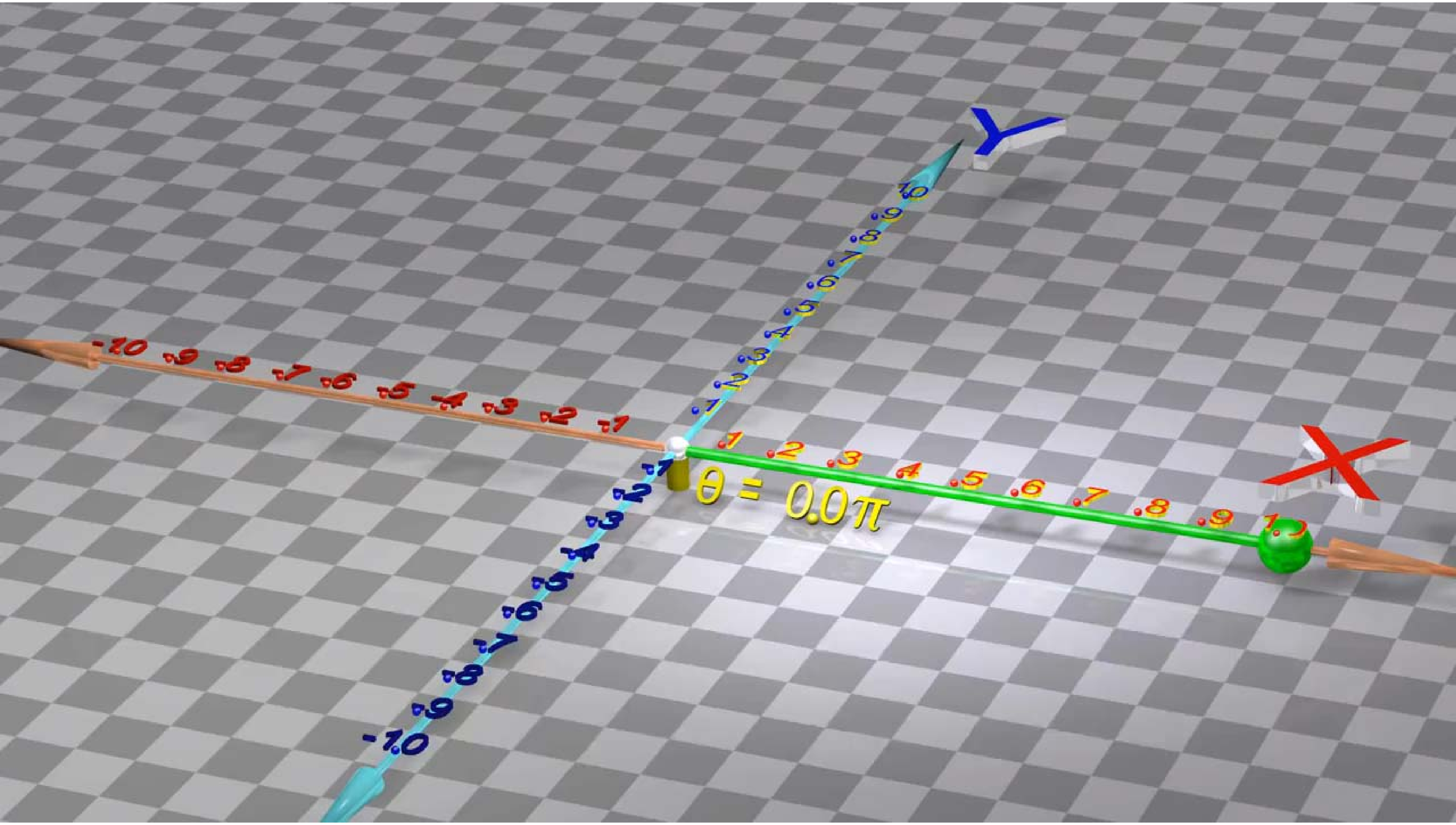
Image Enhancement in the Frequency Domain (Pattern Recognition WK3)

Theekapun Charoenpong



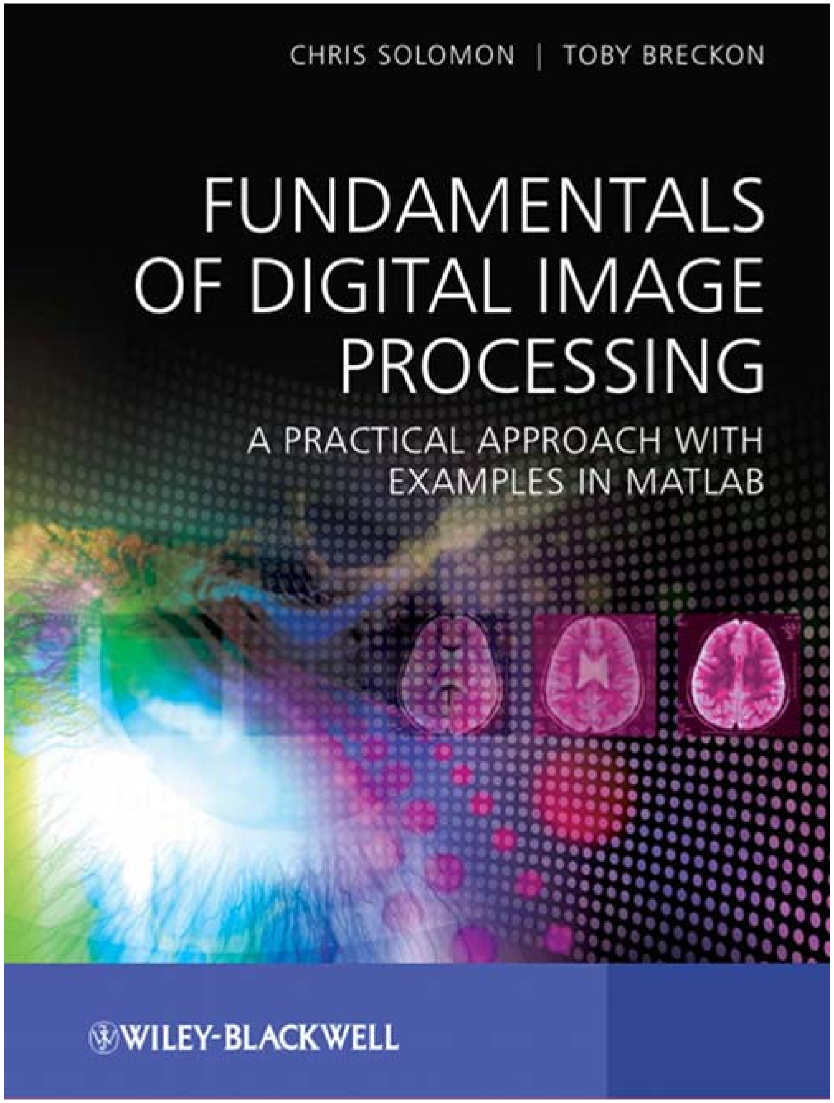
Y
X
θ = 0.0π

Discrete Fourier Transform 2D (DFT)

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

$f(x,y)$ is a function representing an image with size $M \times N$.

