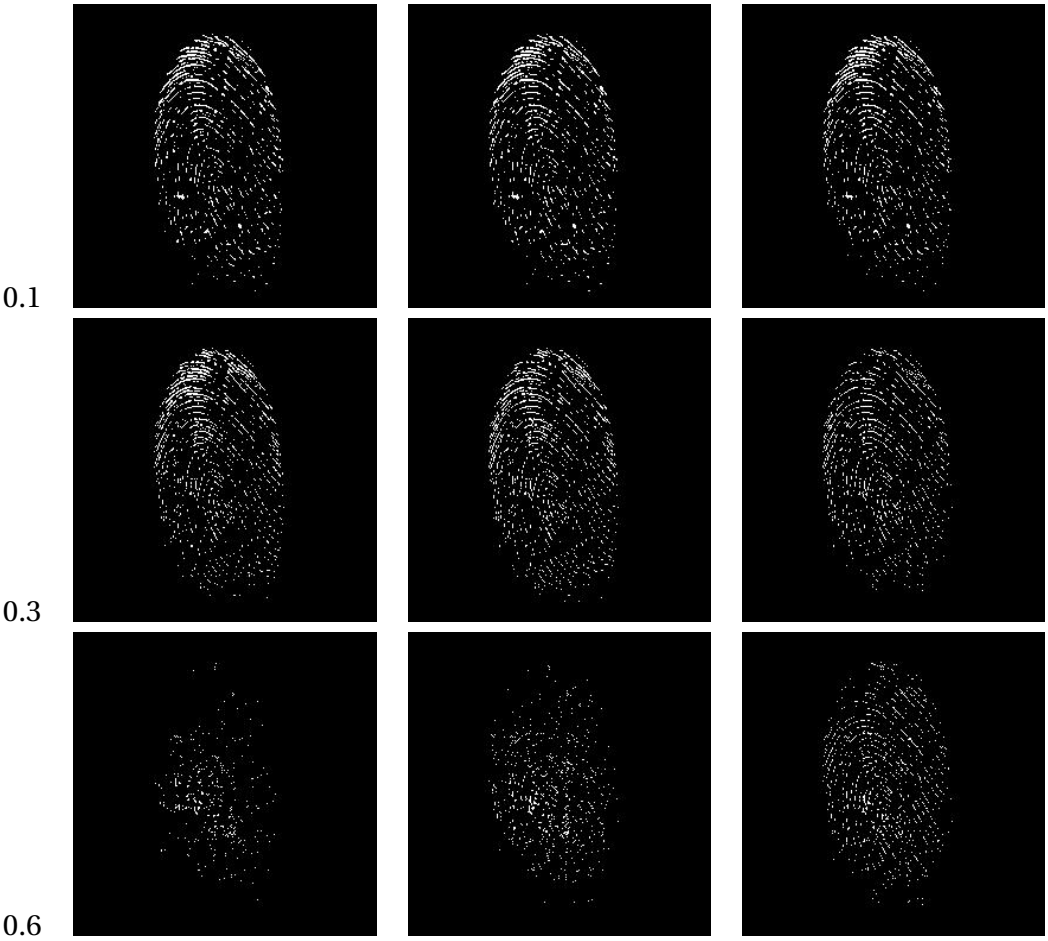


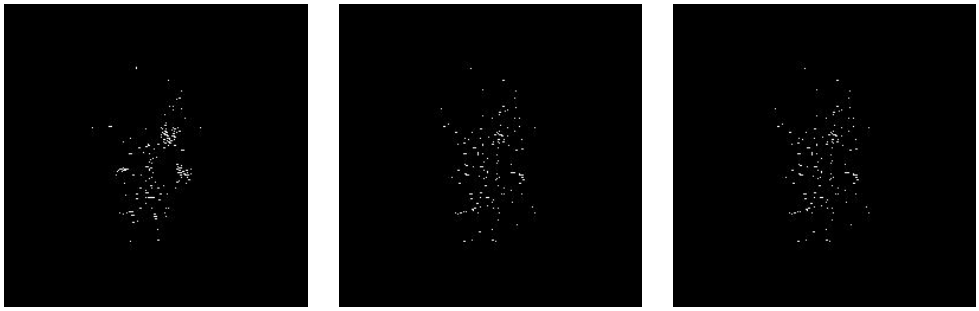


0.9

Figure 9: Fingerprint 1 image sharpen with different filters and D_0

Fig.10 is the sharpen image of fingerprint 2 image with different image and cut off frequency. It is clear to see that only when $D_0 \leq 0.3$, the result is acceptabl





0.9

Figure 10: Fingerprint 2 image sharpen with different filters and D_0

Discussion

In this experiment, we can find that although ideal filter has best frequency cut off performance, it will cause artifact effect. In practical, Gaussain or Butterworth filter has better performance. At the same time, we should select proper cut off frequency, bad cut off frequency will loss too much information, which will cause problem in reconstruction.

