

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT
THÀNH PHỐ HỒ CHÍ MINH**

-----**-----
*

TS. NGUYỄN THANH HẢI

**GIÁO TRÌNH
XỬ LÝ ẢNH**

(Ngành Điện - Điện Tử)

**NHÀ XUẤT BẢN ĐẠI HỌC QUỐC GIA
THÀNH PHỐ HỒ CHÍ MINH – 2014**

Chương 4

LỌC ẢNH

Ảnh trước khi được xử lý trong hệ thống cần được tiền xử lý. Lọc là một trong những bước này. Chương này trình bày các bộ lọc không gian và tần số để áp dụng lên ảnh.

4.1. LỌC TRONG MIỀN KHÔNG GIAN

Một bộ lọc không gian bao gồm một tập các điểm ảnh và toán tử xác định được áp dụng lên điểm ảnh đang xem xét và các điểm ảnh lân cận của nó. Bộ lọc (toán tử lọc ảnh) được di chuyển đến từng điểm ảnh để tạo ra ảnh. Nếu toán tử được thực hiện trên ảnh là tuyến tính thì ta gọi đó là bộ lọc không gian tuyến tính. Ngược lại ta có bộ lọc phi tuyến.

Hình 4.1 mô tả cơ chế của phép lọc không gian với cửa sổ 3×3 . Tại bất kỳ điểm ảnh nào (x, y) , điểm ảnh tương ứng $g(x, y)$ của bộ lọc là tổng các tích của các hệ số của bộ lọc với các giá trị điểm ảnh tại vị trí tương ứng:

$$g(x, y) = w(-1, -1)f(x - 1, y - 1) + w(-1, 0)f(x - 1, y) \quad (4.1)$$

$$+ \dots + w(0, 0)f(x, y) + \dots + w(1, 1)f(x + 1, y + 1)$$

Lưu ý là hệ số trung tâm của bộ lọc, $w(0, 0)$, tương ứng với pixel đang xét là (x, y) . Với cửa sổ lọc có kích thước $m \times n$, chúng ta giả sử rằng $m = 2a + 1$ và $n = 2b + 1$, với a và b là các số nguyên dương. Nói cách khác trong phần này, chúng ta chỉ tập trung vào các bộ lọc có kích thước lẻ (nhỏ nhất là 3×3). Tổng quát, lọc không gian tuyến tính cho một ảnh có kích thước $M \times N$ với cửa sổ lọc là $m \times n$:

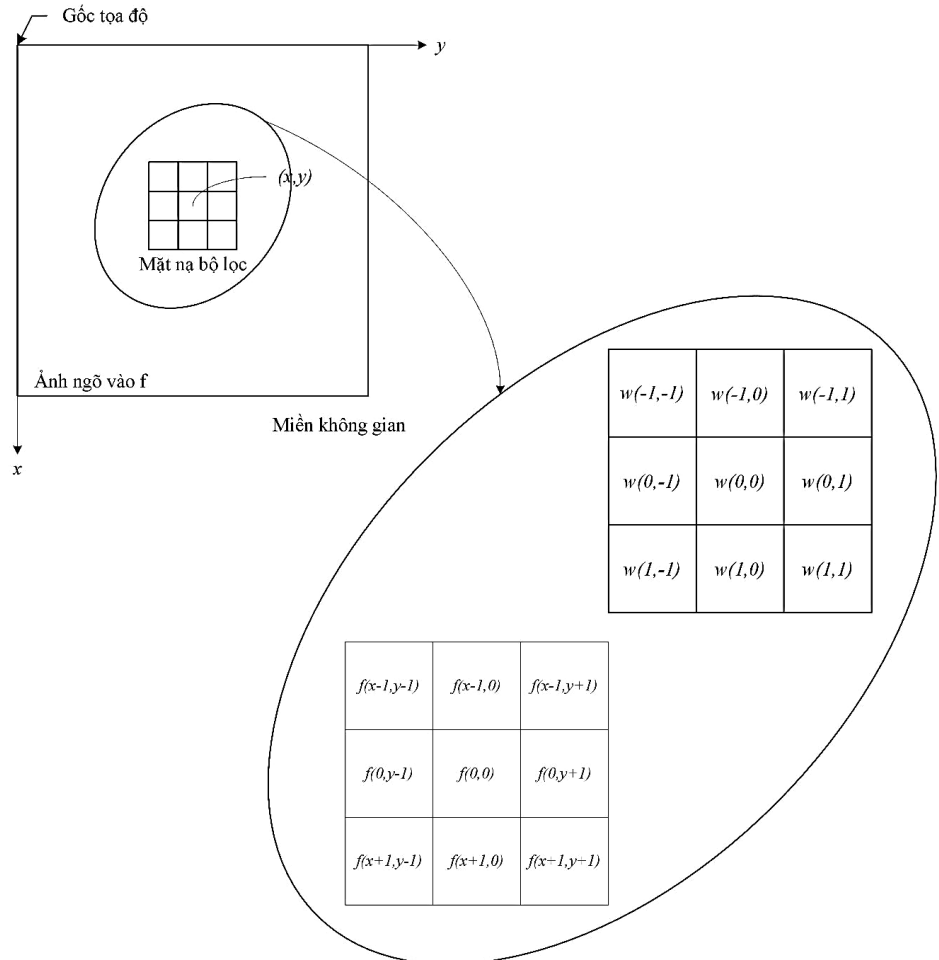
$$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t)f(x + s, y + t) \quad (4.2)$$

Với x và y được thay đổi sao cho mỗi điểm ảnh trong w đi qua từng điểm ảnh trong f .