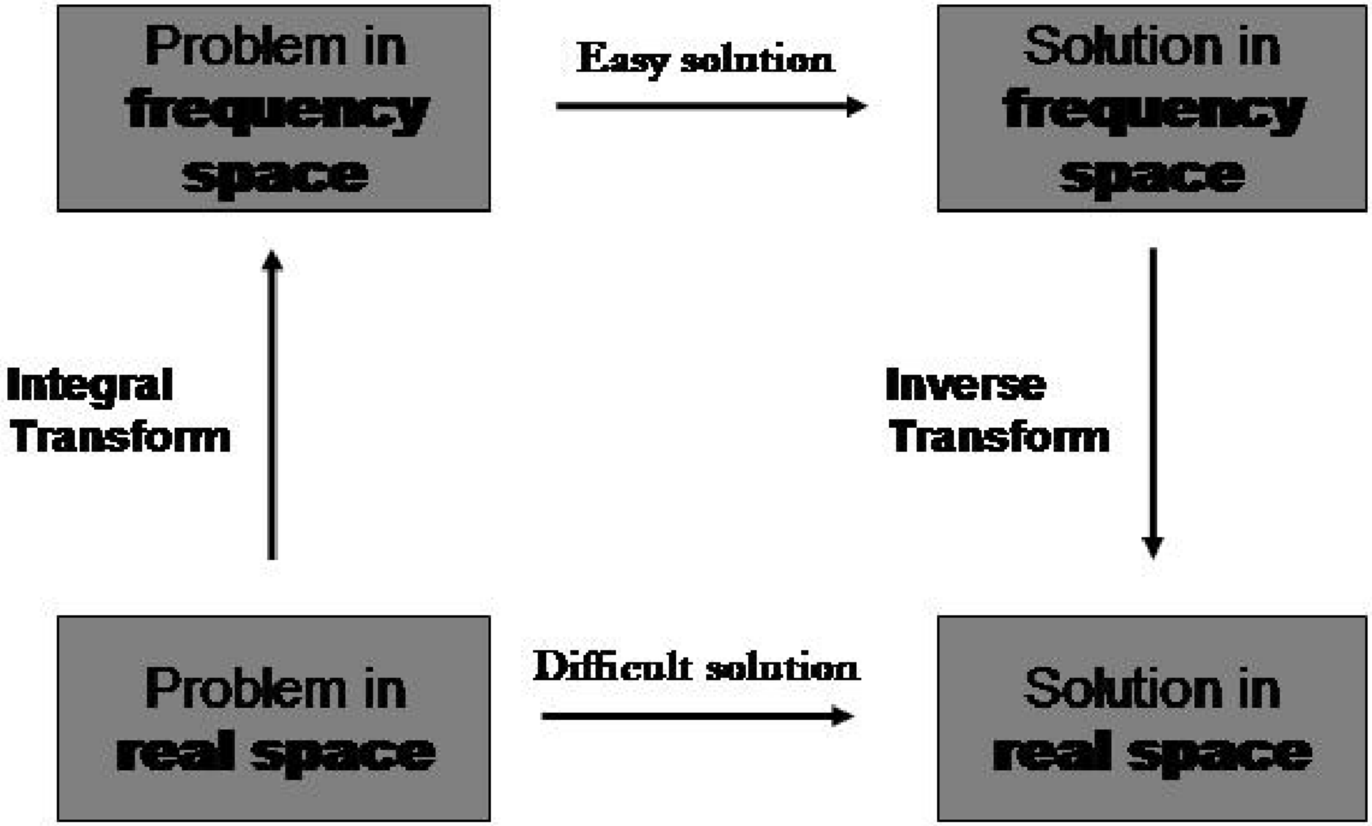
- considering the 2-D case: x, y are coordinates, u, v are frequencies in each direction.



http://www.fundipbook.com/

Fundamental Idea of Fourier Method

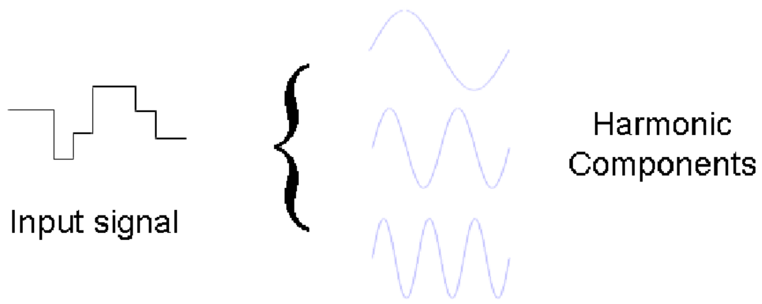
- The harmonic content of signal
- The Fourier representation is a complete alternative
- Fourier processing concerns the relation between the harmonic content of the output signal.



Frequency-space methods are used to make otherwise difficult problems easier to solve

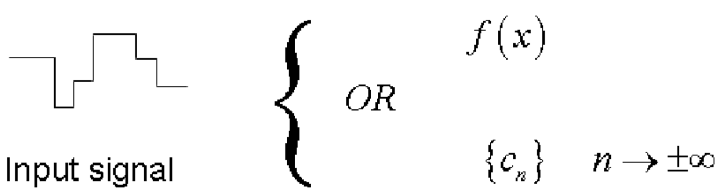
Frequency Space: the Fundamental Idea

1.



