University of Technology and Education
Faculty of Electrical & Electronic Engineering

Lecture:
IMAGE PROCESSING

Chapter 4:

Image Filtering

1

Nguyen Thanh Hai, PhD

University of Technology and Education
Faculty of Electrical & Electronic Engineering

Image Filtering

INTRODUCTION

-Image Filtering allows to apply various effects on images in order to filter noise or to smooth image.

-The 2D filter is a 2D filter matrix

*** Filtering in the frequency domain**

- Lots of noises: Gauss, impulse, etc.

- Filters: lowpass, highpass, bandpass.

*** Filtering in the spatial domain**

- Linear filtering is a weighted mask used to express the effects of the filter on each pixel of the image.

2

Nguyen Thanh Hai, PhD

University of Technology and Education
Faculty of Electrical & Electronic Engineering

Image Filtering

IMAGE FILTERING IN THE SPATIAL DOMAIN

- A spatial filter means that a typical pixel in image is calculated based on neighborhood pixels. The filter is moved to all image pixels to create the output image.

Fig 4.1 describes a 3x3 spatial filter. Assume that one needs to calculate a image pixel (x, y) in an image, the corresponding pixel of output image $g(x, y)$ is calculated by the sum of products of coefficients (values) between the filter and image.

Nguyen Thanh Hai, PhD

University of Technology and Education
Faculty of Electrical & Electronic Engineering

Image Filtering

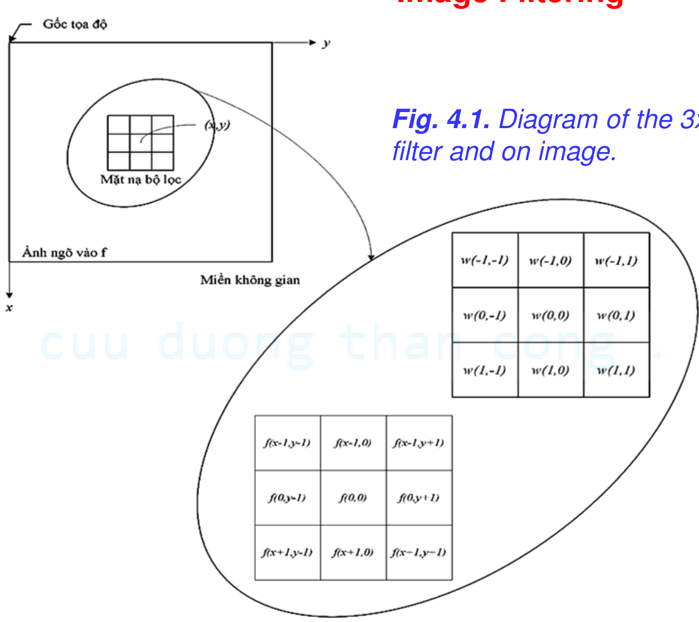


Fig. 4.1. *Diagram of the 3x3 spatial filter and on image.*

Nguyen Thanh Hai, PhD

University of Technology and Education
Faculty of Electrical & Electronic Engineering

Image Filtering

IMAGE FILTERING IN THE SPATIAL DOMAIN

- Notice: the center coefficient of filter is $w(0,0)$, corresponding to the image pixel considered (x, y). With a filter of $m \times n$, assume that $m = 2a + 1$ and $n = 2b + 1$, in which a and b are positive integers. The size of the odd and smallest filter is 3×3).

- In general, the $m \times n$ spatial filter used on the $M \times N$ input image to produce the output image $g(x, y)$ is expressed as follows:

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

$$g(x,y) = w(-1,-1)f(x-1,y-1) + w(-1,0)f(x-1,y) + \ldots + w(0,0)f(x,y) + \ldots + w(1,1)f(x+1,y+1)$$

Nguyen Thanh Hai, PhD

University of Technology and Education
Faculty of Electrical & Electronic Engineering

Image Filtering

IMAGE FILTERING IN THE SPATIAL DOMAIN

Correlation and convolution in spatial domain

- Correlation in image processing is to move a window (filter) on image and calculate the sum of products at every position.
- In convolution, except rotating the filter of 180^0.
- Notice: convolution means that a filter moves and rotate a degree of 180^0 and produce result rotated 180^0 as described in Figs 4.2(a) and 4.2(b).
- Fig. 4.2. is description of correlation and convolution methods between the 1D signal and the unit impulse. Notice of transposing.