Tương quan và tích chập không gian

Có hai khái niệm cần được hiểu rõ trước khi thực hiện lọc không gian: một là tính tương quan và hai là tích chập. Tương quan trong xử lý ảnh được thực hiện bằng cách di chuyển cửa sổ lọc trên toàn bộ ảnh và tính

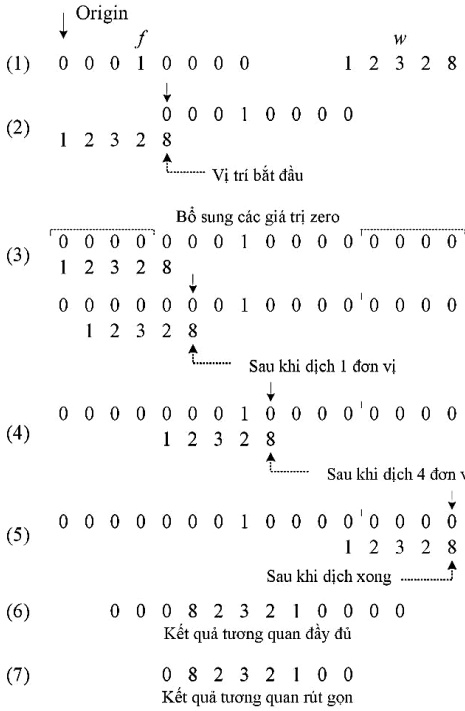
tổng của tích tại từng vị trí. Thực hiện tương tự khi tính tích chập, ngoại trừ bộ lọc được xoay 180⁰. Tham khảo ví dụ bên dưới trong hình 4.2 để hiểu rõ hơn về cách thực hiện các phép tính này.



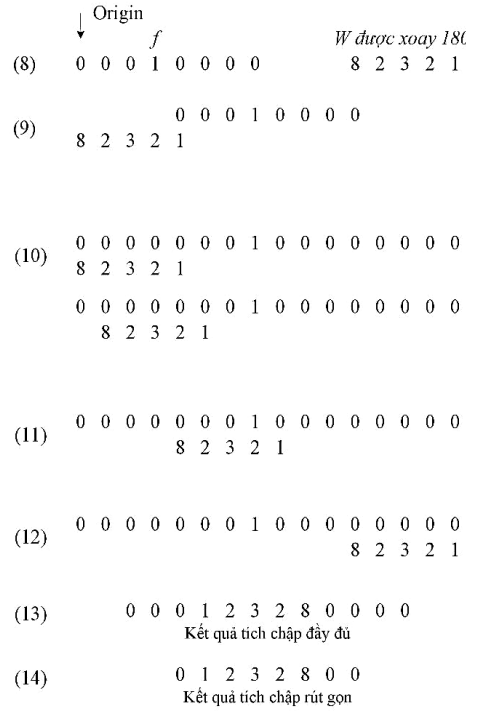
Hình 4.1. Cơ chế hoạt động của bộ lọc không gian tuyến tính sử dụng của số lọc 3 × 3

Có hai điểm cần lưu ý trong phần này: thứ nhất, tương quan được xem như một hàm di chuyển của bộ lọc như trong hình 4.2(a); thứ hai, tính tương quan của bộ lọc với hàm được tính toán tương tự như tương quan nhưng giá trị bộ lọc được xoay đi 180⁰ làm kết quả cũng được xoay 180⁰ như trong hình 4.2(b).

(a) Tính tương quan



(b) Tính tích chập



Hình 4.2. Mô tả phương pháp tính tương quan và tích chập 1 chiều với xung đơn vị dạng rời rạc. Lưu ý là tương quan hay tích chập đều là các hàm dạng chuyển vị.

Một cách tổng quát, chúng ta có tương quan của một bộ lọc $w(x, y)$ có kích thước $m \times n$ với ảnh $f(x, y)$ được cho, gọi là $w(x, y) \star f(x, y)$, được cho bởi công thức dưới đây:

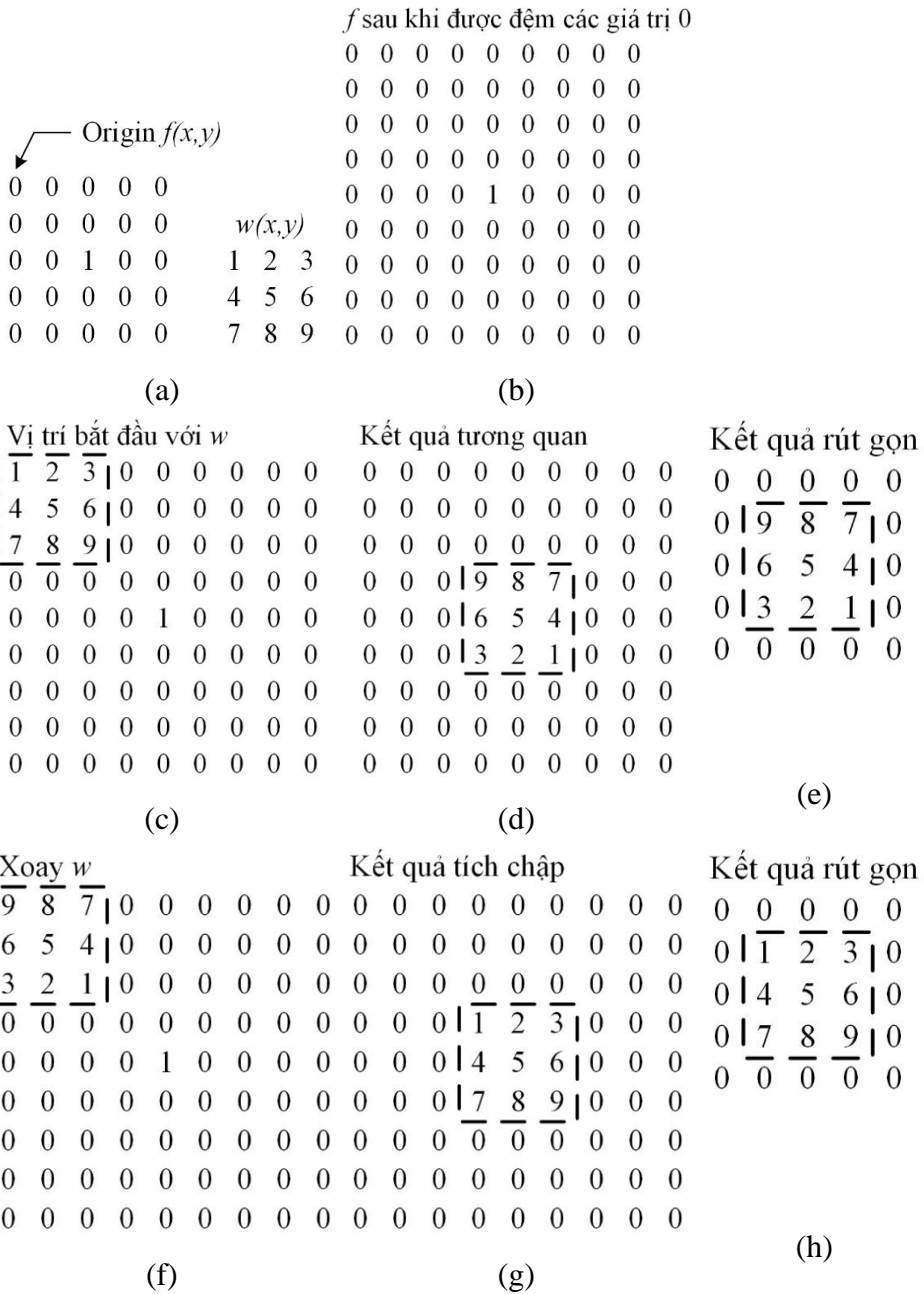
$$w(x, y) \star f(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t) \quad (4.3)$$

Biểu thức trên được thực hiện cho tất cả các giá trị của các biến x và y để đảm bảo rằng tất cả các phần tử trong w được di chuyển đến từng điểm ảnh trong f . Như đã đề cập trước đó, $a = (m - 1)/2$, $b = (n - 1)/2$ được giả định mang giá trị lẻ.

Với phương thức tương tự, tích chập của $w(x, y)$ và $f(x, y)$, được gọi là $w(x, y) \star f(x, y)$, được cho bởi công thức sau:

$$w(x, y) \star f(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x - s, y - t) \quad (4.4)$$

Dấu trừ trong biểu thức của f biểu diễn cho việc xoay kết quả đi 180° . Lật và xoay f thay vì w được thực hiện để cho việc tính toán được đơn giản và thuận tiện. Tham khảo thêm hình 4.3 mô tả việc thực hiện tính tương quan và tích chập dành cho bộ lọc 3×3 .



Hình 4.3. Tính tương quan (hàng giữa) và tích chập (hàng cuối) với bộ lọc hai chiều có kích thước 3×3 .

Trong MATLAB, hàm `imfilter` thực hiện phép lọc ảnh với số chiều tương ứng. Hàm `ones` cho phép tạo ra ma trận với các hệ số có giá trị bằng 1. Cú pháp hoàn chỉnh của hàm `imfilter` như sau:

```
g=imfilter(f,h,'Boundary Options','OutputSize','Correlation and Convolution Options')
```

với `f` là ảnh ngõ vào, `g` là ảnh ngõ ra, `h` là cửa sổ lọc, `'Boundary Options'` cho phép lựa chọn đường bao của ảnh ngõ ra, `'Output Size'` cho phép lựa chọn kích thước ảnh ngõ ra và cơ chế thực hiện bộ lọc được thay đổi dựa vào thông số `'Correlation and Convolution Options'`. Các thông số trên có thể tham khảo trong bảng 4.1

Bảng 4.1. Các thông số hàm `imfilter`

Tùy chọn	Mô tả
Chế độ lọc	
<code>'corr'</code>	Sử dụng phép toán tương quan cho quá trình lọc. Đây là chế độ mặc định
<code>'conv'</code>	Sử dụng phép toán tích chập cho quá trình lọc
Đường bao	
<code>P</code>	Các đường bao của ảnh được mở rộng bằng cách gia tăng thêm điểm ảnh có giá trị <code>P</code> . Mặc định <code>P</code> bằng 0.
<code>'replicate'</code>	Kích thước của ảnh được mở rộng bằng cách tạo một bản sao ngay đường biên chính nó
<code>'symmetric'</code>	Kích thước của ảnh được mở rộng bằng cách ánh xạ các giá trị mức xám thông qua đường biên
<code>'circular'</code>	Kích thước của ảnh được mở rộng bằng cách dàn trải ảnh như một hàm có chu kỳ 2-D
Kích thước ảnh	
<code>'full'</code>	Ảnh ngõ ra có cùng kích thước với ảnh đã được mở rộng
<code>'same'</code>	Ảnh ngõ ra giữ nguyên kích thước. Đây là tùy chọn mặc định