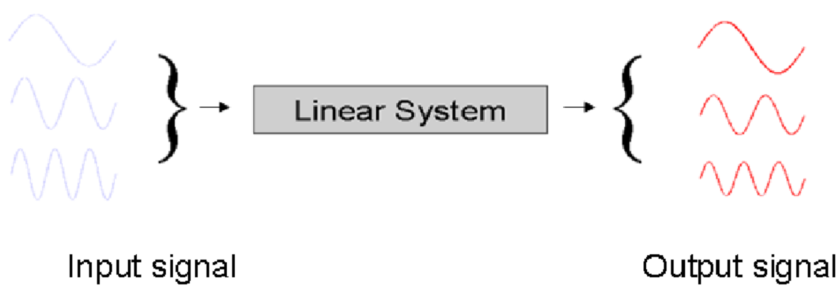
Input signals are decomposed into harmonic components.

2.



The decomposition is a complete and valid representation.

3.



From the frequency domain perspective, the action of any linear system on the input signal is to modify the amplitude and phase of the input components

1-D Discrete Fourier Transform

Fourier Transform of a discrete function of one variable, f(x), x=0,1,2,…,M-1 is given by the equation

$$F(u) = \frac{1}{M}\sum_{x=0}^{M-1} f(x)e^{-j2\pi ux/M}$$

For u=0,1,2,…,M-1

Inverse Discrete Fourier Transform of one variable, f(x), x=0,1,2,…,M-1 is given by the equation

$$f(x) = \sum_{u=0}^{M-1} F(u)e^{j2\pi ux/M}$$

For x=0,1,2,…,M-1

Fourier Spectrum: 1-Dimension

These parameters are used for signal processing

F(u) in polar coordinate

$$F(u) = |F(u)|e^{-j\emptyset(u)}$$

where

$$|F(u)|=[R^2(u) + I^2(u)]^{1/2}$$

Spectrum or Magnitude

$$\emptyset(u)=\tan^{-1}\frac{I(u)}{R(u)}$$

Phase Spectrum or Phase Angle

$R(u)$ is real part of F(u), $I(u)$ is imagine part of F(u)

Power Spectrum is defined as the squre of the Fourier Spectrum

$$P(u)= |F(u)|^2=R^2(u) + I^2(u)$$

The 2-D DFT and its Inverse

The discrete Fourier Transform of a function (image) f(x,y) of size MxN is given by the equation

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

For u=0,1,2,…,M-1 and v=0,1,2,…,N-1

Inverse Discrete Fourier Transform of two variable, f(x,y), x=0,1,2,…,M-1 is given by the equation

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

For x=0,1,2,…,M-1, and y=0,1,2,…,N-1

Fourier Spectrum: 2-Dimension

These parameters are depicted by image

Fourier spectrum, phase angle, and power spectrum

$$|F(u,v)|=[R^2(u,v)+I^2(u,v)]^{1/2}$$

Spectrum or Magnitude

$$\emptyset(u,v)=\tan^{-1}\frac{I(u,v)}{R(u,v)}$$

Phase Spectrum or Phase Angle

$R(u,v)$ is real part of F(u,v), $I(u,v)$ is imagine part of F(u,v)

Power Spectrum is defined as the squre of the Fourier Spectrum

$$P(u,v)=|F(u,v)|^2=R^2(u,v)+I^2(u,v)$$

Common Practice

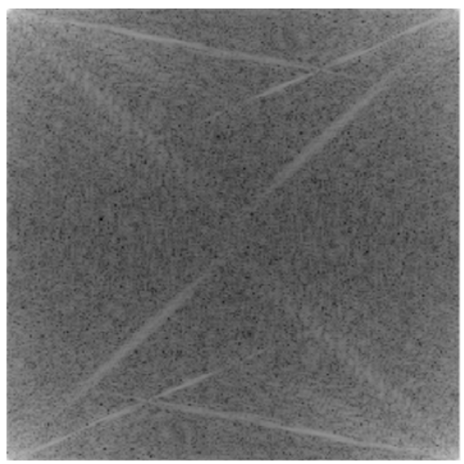
- Shift the origin of F(u,v)

Shifts the original of F(u,v) to frequency domain (M/2,N/2)

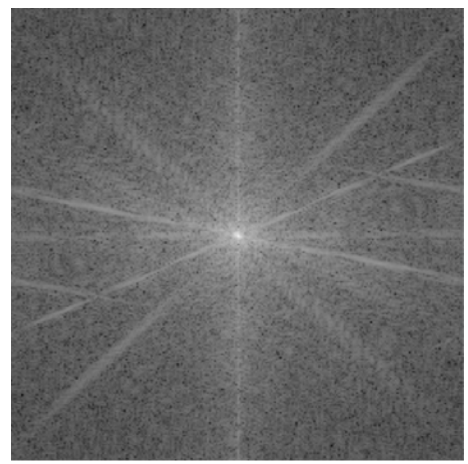
$$\delta[f(x,y)(-1)^{x+y}] = F(u-\frac{M}{2}, v-\frac{N}{2})$$

$\delta[*]$ Denotes the Fourier Transform of the argument

Input



Fourier without origin shift



Fourier with origin shift

Common Practice

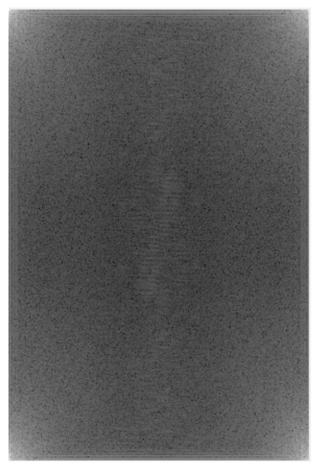
- Shift the origin of F(u,v)

Shifts the original of F(u,v) to frequency domain (M/2,N/2)

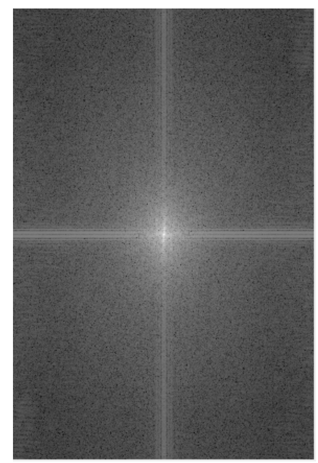
$$\delta[f(x,y)(-1)^{x+y}] = F(u - \frac{M}{2}, v - \frac{N}{2})$$