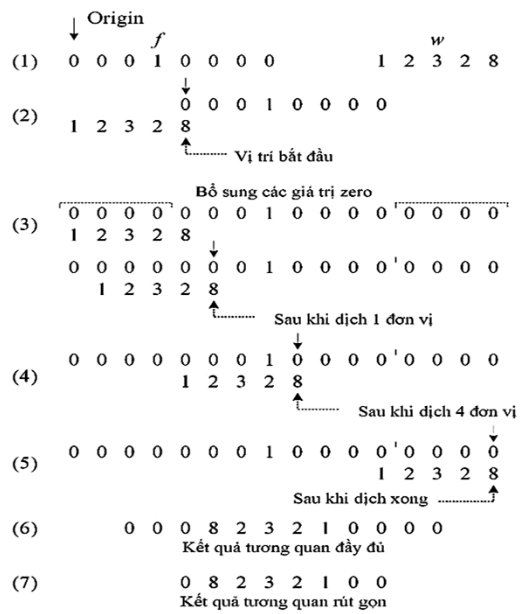
Nguyen Thanh Hai, PhD

University of Technology and Education
Faculty of Electrical & Electronic Engineering

Image Filtering

Fig. 4.2(a) *correlation method.*

Original is the 1st pixel, to correlate to 5 values of w, one can add zeros as Step-3 and then shift 1 by 1 from left to right. Every correlation is calculated by the sum of products of every point and in this case, 12 times shifted as result of Step-6



Nguyen Thanh Hai, PhD

University of Technology and Education
Faculty of Electrical & Electronic Engineering

Image Filtering

Fig. 4.2(b) *convolution method*

Original is the 1st pixel, to convolute to 5 values of w, one can add zeros as Step-10 and rotate the window w with a degree of 180 and then shift 1 by 1 from left to right. It is calculated by the sum of products of every point and in this case, 12 times shifted and then as result of Step-14

Nguyen Thanh Hai, PhD

University of Technology and Education
Faculty of Electrical & Electronic Engineering

Image Filtering

In general, the correlation expression is described as below equation, the filter $w(x,y)$ with the size of $m \times n$, the image $f(x,y)$ and called $w(x,y) \star f(x,y)$

$$w(x,y) \star f(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

Similarly, the convolution of $w(x,y)$ and $f(x,y)$, called $w(x,y) \bigstar f(x,y)$, is expressed as follows:

$$w(x,y) \bigstar f(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x-s, y-t)$$

The symbol *minus* means that the image f is rotated 180 degrees.

Nguyen Thanh Hai, PhD

University of Technology and Education
Faculty of Electrical & Electronic Engineering

Image Filtering

(a)

Origin $f(x,y)$

0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

$w(x,y)$

1	2	3
4	5	6
7	8	9

(b)

f sau khi được đệm các giá trị 0

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(b)

Vị trí bắt đầu với w

1	2	3	0	0	0	0	0	0
4	5	6	0	0	0	0	0	0
7	8	9	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(d)

Kết quả tương quan

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	9	8	7	0	0	0
0	0	0	6	5	4	0	0	0
0	0	0	3	2	1	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

Fig 4.3(a) *in similarity to Fig 4.2(a), correlation of an image f and the 3x3 filter w, result as (e)*

Nguyen Thanh Hai, PhD

University of Technology and Education
Faculty of Electrical & Electronic Engineering

Image Filtering

Kết quả rút gọn (e)

0	0	0	0	0
0	9	8	7	0
0	6	5	4	0
0	3	2	1	0
0	0	0	0	0

Xoay w (f)

9	8	7	0	0	0	0	0	0
6	5	4	0	0	0	0	0	0
3	2	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

Kết quả tích chập (g)

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	1	2	3	0	0	0
0	0	0	4	5	6	0	0	0
0	0	0	7	8	9	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

Kết quả rút gọn (h)

0	0	0	0	0
0	1	2	3	0
0	4	5	6	0
0	7	8	9	0
0	0	0	0	0

Fig 4.3(b) *in similarity to Fig 4.2(b), convolution of an image f and the 3x3 filter w, result as (h)*

Nguyen Thanh Hai, PhD

University of Technology and Education
Faculty of Electrical & Electronic Engineering

Image Filtering

Ex: Using the correlation and convolution methods to calculate the output image at the position g(2,1).