MATLAB hỗ trợ việc tạo ra một số bộ lọc tuyến tính 2-D bằng hàm `fspecial` có cú pháp như sau:

```
w=fspecial('type',parameters);
```

trong đó 'type' đánh dấu loại bộ lọc và `parameters` định nghĩa một số đặc trưng của bộ lọc. Các bộ lọc không gian này được hỗ trợ bởi hàm `fspecial` được tổng kết trong bảng 4.2, bao gồm các thông số đặc trưng của từng loại.

Bảng 4.2.

Type	Cú pháp và các thông số
'average'	<code>fspecial('average',[r c])</code> . Bộ lọc trung bình có kích thước $r \times c$. Mặc định là 3×3 .
'disk'	<code>fspecial('disk',r)</code> . Bộ lọc trung bình dạng hình tròn có bán kính r . Mặc định bán kính có giá trị là 5.
'gaussian'	<code>fspecial('gaussian',[r c],sig)</code> . Bộ lọc thông thấp Gaussian có kích thước $r \times c$ và độ lệch chuẩn sig (dương). Mặc định không khai báo thì $r \times c$ là 3×3 và $sig = 0.5$.
'laplacian'	<code>fspecial('laplacian',alpha)</code> . Bộ lọc Laplacian với hình dạng được đặc trưng bởi $alpha$ nằm trong khoảng $[0,1]$. Mặc định $alpha = 0.5$
'log'	<code>fspecial('log',[r c],sig)</code> . Bộ lọc Laplacian của hàm Gauss có kích thước $r \times c$ và độ lệch chuẩn sig (dương). Mặc định không khai báo thì $r \times c$ là 3×3 và $sig = 0.5$.
'motion'	<code>fspecial('motion',len,theta)</code> . Đây là bộ lọc giả định sự dịch chuyển của camera với len là số điểm ảnh bị dịch chuyển và $theta$ là hướng dịch chuyển. Mặc định thì len và $theta$ có giá trị lần lượt là 9 và 0.
'prewitt'	<code>fspecial('prewitt')</code> . Bộ lọc Prewitt 3×3 giúp tăng cường góc theo chiều ngang bằng phương pháp xấp xỉ các gradient theo chiều dọc. Chuyển vị bộ lọc để thu được kết quả tăng cường góc theo chiều dọc.
'sobel'	<code>fspecial('sobel')</code> . Bộ lọc Sobel 3×3 giúp tăng cường góc theo chiều ngang dùng hiệu ứng làm trơn bằng phương pháp xấp xỉ các gradient theo chiều dọc. Chuyển vị bộ lọc để thu được kết quả tăng cường góc theo chiều dọc.

Để hiểu rõ hơn các thành phần cấu tạo, một số bộ lọc được biểu diễn dưới dạng toán học:

- Với bộ lọc Gaussian

$$h_g(n_1, n_2) = e^{-\frac{(n_1^2 + n_2^2)}{2\sigma^2}} \quad (4.5)$$

$$h(n_1, n_2) = \frac{h_g(n_1, n_2)}{\sum n_1 \sum n_2 h_g} \quad (4.6)$$

- Với bộ lọc Laplacian

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \quad (4.7)$$

$$\begin{bmatrix} \frac{\alpha}{1+\alpha} & \frac{1-\alpha}{1+\alpha} & \frac{\alpha}{1+\alpha} \\ \frac{1-\alpha}{1+\alpha} & -4 & \frac{1-\alpha}{1+\alpha} \\ \frac{\alpha}{1+\alpha} & \frac{1-\alpha}{1+\alpha} & \frac{\alpha}{1+\alpha} \end{bmatrix} \quad (4.8)$$

- Với bộ lọc Log (Laplacian of Gaussian)

$$h(n_1, n_2) = \frac{(n_1^2 + n_2^2 - 2\sigma^2)h_g(n_1, n_2)}{2\pi\sigma^6 \sum n_1 \sum n_2 h_g} \quad (4.9)$$