$\delta[*]$ Denotes the Fourier Transform of the argument

Input



Fourier without origin shift



Fourier with origin shift

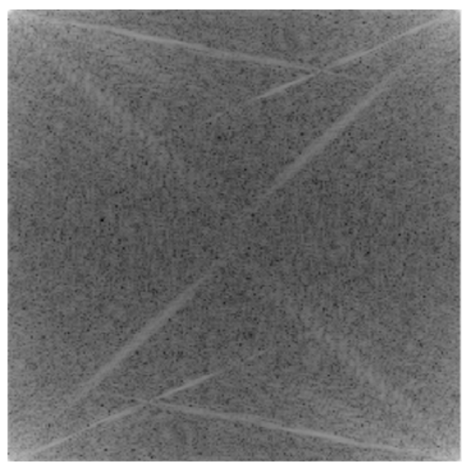
Common Practice

If f(x,y) is real, its Fourier transform is conjugate symmetric

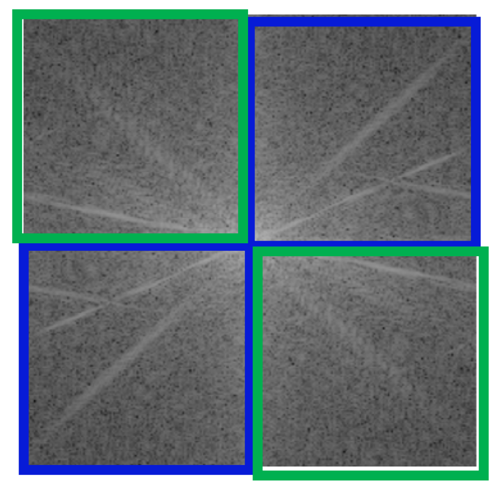
$$F(u, v) = F^{*}(-u, -v)$$

$$|F(u, v)| = |F^{*}(-u, -v)\,|$$

Input



Fourier without origin shift



Fourier with origin shift

Basic Steps for Filtering in the Frequency Domain

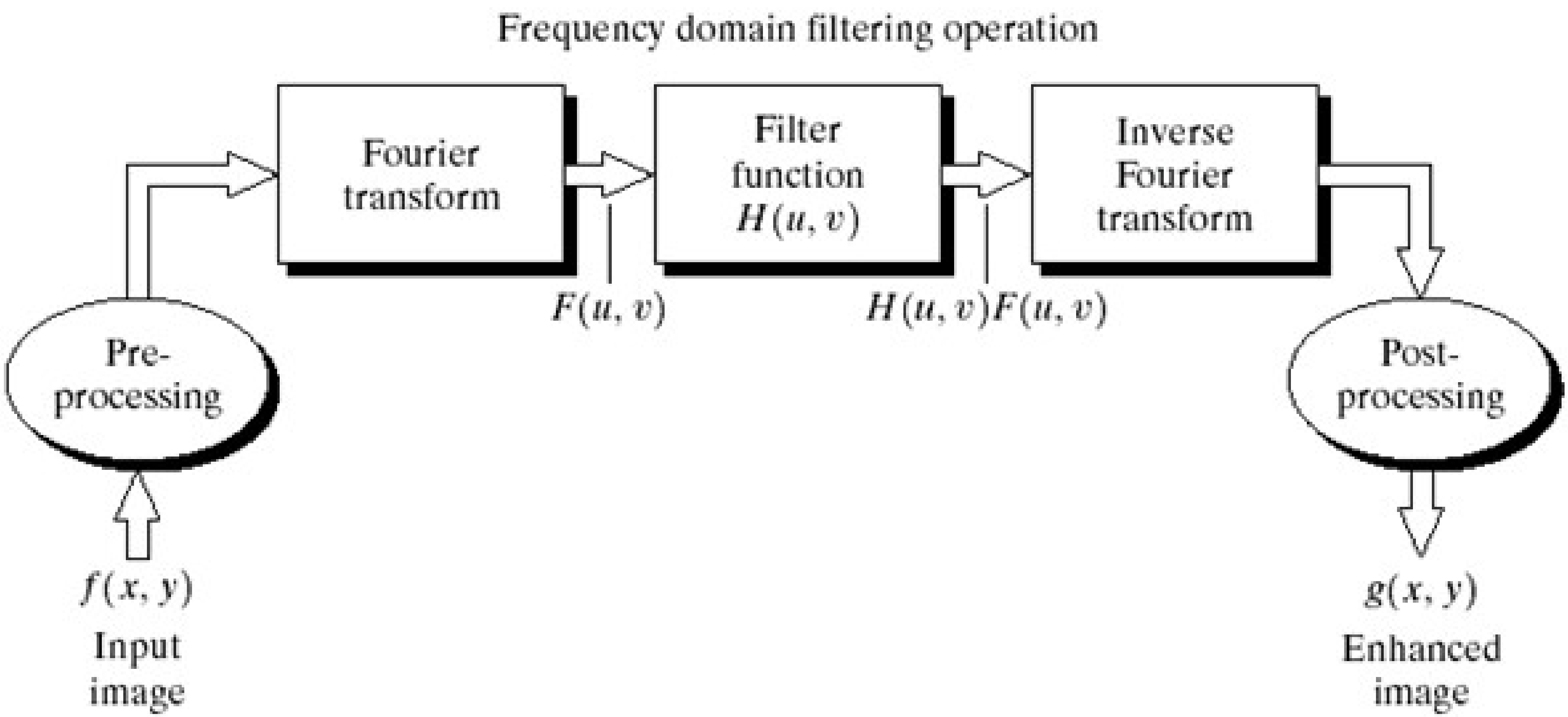


FIGURE 4.5 Basic steps for filtering in the frequency domain.