Input image (x: 0 to 4, y: 0 to 4)

1	1	1	2	2
1	1	1	2	2
1	**1**	1	2	2
2	2	2	2	2
2	2	2	2	2

Mask (filter) (s: -1 to 1, t: -1 to -1)

1	1	1
1	1	1
1	1	1

Output image (x, y)

	12			

$g(x,y)=g(2,1)$

$$g(2,1)=\omega(-1,-1)f(1,0)+\omega(-1,0)f(1,1)+\omega(-1,1)f(1,2)$$
$$+\omega(0,-1)f(2,0)+\omega(0,0)f(2,1)+\omega(0,1)f(2,2)$$
$$+\omega(1,-1)f(3,0)+\omega(1,0)f(3,1)+\omega(1,1)f(3,2)=12$$

Nguyen Thanh Hai, PhD

University of Technology and Education
Faculty of Electrical & Electronic Engineering

Image Filtering

IMAGE FILTERING IN THE SPATIAL DOMAIN

Some functions in MATLAB

- *Ones:* matrix 1.
- imfilter(f,h,'Boundary Options','OutputSize','Correlation and Convolution Options')
- 'Boundary Options' : allow to select a boundary of the output image
- 'Output Size': select the output size with 'Correlation and Convolution Options'.
- More functions in Table. 4.1 of Image Processing book

13

Nguyen Thanh Hai, PhD

University of Technology and Education
Faculty of Electrical & Electronic Engineering

Image Filtering

Some filters expressed by mathematic methods:

* Gaussian

$$h_g(n_1,n_2)=e^{\frac{-(n_1^2+n_2^2)}{2\sigma^2}}$$

or

$$h(n_1,n_2)=\frac{h_g(n_1,n_2)}{\sum n_1 \sum n_2 h_g}$$

* Laplacian

$$\nabla^2=\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}$$

or

$$\begin{bmatrix} \frac{\alpha}{1+\alpha} & \frac{1-\alpha}{1+\alpha} & \frac{\alpha}{1+\alpha} \\ \frac{1-\alpha}{1+\alpha} & \frac{-4}{1+\alpha} & \frac{1-\alpha}{1+\alpha} \\ \frac{\alpha}{1+\alpha} & \frac{1-\alpha}{1+\alpha} & \frac{\alpha}{1+\alpha} \end{bmatrix}$$

14

Nguyen Thanh Hai, PhD

Image Filtering

* Log (Laplacian of Gaussian)

$$h(n_1,n_2)=\frac{\left(n_1^2+n_2^2-2\sigma^2\right)h_g(n_1,n_2)}{2\pi\sigma^6\sum n_1\sum n_2 h_g}$$

* Prewitt

$$h=\begin{bmatrix}1 & 1 & 1\\ 0 & 0 & 0\\ -1 & -1 & -1\end{bmatrix}$$

* Sobel

$$h=\begin{bmatrix}1 & 2 & 1\\ 0 & 0 & 0\\ -1 & -2 & -1\end{bmatrix}$$

Image Filtering

Ex 4.1: Express image filtering using filters in toolbox of MATLAB

```
clear all;
f=imread('cameraman.bmp');
ha=fspecial('average');
ga=imfilter(f,ha,'replicate');
hd=fspecial('disk');
gd=imfilter(f,hd,'replicate');
hg=fspecial('gaussian');
gg=imfilter(f,hg,'replicate');
hlap=fspecial('laplacian');
glap=imfilter(f,hlap,'replicate');
hlog=fspecial('log');
glog=imfilter(f,hlog,'replicate');
```

```
hm=fspecial('motion');
gm=imfilter(f,hm,'replicate');
hp=fspecial('prewitt');
gp=imfilter(f,hp,'replicate');
hs=fspecial('sobel');
gs=imfilter(f,hs,'replicate');
```

