بهبودی تصویر توسط تبدیل فوریه

پروژه پنجم

علی رضاقلی زاده ٦١٠٣٨٩١٢٦

۱. تبدیل فوریه

با اعمال رابطه ی (۱-۱) بر روی تصویر دامنه ی تصویر را به بعدی دیگر می بریم .

$$F(u,v)=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(ux/M+vy/N)} \qquad (۱-۱)$$

که $f(x,y)$ مقدار پیکسل x و y ام در ماتریس M×N تصویراست و $F(u,v)$ حاصل اعمال فوریه روی تصویر است .

می توان این تبدیل را به صورت دیگر و با استفاده از ماسک متناظر آن که در **شکل ۱** نشان داده شده واعمال این ماسک روی تصویر، مطابق رابطه (۲-۱) نیز بدست آورد .

$$\mathbf{F}_N = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N & W_N^2 & \cdots & W_N^{N-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & W_N^{N-1} & W_N^{2(N-1)} & \cdots & W_N^{(N-1)^2} \end{bmatrix}$$

شکل ۱

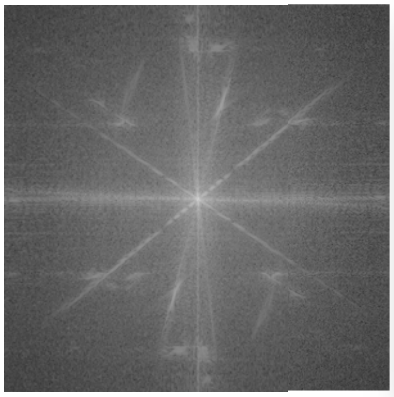
$$\tilde{\mathbf{X}} = \mathbf{F}_N \mathbf{X} \mathbf{F}_N \qquad (۱-۲)$$

پیش تر در پروژه ها ،با فیلتر های مختلف آشنا شدیم و به این صورت استفاده می شد که آنرا روی ماتریس تصویر حرکت می دادیم ، حال اگر بخواهیم همین کار را با استفاده از ماتریس فوریه شده ی تصویر (که حوزه ی تصویر را به حوزه ی دیگر به نام فرکانس برده است) روی تصویر اعمال کنیم باید در این ماتریس فوریه شده ضرب کنیم (رابطه ی ۳-۱).

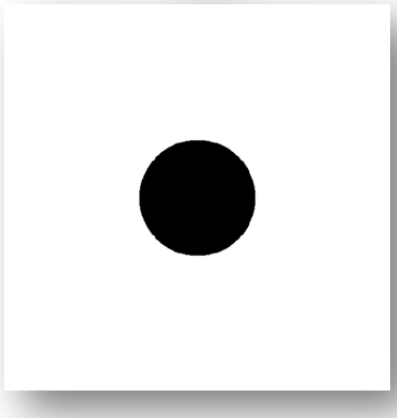
$$G(u,v)= F(u,v)\times H(u,v) \qquad (1\text{-}3)$$

Original Image

centered Of Fourier transformed



HighPass Mask

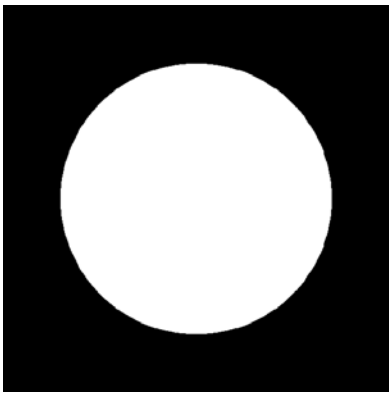


Enhanced Image

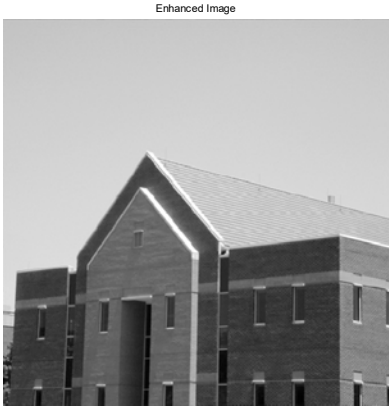
BandPass Mask

Enhanced Image

LowPass Mask



Enhanced Image



۲. ضمیمه

(Code 1) : preprocessing of image that is resized into N×N and every entry multiply into $(-1)^{i+j}$.

```
function preprocessedIm=preprocessor(image)
%first image is resized to become N*N  then we multiple image into (-1)^(i+j)
image=imresize(image,[min(size(image,1),size(image,2)),
min(size(image,1),size(image,2))]);
image=double(image);
for i=1:size(image,1)
    for j=1:size(image,2)
        image(i,j)=image(i,j)*(-1)^(i+j);
    end
end
preprocessedIm=image;
end
```

(Code 2) : Fourier Transform function

```
function FTedIm=FourierTransform(image)
%image is N*N matrix
F=zeros(uint8(size(image,1)));
N=size(image,1);
w=exp((-1*complex(0,1)*2*pi)/N);

for i=1:N
    for j=1:N
        if i==1 || j==1
            F(i,j)=1;
        else
            F(i,j)=w^((i-1)*(j-1));
        end
    end
end

FTedIm=F*image*F;

end
```

(Code 3) : Filter Function that produce zero-one mask

```
function improvedIm=FilterFunction(signal,type,percentage)
  N=size(signal,1);

  if mod(N,2)==0
      N=N-1;
  end
   H=zeros(N);

  u=-floor(N/2):floor(N/2);
  v=-floor(N/2):floor(N/2);
  [U,V]=meshgrid(u,v);


   switch type
       case 'lowpass'
           Dist=sqrt(U.^2+V.^2);
```

```
            J=find(Dist<percentage*N/2);
            H(J)=1;
        case 'highpass'
            Dist=sqrt(U.^2+V.^2);
            J=find(Dist<percentage*N/2);
            H=ones(size(H));
            H(J)=0;
        case 'bandpass'
            Dist=sqrt(U.^2+V.^2);
            J1=find(Dist<percentage(2)*N/2);
             J2=find(Dist<percentage(1)*N/2);
            H(J1)=1;
            H(J2)=0;
    end

    H=imresize(H,size(signal));
    improvedIm=H.*signal;

end
```

(Code 4) : Change domain into image domain

```
function ChDomSignal=InverseFT(signal)
Fs=zeros(size(signal,1));
N=size(signal,1);
w=exp((-1*complex(0,1)*2*pi)/N);

for i=1:N
    for j=1:N
        Fs(i,j)=w^(-1*(i-1)*(j-1));
    end
end

ChDomSignal=(1/(N^2))*(Fs*signal*Fs);
end
```

(Code 5) : Every entries multiply into $(-1)^{i+j}$

```
function ImDom=PostProcessing(image)
for i=1:size(image,1)
    for j=1:size(image,2)
        image(i,j)=image(i,j)*((-1)^(i+j));
    end
end
ImDom=image;
end
```