The convolution theorem

$$\mathbf{F}\{f(x,y)h(x,y)\} = F(k_x,k_y) * * H(k_x,k_y)$$

The optical transfer function

$$\mathbf{F}\{g(x,y)\} = \mathbf{F}\{f(x,y) ** h(x,y\}$$

$$G(k_x,k_y) = F(k_x,k_y)H(k_x,k_y)$$

$$\mathbf{F}\{f(x,y) * h(x,y\} = \underbrace{G(k_x,k_y)}_{\substack{\text{output}\\ \text{Fourier}\\ \text{spectrum}}} = \underbrace{F(k_x,k_y)}_{\substack{\text{input}\\ \text{Fourier}\\ \text{spectrum}}}\ \underbrace{H(k_x,k_y)}_{\text{OTF}}$$

Fourier Transform MATLAB

FourierTransform51.m

```
function FourierTransform51
% Example Matlab script as provided with textbook:
%
%  Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab
%  Chris J. Solomon and Toby P. Breckon, Wiley-Blackwell, 2010
%  ISBN: 0470844736, DOI:10.1002/9780470689776, http://www.fundipbook.com
%
A=imread('BBC_grey_testcard.png');      %Read in test card image
FA=fft2(A);
FA=fftshift(FA);            %Take FFT and centre it
PSF=fspecial('gaussian',size(A),6);         %Define PSF
OTF=fft2(PSF); OTF=fftshift(OTF);      %Calculate corresponding OTF
figure;
Afilt=ifft2(OTF.*FA); Afilt=fftshift(Afilt);        %Calculate filtered image
subplot(2,2,1);imshow(A,[]);  colormap(gray);title('Original'); %Display Results
subplot(2,2,2); imagesc(log(1+(PSF))); axis image; axis off;title('Gaussian PSF');
subplot(2,2,3); imagesc(log(1+abs(OTF))); axis image; axis off;title('MTF PSF');
subplot(2,2,4); imagesc(abs(Afilt)); axis image; axis off;title('Result');
```

แสดงขั้นตอนการใช้ คำสั่งใน MATLAB

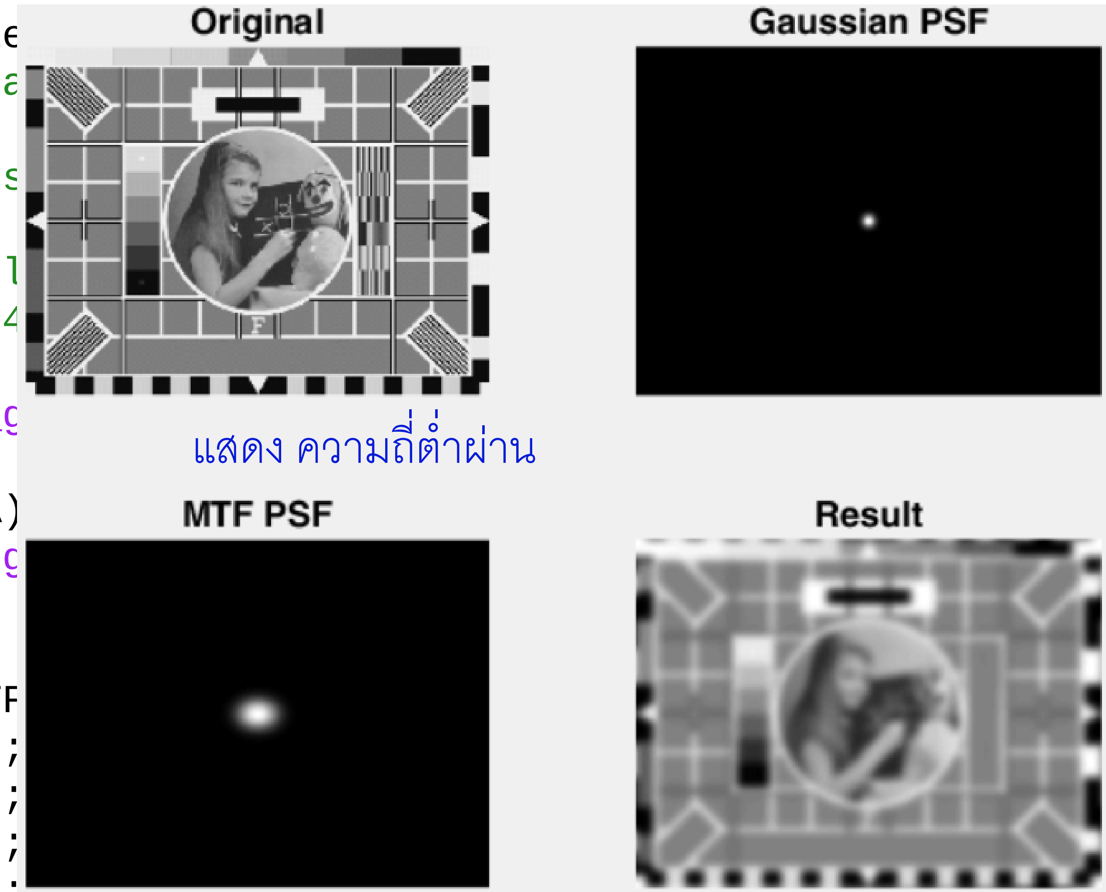
Fourier Transform MATLAB

```
function FourierTransform51
% Example Matlab script as provided with textbook:
%
%  Fundamentals of Digital Image Processing: A Practical Approach with Examples in
Matlab
%  Chris J. Solomon and Toby P. Breckon, Wiley-Blackwell, 2010
%  ISBN: 0470844736, DOI:10.1002/9780470689776, http://www.fundipbook.com
%
A=imread('BBC_grey_testcard.png');      %Read in test card image
FA=fft2(A);
FA=fftshift(FA);            %Take FFT and centre it
PSF=fspecial('gaussian',size(A),6);         %Define PSF
OTF=fft2(PSF); OTF=fftshift(OTF);      %Calculate corresponding OTF
figure;
Afilt=ifft2(OTF.*FA); Afilt=fftshift(Afilt);        %Calculate filtered image
subplot(2,2,1);imshow(A,[]);  colormap(gray);title('Original'); %Display Results
subplot(2,2,2); imagesc(log(1+(PSF))); axis image; axis off;title('Gaussian PSF');
subplot(2,2,3); imagesc(log(1+abs(OTF))); axis image; axis off;title('MTF PSF');
subplot(2,2,4); imagesc(abs(Afilt)); axis image; axis off;title('Result');
```

แสดงขั้นตอนการใช้ คำสั่งใน MATLAB

Fourier Transform MATLAB

```
function Fourie
% Example Matla
%
%  Fundamentals                                      roach with Examples in
Matlab
%  Chris J. Sol                                      .0
%  ISBN: 047084                                      ndipbook.com
%
A=imread('BBC_g                                      age
FA=fft2(A);
FA=fftshift(FA)
PSF=fspecial('g
OTF=fft2(PSF);                                       ling OTF
figure;
Afilt=ifft2(OTF                                      e filtered image
subplot(2,2,1);                                      ); %Display Results
subplot(2,2,2);                                      itle('Gaussian PSF');
subplot(2,2,3);                                      f;title('MTF PSF');
subplot(2,2,4); imagesc(abs(Afilt)); axis image; axis off;title('Result');
```