University of Technology and Education
Faculty of Electrical & Electronic Engineering

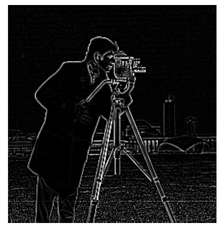
Image Filtering


(a)

(b)

(c)


(d)

(e)

(f)

Nguyen Thanh Hai, PhD

University of Technology and Education
Faculty of Electrical & Electronic Engineering

Image Filtering


(g)

(h)

(i)

Fig 4.4. *Express images with different filters*
(a) Original image; (b) Average; (c) Disk; (d) Gaussian; (e) Laplacian; (f) Log; (g) Motion; (h) Prewitt; (i) Sobel

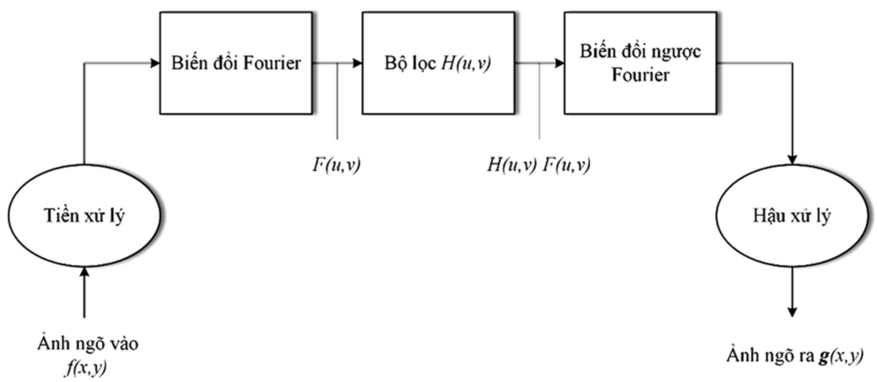
Nguyen Thanh Hai, PhD

University of Technology and Education
Faculty of Electrical & Electronic Engineering

Image Filtering

IMAGE FILTERING IN THE FREQUENCY DOMAIN

- Some applications in frequency domain, one often uses the FT.
- Image filtering in the frequency domain



University of Technology and Education
Faculty of Electrical & Electronic Engineering

Image Filtering

IMAGE FILTERING IN THE FREQUENCY DOMAIN

According to diagram of Fig above, frequency filtering can be expressed in MATLAB using the following functions:

```
function g=dftfilt(f,H)
F=fft2(f,size(H,1),size(H,2));
G=H.*F;
g=ifft2(G);
f=real(g);
g=g(1:size(f,1),1:size(f,2))
end
```

University of Technology and Education
Faculty of Electrical & Electronic Engineering

Image Filtering

IMAGE FILTERING IN THE FREQUENCY DOMAIN

- H is the impulse response of filter with frequency, assume that its size is 4 times of the input image. In the spatial domain, one uses convolution, but in the frequency, the multiplication is used. After filtering, the output values (matrix) of the filter will be converted using IFT to obtain the image with the same size of the original image.

- Building the frequency filter using functions in MATLAB as freqz2 with FIR (Finite Impulse Response)

- H=freqz2(h,R,C)
In which h is the 2D spatial filter, H is the 2D frequency filter response, R and C are the number of rows and columns of H.