- Mặt nạ của bộ lọc Prewitt

$$h = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad (4.10)$$

- Mặt nạ của bộ lọc Sobel

$$h = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (4.11)$$

Ví dụ 4.1: Thực hiện lọc tuyến tính với các bộ lọc có sẵn trong 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');
```

Trong thực tế, các bộ lọc này thường được sử dụng trong một chuỗi các tiến trình xử lý với mục đích cải thiện chất lượng ảnh, triệt nhiễu, tách biên và những ứng dụng khác. Do đó, rất ít khi thấy được ứng dụng độc lập được xây dựng từ các bộ lọc này.



(a)



(b)



(c)



(d)



(e)



(f)



(g)



(h)

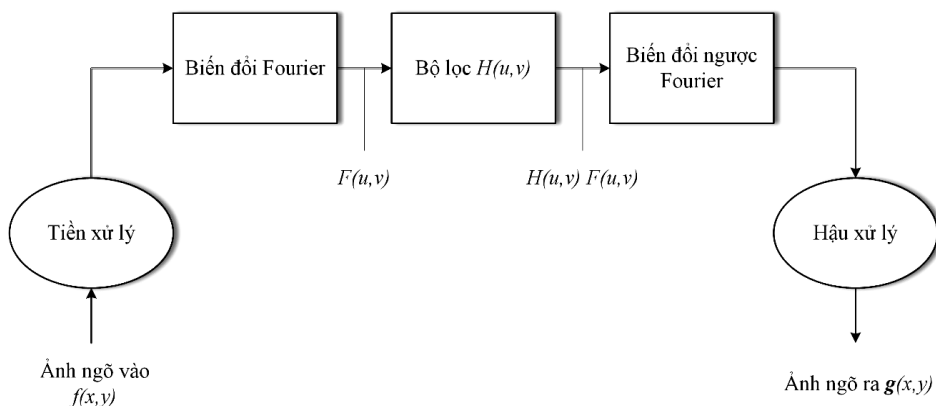


(i)

Hình 4.4. Thực hiện lọc ảnh với nhiều bộ lọc khác nhau
 (a) Ảnh gốc; (b) Average; (c) Disk; (d) Gaussian; (e) Laplacian
 (f) Log; (g) Motion; (h) Prewitt; (i) Sobel

4.2. LỌC TRONG MIỀN TẦN SỐ

Trong nhiều ứng dụng đòi hỏi cần phải xử lý trên miền tần số, biến đổi Fourier luôn là lựa chọn ưu tiên như một phép biến đổi phổ biến và dễ sử dụng. Một trong số nhiều ứng dụng phổ biến nhất là thực hiện lọc trong miền tần số, khi đó, ảnh được xem như tín hiệu hai chiều, cũng tồn tại các thành phần tần số khác nhau. Các bước thực hiện lọc trong miền tần số cơ bản được trình bày trong hình 4.5.



Hình 4.5. Các bước cơ bản khi thực hiện phép lọc trong miền tần số

Dựa trên các bước được trình bày trong hình 4.5, quá trình lọc trên miền tần số có thể được viết dưới dạng một hàm riêng trong MATLAB với ảnh ngõ vào và hàm lọc cho trước. Một điều cần lưu ý, hàm bộ lọc được sử dụng cũng phải được biến đổi về miền tần số.

```
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
```

Biến H ở đây là đáp ứng xung bộ lọc trên miền tần số được giả sử có kích thước gấp 4 lần ảnh ngõ vào. Trên miền không gian, quá trình lọc được xem xét như phép tính tích chập thì trên miền tần số là phép nhân trực tiếp giữa dữ liệu ngõ vào và giá trị bộ lọc. Sau khi thực hiện, ảnh ngõ ra thu được cần phải biến đổi ngược Fourier và lấy về đúng kích thước ban đầu.

Xây dựng bộ lọc tần số từ các bộ lọc không gian đã biết trước được thực hiện nhờ vào sự hỗ trợ của MATLAB với hàm `freqz2` với ngõ ra là các đáp ứng xung hữu hạn, vì thế, các bộ lọc được tạo ra từ hàm này còn được gọi là bộ lọc đáp ứng xung hữu hạn FIR (Finite Impulse Response). Cú pháp của hàm `freqz2` được mô tả như sau:

$$H = \text{freqz2}(h, R, C)$$

Với h là bộ lọc không gian hai chiều, H là bộ lọc đáp ứng tần số hai chiều. R và C là số hàng và cột mà ta mong muốn có ở H .

Ví dụ 4.2: Thực hiện việc lọc ảnh trong miền không gian dùng bộ lọc Sobel và lọc trong miền tần số dùng 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);
gs=uint8(gs);imshow(gs);
```

Để kiểm tra sự tương đồng giữa ảnh sau khi lọc bằng hai phương pháp khác nhau, chúng ta tính độ sai lệch theo biểu thức sau:

$$d = |gs - gd|$$

Trong đó gs là ảnh được lọc trực tiếp trên miền không gian, gd là ảnh được lọc trên miền tần số. Kết quả của ví dụ 4.2 được trình bày trong hình 4.6.



(a)



(b)



(c)

Hình 4.6. Lọc ảnh bằng bộ lọc Sobel

(a). Trên miền tần số

(b) Trên miền không gian

(c) Sai lệch giữa hai phương pháp

$$\max D = 3.4106 \times 10^{-13}$$

Do việc tính FFT trong MATLAB giả định góc tọa độ của hàm truyền tại vị trí góc trái phía trên của hình chữ nhật tần số, vì vậy, việc tính toán khoảng cách cần phải chú ý. Trong trường hợp này, góc tọa độ cần phải được sắp xếp lại sao cho góc nằm ngay tại trung tâm bằng cách sử dụng hàm `fftshift` như đã trình bày trước đó. Để cung cấp một sơ đồ lưới dùng tính toán khoảng cách và cho các ứng dụng tương tự, hàm `dftuv` được viết như sau:

```
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
```

Để minh họa, ta tính bình phương khoảng cách từ mỗi điểm trong hình chữ nhật có kích thước 8×5 đến góc:

```
[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

Giá trị khoảng cách bằng không tại góc trên bên trái và các giá trị lớn nhất tại trung tâm của hình chữ nhật được giải thích như ở hình 3.2(a). Hiển nhiên, ta có thể dùng hàm `fftshift` để đạt được các giá trị khoảng cách với mối liên hệ đến trung tâm của hình chữ nhật dùng biểu diễn tần số (frequency rectangle).