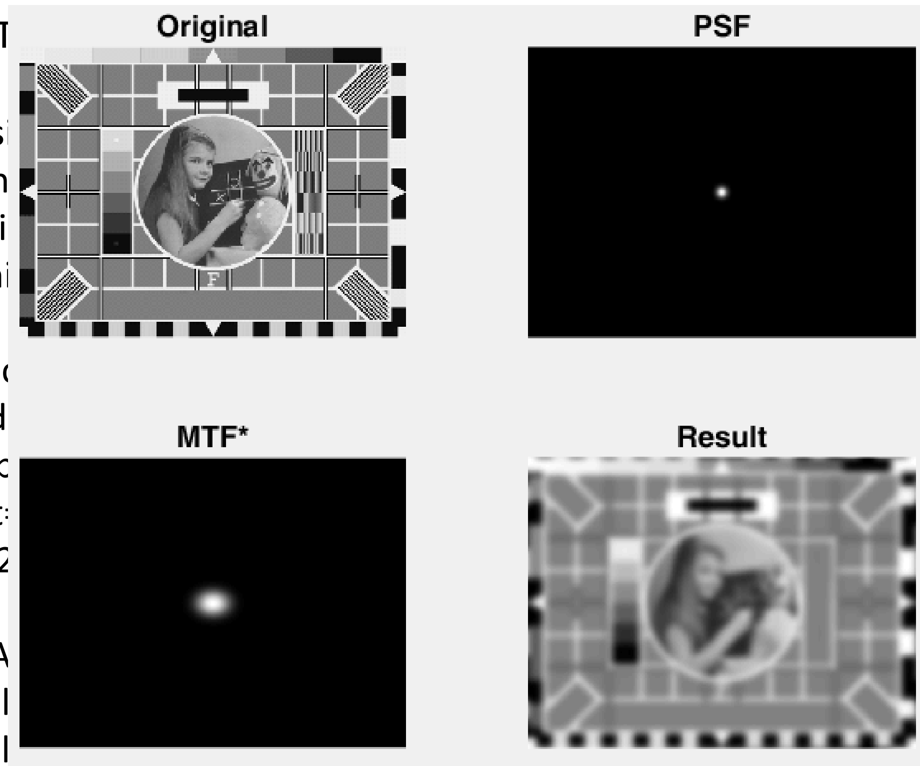


Fourier Transform MATLAB

```
function FourierTransform51
[ต่อ]
PSF=fspecial('gaussian',size(A),6);        %Define PSF
OTF=fft2(PSF); OTF=fftshift(OTF);      %Calculate corresponding OTF
rlow=(size(A,1)./2)-3; rhigh=(size(A,1)./2)+3;  %Define range to be altered
clow=(size(A,2)./2)-3; chigh=(size(A,2)./2)+3;
Fphase=angle(OTF);            %Extract Fourer phase
Fphase(rlow:rhigh,clow:chigh)=Fphase(rlow:rhigh,clow:chigh)+0.*pi.*rand;
                    %Add random component to selected phase
OTF=abs(OTF).*exp(i.*Fphase);          %Recombine phase and modulus
Afilt=ifft2(OTF.*FA); Afilt=fftshift(Afilt);       %Calculate filtered image
psfnew=abs(fftshift((otf2psf(OTF))));      %Calculate corresponding PSF
figure;
subplot(2,2,1);imshow(A,[]);  title('Original');
subplot(2,2,2); imagesc(log(1+psfnew)); axis image; axis off; colormap(gray);title('PSF');
subplot(2,2,3); imagesc(log(1+abs(OTF))); axis image; axis off;title('MTF*');
subplot(2,2,4); imagesc(abs(Afilt)); axis image; axis off;title('Result');
```

Fourier Transform MATLAB

```
function FourierT
[ต่อ]
PSF=fspecial('gaussian',s
OTF=fft2(PSF); OTF=fftsh
rlow=(size(A,1)./2)-3; rhi
clow=(size(A,2)./2)-3; chi
Fphase=angle(OTF);
Fphase(rlow:rhigh,clow:c
                    %Add rand
OTF=abs(OTF).*exp(i.*Fp
Afilt=ifft2(OTF.*FA); Afilt
psfnew=abs(fftshift((otf2
figure;
subplot(2,2,1);imshow(A
subplot(2,2,2); imagesc(l                                    PSF');
subplot(2,2,3); imagesc(l
subplot(2,2,4); imagesc(abs(Afilt)); axis image; axis off;title('Result');
```