Nguyen Thanh Hai, PhD

University of Technology and Education
Faculty of Electrical & Electronic Engineering

Image Filtering

IMAGE FILTERING IN THE FREQUENCY DOMAIN

Ex 4.2: Express image filtering in the spatial domain using Sobel and in the frequency domain using DFT

```
clear all;
f=imread('cameraman.tif');
h=fspecial('sobel');
sizeH=size(f)*2;
H=freqz2(h,sizeH);
H1=ifftshift(H);
gd=dftfilt(f,H1);
gs=imfilter(double(f),h);
d=abs(gd-gs);
gd=uint8(gd);imshow(gd);
gd=uint8(gd);imshow(gs);
```


Nguyen Thanh Hai, PhD

IMAGE FILTERING IN THE FREQUENCY DOMAIN

Checking homologous between two methods:

$$d = |gs - gd|$$

In which *gs* is the image filtered in the spatial and *gd* is filtered in the frequency, results as in Fig. 4.6

***Fig. 4.6.** Sobel filter*
(a). in the frequency
(b) in the spatial
(c) differential $\max d = 3.4106 \times 10^{-13}$

IMAGE FILTERING IN THE FREQUENCY DOMAIN

Build function dftuv for calculation of distance with grid of .

```
function [U,V]=dftuv(m,n)
u=0:(m-1);
v=0:(n-1);
idx=find(u>m/2);
u(idx)=u(idx)-m;
idy=find(v>n/2);
v(idy)=v(idy)-n;
[V,U]=meshgrid(v,u);
end
```

Ex of an image with 8×5 :

```
[U,V]=dftuv(8,5);
D=U.^2+V.^2
D =
   0    1    4    4    1
   1    2    5    5    2
   4    5    8    8    5
   9   10   13   13   10
  16   17   20   20   17
   9   10   13   13   10
   4    5    8    8    5
   1    2    5    5    2
```

University of Technology and Education
Faculty of Electrical & Electronic Engineering

Image Filtering

IMAGE FILTERING IN THE FREQUENCY DOMAIN

Using function fftshift (D) move coordinate to the center of the frequency rectangular image (frequency retangle) and the result:

ans =

20	17	16	17	20
13	10	9	10	13
8	5	4	5	8
5	2	1	2	5
4	1	0	1	4
5	2	1	2	5
8	5	4	5	8
10	9	10	13	

Nguyen Thanh Hai, PhD

University of Technology and Education
Faculty of Electrical & Electronic Engineering

Image Filtering

Lowpass Filter: blurring image

* Ideal LowPass Filter - ILPF) : the transfer functions is described as:

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

where D_0 is the cut-off frequency with non-negative value, $D(u,v)$ is the distance from (u,v) to the filter center. Orbit of $D(u,v) = D_0$ is a circle.

* Butterworth LowPass Filter - BLPF) with *n* orders, the cut-off frequency D_0, the transfer functions is described as:

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

Often define the cut-off frequency at positions of $H(u,v) = 0.5$ (reduce 50% compared with the max value 1) when $D(u,v) = D_0$.

Nguyen Thanh Hai, PhD

University of Technology and Education
Faculty of Electrical & Electronic Engineering

Image Filtering

IMAGE FILTERING IN THE FREQUENCY DOMAIN

Transfer function of the Gaussian Lowpass Filter (GLPF) is expressed as follows:

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

In which σ is the standard deviation. If $\sigma = D_0$, we have :

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

When $D(u,v) = D_0$, the value at the cut-off frequency is 0.0607. Fig. 4.7 describes the 3-D shapes of the frequency response of every filter.

Nguyen Thanh Hai, PhD

University of Technology and Education
Faculty of Electrical & Electronic Engineering

Image Filtering

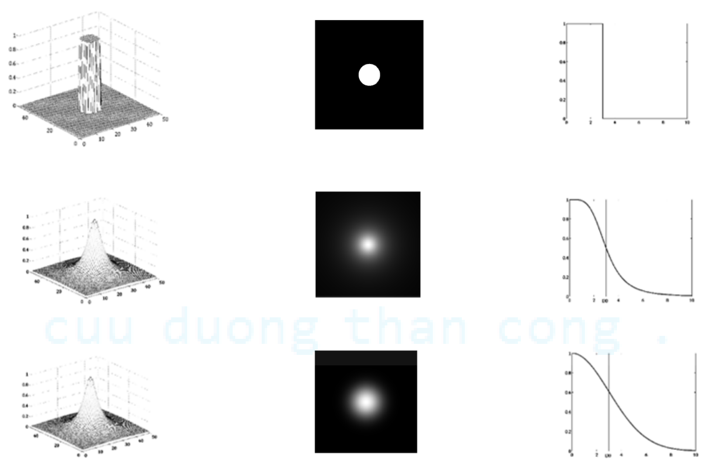


Fig. 4.7. *left column: 3-D shape of the window filter; middle columns: images of the filters; right one: the frequency response D_0 of the ideal filters. First row: the ideal lowpass filter; middle one: the Butterworth lowpass filter; last one: the Gaussian lowpass filter*