```
fftshift(D)
```

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
13	10	9	10	13

4.2.1. Lọc thông thấp

Bộ lọc thông thấp lý tưởng (Ideal LowPass Filter - ILPF) có hàm truyền như sau:

$$H(u, v) = \begin{cases} 1 & D(u, v) \leq D_0 \\ 0 & D(u, v) > D_0 \end{cases} \quad (4.12)$$

Với D_0 được chỉ định là một giá trị không âm, $D(u, v)$ là khoảng cách từ điểm (u, v) đến trung tâm của bộ lọc. Cần phải nhớ rằng quỹ đạo của các điểm $D(u, v) = D_0$ là hình tròn. Mặc dù đây không phải là bộ lọc thực tiễn do những yếu tố khách quan nhưng nó vẫn có thể được mô phỏng trên MATLAB bằng các hàm cho trước.

Bộ lọc thông thấp Butterworth (Butterworth LowPass Filter - BLPF) bậc n với tần số cắt tại giá trị khoảng cách D_0 , có hàm truyền được cho như sau:

$$H(u, v) = \frac{1}{1 + [D(u, v)/D_0]^{2n}} \quad (4.13)$$

Không giống như hàm ILPF, hàm BLPF không có được góc cắt tuyệt đối ngay tại D_0 . Để có được bộ lọc, thông thường người ta định nghĩa tần số cắt đặt tại các điểm mà $H(u, v) = 0.5$ (giảm 50% so với trị lớn nhất là 1) khi $D(u, v) = D_0$.

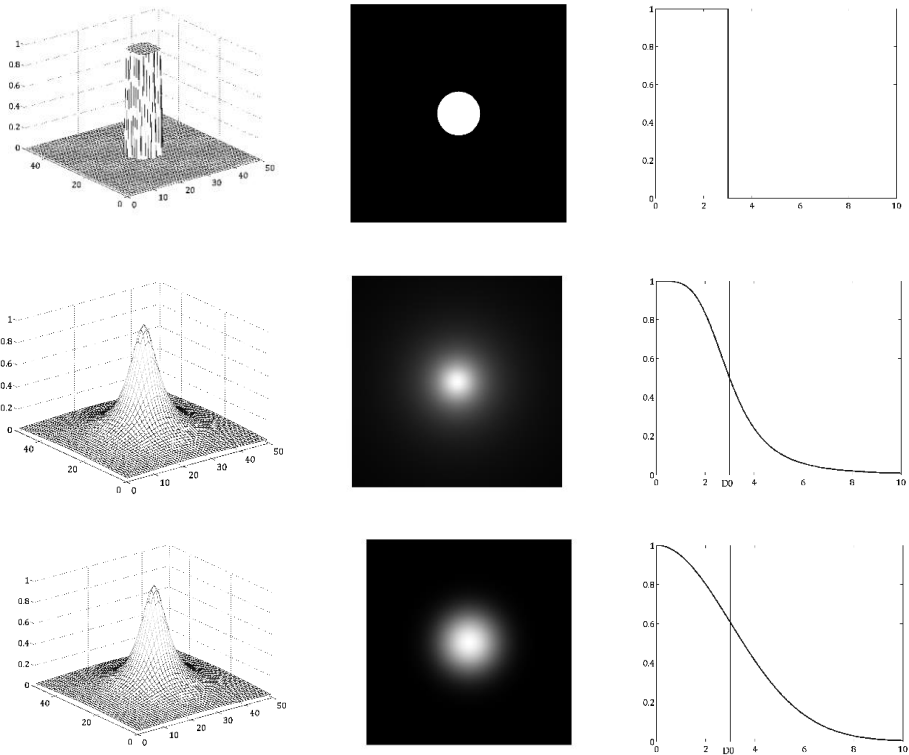
Hàm chuyển đổi của bộ lọc thông thấp Gaussian (GLPF) được cho như sau:

$$H(u, v) = e^{-D^2(u, v)/2\sigma^2} \quad (4.14)$$

Với σ là độ lệch chuẩn. Bằng cách cho $\sigma = D_0$, chúng ta có được dạng triển khai như sau:

$$H(u, v) = e^{-D^2(u, v)/2D_0^2} \quad (4.15)$$

Khi $D(u, v) = D_0$, giá trị ngay tại tần số cắt là 0.0607. Hình 4.7 mô tả dạng 3-D cũng như biểu đồ đáp ứng tần số của từng bộ lọc.



Hình 4.7. Cột trái: minh họa 3-D của đáp ứng của số lọc. Cột giữa: bộ lọc được biểu diễn như một ảnh. Cột phải: đáp ứng tần số với tần số cắt D_0 . Hàng đầu: lọc thông thấp lý tưởng. Hàng giữa: lọc thông thấp Butterworth. Hàng cuối: lọc thông thấp Gaussian.

Từ đây, ta có thể tự xây dựng một hàm chuyên dành để thực hiện lọc thông thấp `lpfilter` với ba bộ lọc cơ bản cho trên:

```
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

```

Một trong những phương pháp hữu hiệu để biểu diễn bộ lọc trong không gian ba chiều là sử dụng hàm `mesh` với cú pháp cho sau:

```
mesh(H)
```

Hàm này sẽ cho ra một biểu đồ với $x=1:M$ và $y=1:N$, với $[M,N]=\text{size}(H)$. Biểu đồ này về cơ bản là không chấp nhận mức độ dày đặc nếu M và N đều lớn, trong đó, chúng ta vẽ từng điểm bằng cách dùng hàm

```
mesh(H(1:k:end,1:k:end))
```

Mặc định thì biểu đồ dạng hàm `mesh` sẽ có màu, nhưng để đơn giản, chúng ta thay đổi sang dạng trắng đen bằng cách thiết lập

```
colormap([0 0 0])
```

Một số hàm thường sử dụng trong vẽ biểu đồ như: `grid on` (kích hoạt dạng lưới), `axis` (cho phép hiển thị hay không giá trị các trục).

Ví dụ 4.3: Vẽ biểu đồ dạng `mesh` cho hàm lọc thông thấp Butterworth với các giá trị bậc khác nhau.

```

clear all;
BLPF1=fftshift(lpfilter('btw',500,500,50,1));
mesh(BLPF1(1:10:500,1:10:500));
axis([0 50 0 50 0 1]);
saveas(gcf, 'Hình3.7a.tif', 'tif');

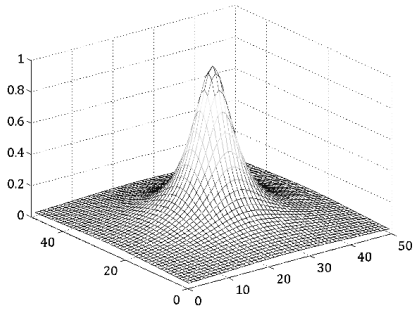
BLPF2=fftshift(lpfilter('btw',500,500,50,2));
figure;mesh(BLPF2(1:10:500,1:10:500));
axis([0 50 0 50 0 1]);
saveas(gcf, 'Hình3.7b.tif', 'tif');

```

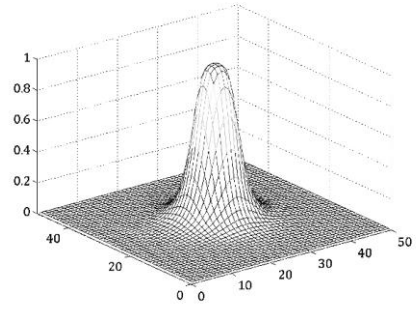


```
BLPF3=fftshift(lpfilter('btw',500,500,50,3));
figure;mesh(BLPF3(1:10:500,1:10:500));
axis([0 50 0 50 0 1]);
saveas(gcf, 'Hinh3.7c.tif', 'tif');
```

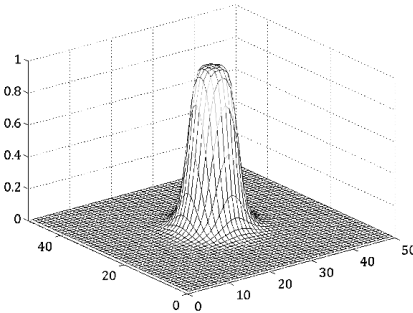
```
BLPF4=fftshift(lpfilter('btw',500,500,50,4));
figure;mesh(BLPF4(1:10:500,1:10:500));
axis([0 50 0 50 0 1]);
saveas(gcf, 'Hinh3.7d.tif', 'tif');
```



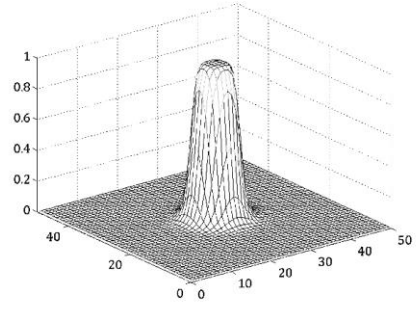
(a)



(b)



(c)



(d)

Hình 4.8. Biểu đồ dạng mesh của các bộ lọc Butterworth với bậc khác nhau

(a) Bậc một

(b) Bậc hai

(c) Bậc ba

(d) Bậc bốn

Ví dụ 4.4: Khảo sát bộ lọc thông thấp với nhiều tần số cắt khác nhau. So sánh kết quả.

```
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);

```



(a)



(b)



(c)



(d)



(e)



(f)

Hình 4.9. Ảnh cameraman $M \times N$ được lọc thông thấp dùng bộ lọc lý tưởng với tần số cắt khác nhau

(a) Ảnh gốc

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

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

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

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

$$(f) D_0 = M$$

4.2.2. Lọc thông cao

Không như bộ lọc thông thấp làm mờ ảnh, bộ lọc thông cao thường dùng trong các ứng dụng cần làm ảnh trở nên sắc nét hơn bằng cách loại bỏ thành phần tần số thấp thông qua biến đổi Fourier. Cho bộ lọc thông thấp có hàm truyền $H_{lp}(u, v)$, chúng ta dễ dàng có được bộ lọc thông cao bằng cách đơn giản sau:

$$H_{hp}(u, v) = 1 - H_{lp}(u, v) \quad (4.16)$$

Do đó, với hàm `lpfilter` được xây dựng trước đó, ta hoàn toàn có thể sử dụng lại để xây dựng hàm dành riêng cho lọc thông cao

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

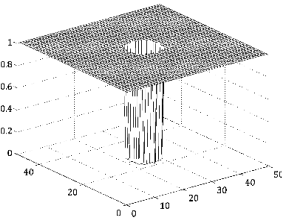
```
Hlp=lpfilter(type,M,N,D0,n);
H=1-Hlp;
end
```

Ví dụ 4.5: Biểu diễn các bộ lọc thông cao dưới dạng mesh và ảnh.

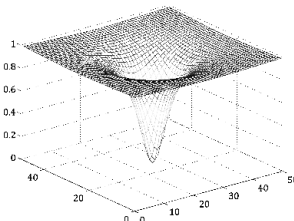
```
clear all;close all;clc
ILPF=fftshift(hpfilter('ideal',500,500,50));
mesh(ILPF(1:10:500,1:10:500));
axis([0 50 0 50 0 1]);