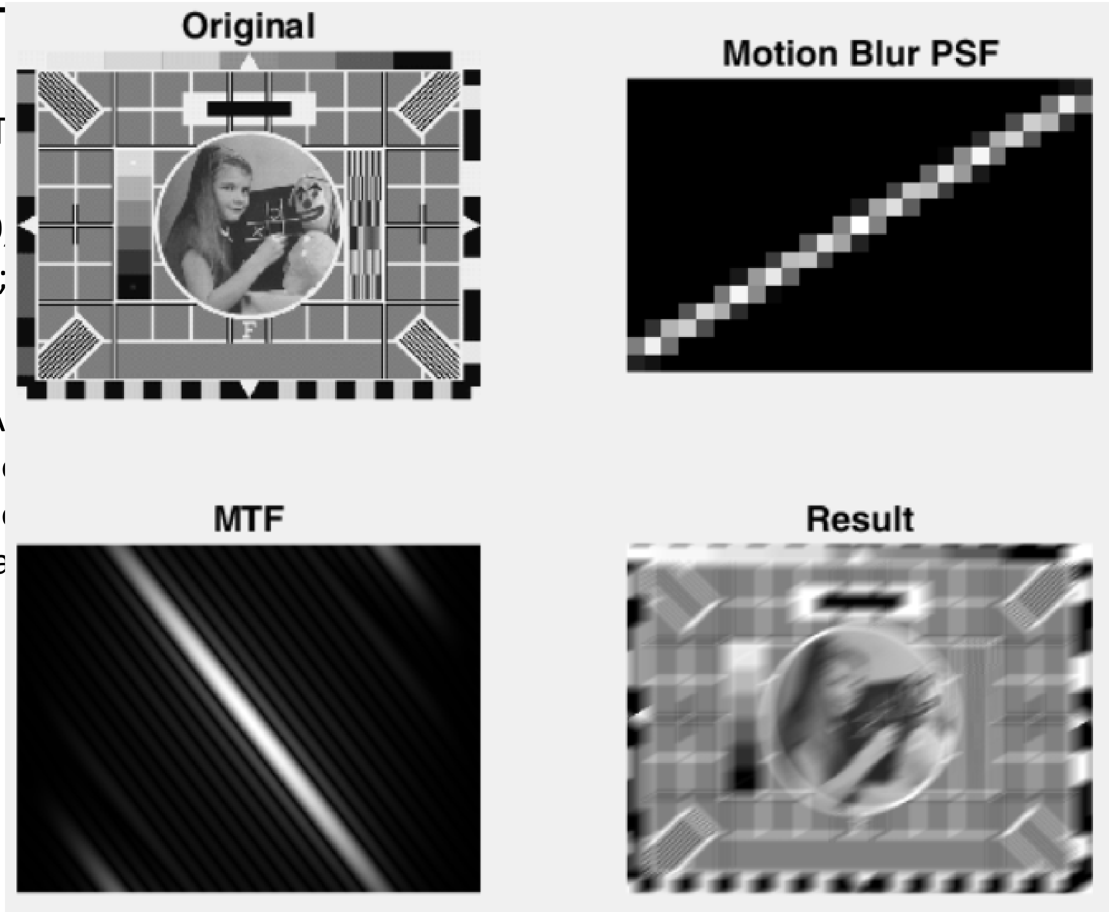


Fourier Transform MATLAB

```
function FourierTransform51
[ต่อ]
PSF=fspecial('motion',30,30);         %Define motion PSF
OTF=psf2otf(PSF,size(A)); OTF=fftshift(OTF);    %Calculate corresponding OTF
Afilt=ifft2(OTF.*FA);             %Calculate filtered image
figure;
subplot(2,2,1);imshow(A,[]);  title('Original');
subplot(2,2,2); imshow(log(1+PSF),[]); title('Motion Blur PSF');
subplot(2,2,3); imshow(log(1+abs(OTF)),[]);title('MTF');
subplot(2,2,4); imshow(abs(Afilt),[]);title('Result');
```

Fourier T

```
function FourierT
[ต่อ]
PSF=fspecial('motion',30
OTF=psf2otf(PSF,size(A));
Afilt=ifft2(OTF.*FA);
figure;
subplot(2,2,1);imshow(A
subplot(2,2,2); imshow(l
subplot(2,2,3); imshow(l
subplot(2,2,4); imshow(a
```



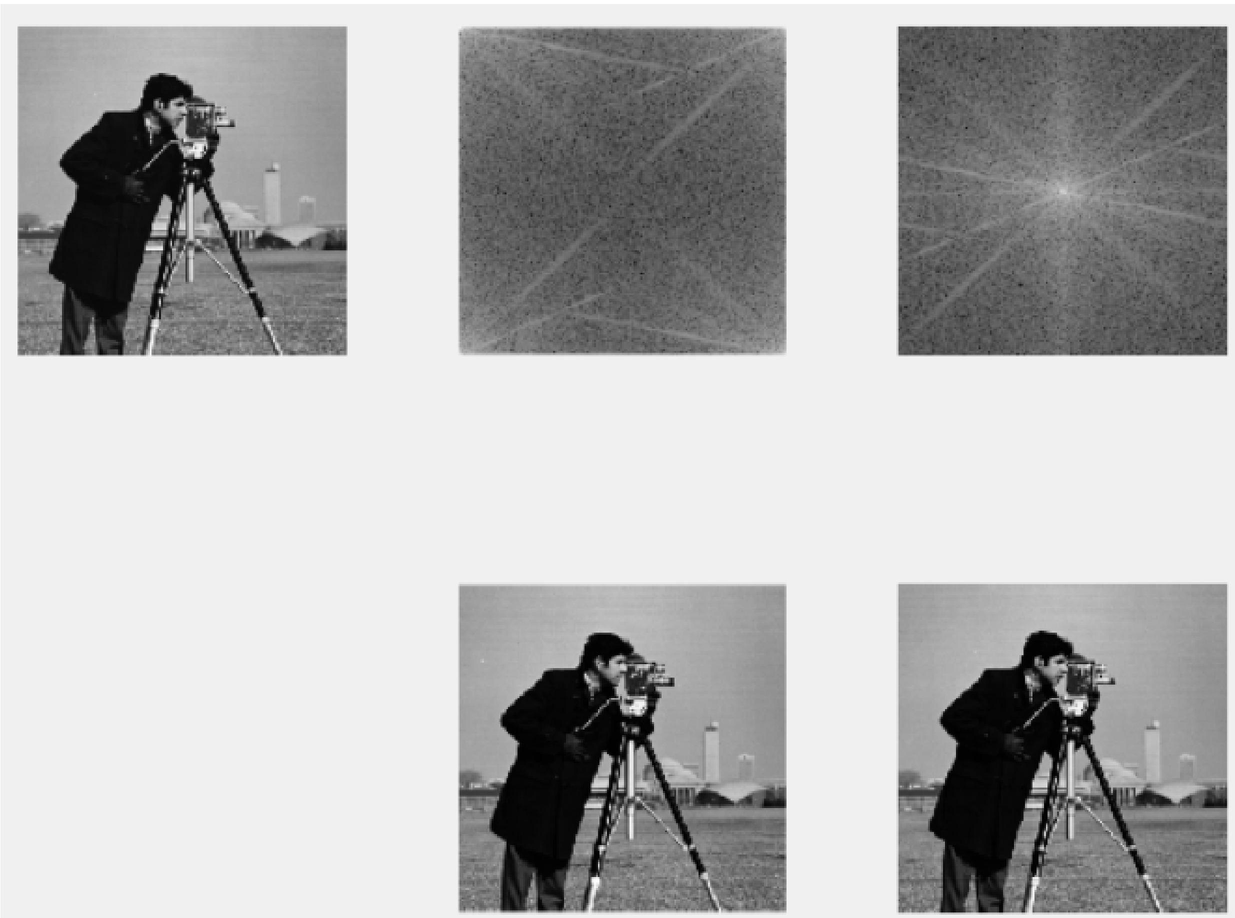
Fourier Transform MATLAB

FourierTransform52.m

```
function FourierTransform52
% Example Matlab script as provided with textbook:
%
%  Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab
%  Chris J. Solomon and Toby P. Breckon, Wiley-Blackwell, 2010
%  ISBN: 0470844736, DOI:10.1002/9780470689776, http://www.fundipbook.com
%
A=imread('cameraman.tif');                          %Read in image
%A=imread('OT0013.jpg');  %Read in image
%A = rgb2gray(A);
FT=fft2(A); FT_centred=fftshift(FT);                %take FT, get centred version too
subplot(2,3,1), imshow(A);                          %Display image
subplot(2,3,2), imshow(log(1+abs(FT)),[]);          %Display FT modulus (log scale)
subplot(2,3,3), imshow(log(1+abs(FT_centred)),[]);  %Display centred FT modulus(log scale)
```