Nguyen Thanh Hai, PhD

University of Technology and Education
Faculty of Electrical & Electronic Engineering

Image Filtering

EX: *Build* a *lowpass filter function* as lpfilter with 3 basic *filter.*

```
function [H,D]=lpfilter(type,M,N,DO,n)
[U,V]=dftuv(M,N);
D=sqrt(U.^2+V.^2);
switch type
    case 'ideal'
        H=double(D<=DO);
    case 'btw'
        if nargin==4
            n=1;
        end
        H=1./(1+(D./DO).^(2*n));
    case 'gaussian'
        H=exp(-
(D.^2)./(2*(DO^2)));
    otherwise
        error('Unknown filter
type');
end
```

Some functions in toolbox:
- mesh(H): express information in 3D such as x=1:M và y=1:N, with [M,N]=size(H).
- mesh(H(1:k:end,1:k:end))
- colormap([0 0 0]): black-white

Nguyen Thanh Hai, PhD

University of Technology and Education
Faculty of Electrical & Electronic Engineering

Image Filtering

IMAGE FILTERING IN THE FREQUENCY DOMAIN

Ex 4.3*: express the plot using mesh for the Butterworth lowpass filter with different orders.*

```
clear all;
BLPF1=fftshift(lpfilter('btw',500,500,5
0,1));
mesh(BLPF1(1:10:500,1:10:500));
axis([0 50 0 50 0 1]);
saveas(gcf, 'Hinh3.7a.tif', 'tif');

BLPF2=fftshift(lpfilter('btw',500,500,5
0,2));
figure;mesh(BLPF2(1:10:500,1:10:50
0));
axis([0 50 0 50 0 1]);
saveas(gcf, 'Hinh3.7b.tif', 'tif');
BLPF3=fftshift(lpfilter('btw',500,500,
50,3));
figure;mesh(BLPF3(1:10:500,1:10:5
00));
axis([0 50 0 50 0 1]);
saveas(gcf, 'Hinh3.7c.tif', 'tif');

BLPF4=fftshift(lpfilter('btw',500,500,
50,4));
figure;mesh(BLPF4(1:10:500,1:10:5
00));
axis([0 50 0 50 0 1]);
saveas(gcf, 'Hinh3.7d.tif', 'tif');
```


Nguyen Thanh Hai, PhD

University of Technology and Education
Faculty of Electrical & Electronic Engineering

Image Filtering

IMAGE FILTERING IN THE FREQUENCY DOMAIN

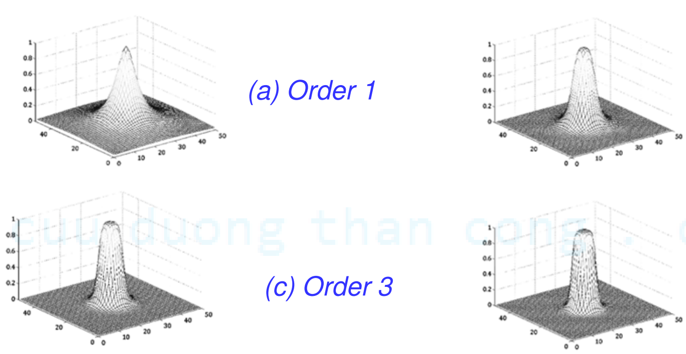
```
BLPF3=fftshift(lpfilter('btw',500,500,50,3));
figure;mesh(BLPF3(1:10:500,1:10:500));
axis([0 50 0 50 0 1]);
saveas(gcf, 'Hinh3.7c.tif', 'tif');

BLPF4=fftshift(lpfilter('btw',500,500,50,4));
figure;mesh(BLPF4(1:10:500,1:10:500));
axis([0 50 0 50 0 1]);
saveas(gcf, 'Hinh3.7d.tif', 'tif');
```