Supplementary

This is the code used in this project.

```
1
2  close all
3  clear
4  clc
5  %%
6  imapath='D:\graduated\Image_process\lab\PGM_images\';
7  savepath='D:\graduated\Image_process\lab\lab_report\lab5\matlab2\';
8  namelist={ 'bridge', 'lena'};
9  Doset=[ 0.1, 0.3, 0.6, 0.9 ];
10 for i=1:length(namelist)
11     name=namelist{i};
12     imgpath=[imapath,name,'.pgm'];
13     ima=imread(imgpath);
14     ima=ima(:,:,1);
15     ima=double(ima);
16     imwrite(disima(ima),[savepath,name,'_original','.jpg'])
17
18
19      fimg= mydft2(ima);
20     for k=1:length(Doset)
21         Do=Doset(k);
22         HIL=ILPF(Do,ima);
```

```
            HBL=BLPF(Do,2,ima);
            HGL=GLPF(Do,ima);

            ifimg=FF(HIL,fimg,'_ILPF',Do,name);
            ifimg=FF(HBL,fimg,'_BLPF',Do,name);
            ifimg=FF(HBL,fimg,'_GLPF',Do,name);
        end

end

namelist={ 'fingerprint1', 'fingerprint2'};

for i=1:length(namelist)
    name=namelist{i};
    imgpath=[imapath,name,'.pgm'];
    ima=imread(imgpath);
    ima=ima(:,:,1);
    ima=double(ima);
    imwrite(disima(ima),[savepath,name,'_original','.jpg'])


    fimg= mydft2(ima);
    for k=1:length(Doset)
        Do=Doset(k);
        HPFI(fimg,name,Do,ima);
        HPFB(fimg,name,Do,ima);
        HPFG(fimg,name,Do,ima);

    end



end


function ifimg=FF(H,fimg,mark,Do,name)
cot=['_',num2str(Do*10)];
savepath='D:\graduated\Image_process\lab\lab_report\lab5\matlab2\';
imwrite(disima(log(H*1000000000+1)),[savepath,name,mark,cot,'_H','.jpg'])
imwrite(disima(log(fimg)),[savepath,name,mark,cot,'_F_log','.jpg'])
F=fimg.*H;
imwrite(disima(log(F)),[savepath,name,mark,cot,'_HF_log','.jpg'])

ifimg=myidft2(F);
imwrite(disima(real(ifimg)),[savepath,name,mark,cot,'.jpg'])
end

function []=HPsharp(ima,H,fimg,mark,Do,name)
cot=['_',num2str(Do*10)];
savepath='D:\graduated\Image_process\lab\lab_report\lab5\matlab2\';
```

```
ifimg=FF(H,fimg,mark,Do,name);

sima=real(ifimg);%


sharpima=im2bw(disima(sima)) ;
imwrite(disima(sharpima),[savepath,name,mark,'_sharp',cot,'.jpg'])



end

function H=ILPF(Do,ima)
    M=size(ima,1);
    N=size(ima,2);
    H=ones(M);
    r=Do*M/2;
    for x=1:M
        for y=1:N
            u = x - M / 2;
      v = N / 2 - y;
            R=sqrt(u^2+v^2);
            if R>r
                H(x,y)=0;
            end


        end
    end

end

function H=BLPF(Do,n,ima)
    M=size(ima,1);
    N=size(ima,2);
    H=ones(M);
    r=Do*M/2;
    for x=1:M
        for y=1:N
            u = x - M / 2;
      v = N / 2 - y;
            R=sqrt(u^2+v^2);

            H(x,y)=1/(1+(R/r)^(2*n));

        end
    end

end


```

```
function H=GLPF(Do,ima)
    M=size(ima,1);
    N=size(ima,2);
    H=ones(M);
    r=Do*M/2;
    for x=1:M
        for y=1:N
            u = x - M / 2;
      v = N / 2 - y;
            R=sqrt(u^2+v^2);
            a=0.5*(R/r)^2;
            H(x,y)=1/exp(a);

        end
    end

end




function H=HPFG(fimg,name,Do,ima)
H=-1*GLPF(Do,ima);
H=H-min(min(H));
H=1-GLPF(Do,ima);
HPsharp(ima,H,fimg,'_HPFG',Do,name);
end

function H=HPFI(fimg,name,Do,ima)
H=-1*ILPF(Do,ima);
H=H-min(min(H));
H=1-ILPF(Do,ima);
HPsharp(ima,H,fimg,'_HPFI',Do,name);
end

function H=HPFB(fimg,name,Do,ima)
H=-1*BLPF(Do,4,ima);
H=H-min(min(H));
H=1-BLPF(Do,4,ima);
HPsharp(ima,H,fimg,'_HPFB',Do,name);
end

function dima= disima(ima)

ima=round(abs(ima));
ima(ima==inf)=0;
maxL=255;
minL=0;
mL=maxL-minL;
maxv=max(max(ima));
minv=min(min(ima));
```

```
L=maxv-minv;
dima=(ima-minv)*mL/L+minL;
%dima=histeq(dima);
dima=uint8(dima);

end



function myfima= mydft2(I)

M=size(I,1);
N=size(I,2);
myfima=zeros(M,N);
for u=0:M-1
    for v=0:N-1
        temp=0;
        for x=0:M-1
            for y=0:N-1
            temp=temp+I(x+1,y+1)*((-1)^(x+y))*exp(-1i*2*pi*(u*x/M+v*y/N));%*((-1)...
    ^(x+y))
            end
        end

        myfima(u+1,v+1)=temp;

    end
end

end

function myfima= myidft2(I)

M=size(I,1);
N=size(I,2);
myfima=zeros(M,N);
for x=0:M-1
    for y=0:N-1
        temp=0;
        for u=0:M-1
            for v=0:N-1
                temp=temp+I(u+1,v+1)*exp(1i*2*pi*(u*x/M+v*y/N));%*((-1)^(x+y))

            end
        end
        myfima(x+1,y+1)=real(temp)/(M*N)*((-1)^(x+y));

    end
end

```

```
226 end
227 }
```

MATLAB code for image processing