Fourier Transform MATLAB

โปรแกรมนี้แสดงให้เห็นว่า การ Shift Fourier ไม่ได้ทำให้ภาพ Output เปลี่ยนไป



with Examples in Matlab

undipbook.com

l version too

ulus (log scale)
tred FT modulus(log scale)

Fourier Transform MATLAB

FourierTransform52.m

```
function FourierTransform52

[ต่อ ]
figure;
[xd,yd]=size(A); x=-xd./2:xd./2-1; y=-yd./2:yd./2-1;
[X,Y]=meshgrid(x,y); sigma=32;
arg=(X.^2+Y.^2)./sigma.^2;
frqfilt=exp(-arg);          %Construct freq domain filter
imfilt1=abs(ifft2(frqfilt.*FT));                    % Centred filter & non-centred spectrum
imfilt2=abs(ifft2(frqfilt.*FT_centred));              %image - Centred filter on centred spectrum
subplot(1,3,1), imshow(frqfilt,[]);                 %Display results
subplot(1,3,2), imshow(imfilt1,[]);
subplot(1,3,3), imshow(imfilt2,[]);
```

Fourier Transform MATLAB

FourierTransform52.m

โปรแกรมนี้แสดงให้เห็นว่า ถ้าใส่ฟิวเตอร์ ต้องใส่ให้ถูกตำแหน่งด้วย Low Frequency เริ่มที่ศูนย์กลางภาพ

function FourierTransform52

[ต่อ]

Low pass filter

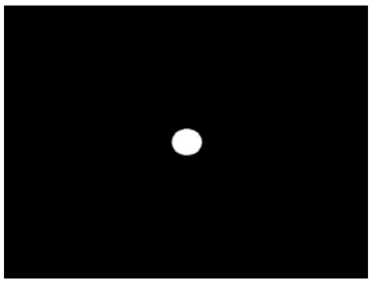
ถ้าเอา Low Pass ไปคูณกับ OTF ที่ไม่shift

ถ้าเอา Low Pass ไปคูณกับ OTF ที่ shift

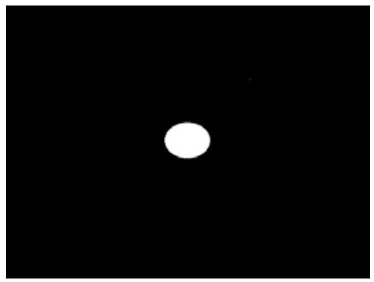
Fourier Transform MATLAB

FourierTransformWK3.m

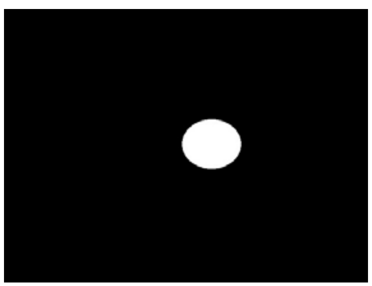
Low pass 1



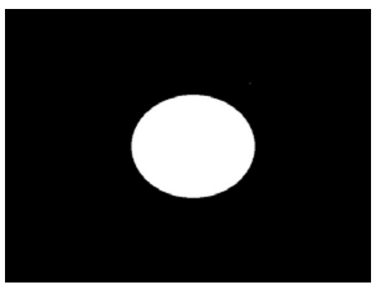
Low pass 2

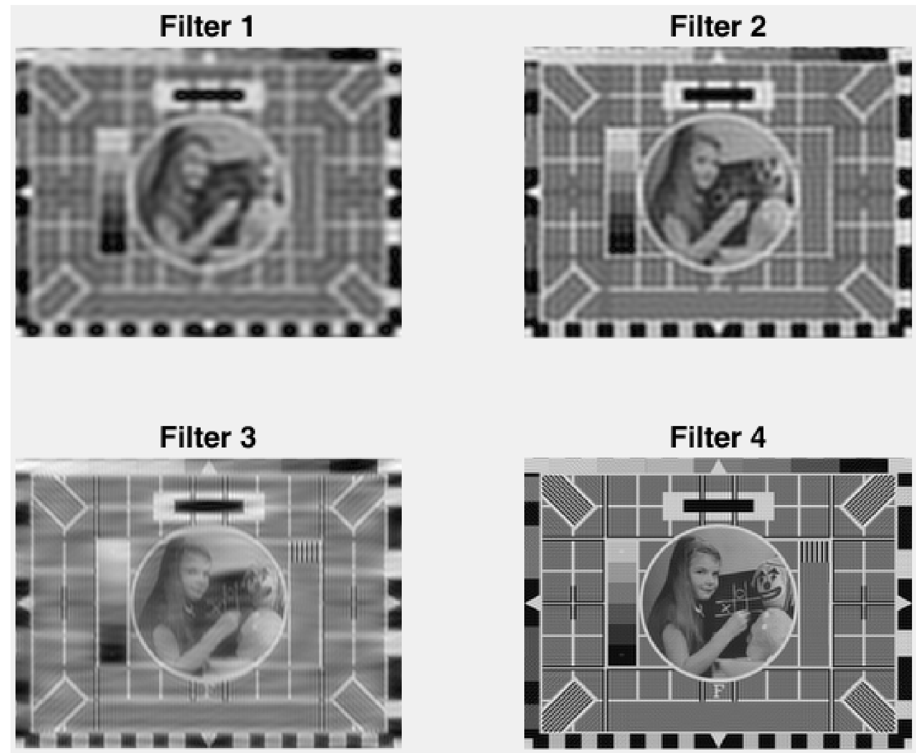


Low pass 3



Low pass 4





END