Nguyen Thanh Hai, PhD

University of Technology and Education
Faculty of Electrical & Electronic Engineering

Image Filtering

IMAGE FILTERING IN THE FREQUENCY DOMAIN



(a) Order 1 *(b) Order 2*

(c) Order 3 *(d) Order 4*

Fig 4.8. *Plots of the Butterworth lowpass filters with different orders.*

Nguyen Thanh Hai, PhD

Image Filtering

IMAGE FILTERING IN THE FREQUENCY DOMAIN

***Ex 4.4**: express the lowpass filter with different cut-off frequencies and compare results*

```
clear all;
f=imread('cameraman.bmp');

D01=1/5*size(f,1);
[M,N]=size(f);
[H,D]=lpfilter('ideal',2*M,2*N,D01);
g=dftfilt(f,H);

D02=2/5*size(f,1);
[M,N]=size(f);
[H,D]=lpfilter('ideal',2*M,2*N,D02);
g=dftfilt(f,H);

D03=3/5*size(f,1);
[M,N]=size(f);
[H,D]=lpfilter('ideal',2*M,2*N,D03);
g=dftfilt(f,H);

D04=4/5*size(f,1);
[M,N]=size(f);
[H,D]=lpfilter('ideal',2*M,2*N,D04);
g=dftfilt(f,H);

D05=size(f,1);
[M,N]=size(f);
[H,D]=lpfilter('ideal',2*M,2*N,D04);
g=dftfilt(f,H);
```


Image Filtering

(a) Original image

(b) $D_0 = \frac{M}{5}$

(c) $D_0 = \frac{2M}{5}$

(d) $D_0 = \frac{3M}{5}$

(e) $D_0 = \frac{4M}{5}$

(f) $D_0 = M$

***Fig 4.9.** cameraman* $M \times N$ *image after the ideal lowpass filter with different cut-off frequencies*

University of Technology and Education
Faculty of Electrical & Electronic Engineering

Image Filtering

Ideal highpass filter

The ideal highpass filter is often implemented to filter image sharper by eliminating the low frequencies

If the transfer of the lowpass filter is $H_{lp}(u,v)$, we have the highpass filter as follows:

$$H_{hp}(u,v) = 1 - H_{lp}(u,v)$$

Build the highpass filter function using Matlab

```
function H=hpfilter(type,M,N,D0,n)
if nargin==4
    n=1;
End
Hlp=lpfilter(type,M,N,D0,n);
H=1-Hlp;
end
```


University of Technology and Education
Faculty of Electrical & Electronic Engineering

Image Filtering

IMAGE FILTERING IN THE FREQUENCY DOMAIN

Ex 4.5: Express the highpass filter using the function *mesh* for image.

```
clear all;close all;clc
ILPF=fftshift(hpfilter('ideal',500,500,50));
mesh(ILPF(1:10:500,1:10:500));
axis([0 50 0 50 0 1]);

BLPF=fftshift(hpfilter('btw',500,500,50));
mesh(BLPF(1:10:500,1:10:500));
axis([0 50 0 50 0 1]);

GLPF=fftshift(hpfilter('gaussian',500,500,50));
mesh(GLPF(1:10:500,1:10:500));
axis([0 50 0 50 0 1]);
```

University of Technology and Education
Faculty of Electrical & Electronic Engineering

Image Filtering

IMAGE FILTERING IN THE FREQUENCY DOMAIN

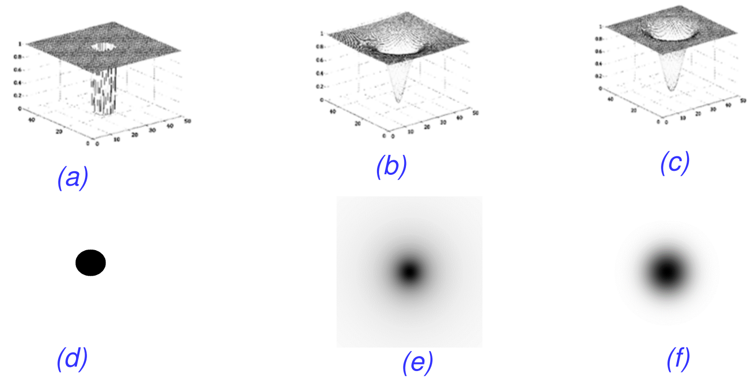


Fig. 4.10. *Express the highpass filter. Top row: mesh with the 3-D shape; bottom row: 2D images of the mesh; left column: the ideal highpass filter, middle columns: the Butterworth highpass filter, right one: the Gaussian highpass filter*

Nguyen Thanh Hai, PhD

University of Technology and Education
Faculty of Electrical & Electronic Engineering

Image Filtering

IMAGE FILTERING IN THE FREQUENCY DOMAIN

- The highpass filter is normalized to be zero to the DC components. Thus, this will reduce the average value of the image onto zero.
- For improvement, the DC offset components are added to the highpass filter.
- The high frequency emphasis filtering with offset is expressed as follows:

$$H_{hfe}(u,v) = a + bH_{hp}(u,v)$$

In which a is the offset, b denotes the multiplication constant and $H_{hp}(u,v)$ describes the transfer of the highpass filter.

Nguyen Thanh Hai, PhD

University of Technology and Education
Faculty of Electrical & Electronic Engineering

Image Filtering

IMAGE FILTERING IN THE FREQUENCY DOMAIN

Ex 4.6: express the high frequency emphasis filtering using Matlab.

```
clear all;
f=imread('satellite.tiff');
[M N]=size(f);
D0=0.1*size(f,1);
HBW=hpfilter('btw',2*M,2*N,D0);
H=0.5+2*HBW;
gbw=dftfilt(f,HBW);
gbw=gscale(gbw);
ghf=dftfilt(f,H);
ghf=gscale(ghf);
ghe=histeq(ghf,256);
```


Nguyen Thanh Hai, PhD

University of Technology and Education
Faculty of Electrical & Electronic Engineering

Image Filtering

IMAGE FILTERING IN THE FREQUENCY DOMAIN

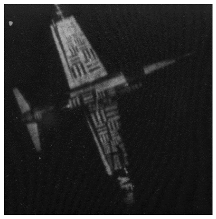
In the Ex 4.6, the *gscale* allows to normalize and change data type containing the gray intensity in image. The *gscale* function is proposed:

```
function g=gscale(f)
fmin=min(f(:));
fmax=max(f(:));
for i=1:size(f,1)
    for j=1:size(f,2)
g(i,j)=255*(f(i,j)-fmin)/(fmax-fmin);
    end
end
g=uint8(g);
end
```

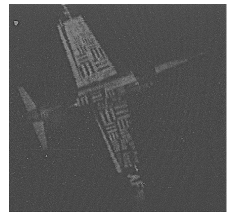

Nguyen Thanh Hai, PhD

University of Technology and Education
Faculty of Electrical & Electronic Engineering

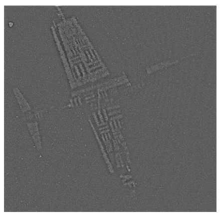
Image Filtering



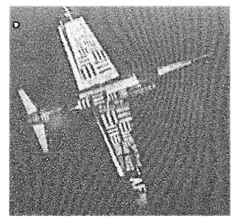
(a) Blurring satellite image



(c) Image after the emphasis highpass filter



(b) Image after the Butterworth highpass filter



(d) Image after the histogram equalization of image (c)

Ex 4.11. *Satellite image after the* highpass filter

University of Technology and Education
Faculty of Electrical & Electronic Engineering

Image Filtering

Problem: Calculate the convolution of the image A and the mask h.

- Suppose the input image is

```
A = [17  24   1   8  15
     23   5   7  14  16
      4   6  13  20  22
     10  12  19  21   3
     11  18  25   2   9]
```


- And the kernel (mask)

```
h = [8   1   6
     3   5   7
     4   9   2]
```

University of Technology and Education
Faculty of Electrical & Electronic Engineering

Image Filtering

The End

43

Nguyen Thanh Hai, PhD