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

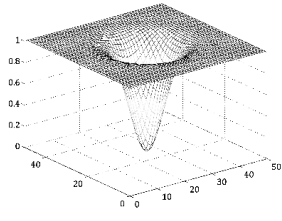
GLPF=fftshift(hpfilter('gaussian',500,500,50));
mesh(GLPF(1:10:500,1:10:500));
axis([0 50 0 50 0 1]);
```



(a)



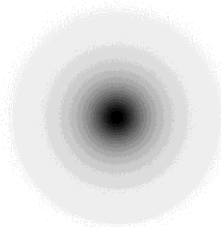
(b)



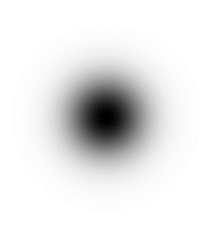
(c)



(d)



(e)



(f)

Hình 4.10. Biểu diễn các bộ lọc thông cao. Hàng trên: biểu diễn dạng sơ đồ mesh. Hàng dưới: biểu diễn dưới dạng ảnh. Cột trái: bộ lọc thông cao lý tưởng. Cột giữa: bộ lọc thông cao Butterworth. Cột phải: bộ lọc thông cao Gaussian

Các bộ lọc thông cao chuẩn hóa về không các thành phần một chiều dc, do đó, vô tình làm giảm trị trung bình của ảnh về không. Để giải quyết vấn đề này chúng ta thêm vào thành phần offset một chiều vào bộ lọc thông cao. Khi thành phần offset này kết hợp cùng phép nhân giữa bộ lọc và hằng số lớn hơn 1, phương pháp này gọi là lọc thông cao tăng cường (high frequency emphasis filtering) do hằng số nhân làm nổi bật các tần số cao. Hàm biểu diễn cho quá trình lọc thông cao tăng cường được trình bày như sau:

$$H_{hfe}(u, v) = a + bH_{hp}(u, v) \quad (4.17)$$

Với a là thành phần offset, b là hằng số nhân và $H_{hp}(u, v)$ là hàm truyền của bộ lọc thông cao.

Ví dụ 4.6: Thực hiện lọc ảnh dùng bộ lọc thông cao tăng cường.

```
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);
```

Trong ví dụ 4.6, hàm `gscale` thực hiện việc chuẩn hóa và đổi kiểu dữ liệu chứa cường độ mức xám của ảnh. Hàm `gscale` này có thể tham khảo như dưới đây:

```
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
```

Hình 4.11 trình bày kết quả trong ví dụ 4.6 với ảnh gốc satellite khá mờ. Hình 4.11(b) trình bày ảnh sau khi lọc thông cao dùng bộ lọc Butterworth, trong khi đó, ảnh sau khi dùng bộ lọc thông cao tăng cường được trình bày trong hình 4.11(c). Ưu điểm của bộ lọc này là một số thành phần tần số thấp vẫn còn được giữ lại, và chúng được làm rõ hơn bằng thuật toán cân bằng histogram như trong hình 4.11(d).

Hình 4.11.

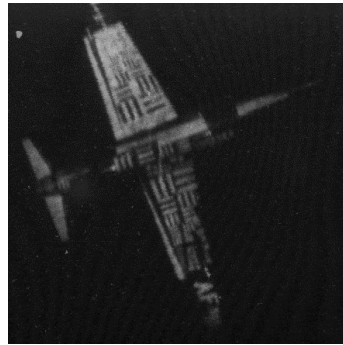
Lọc thông cao ảnh Satellite

(a) Ảnh gốc

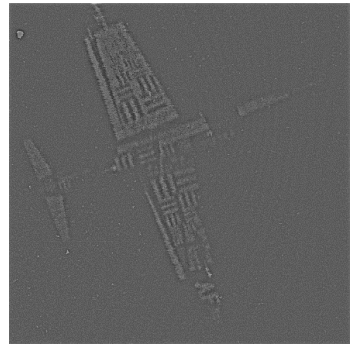
(b) Lọc thông cao Butterworth

(c) Lọc thông cao tăng cường

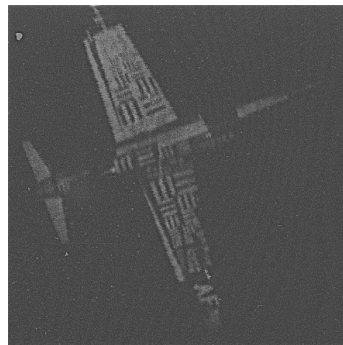
(d) Cân bằng histogram ảnh (c)



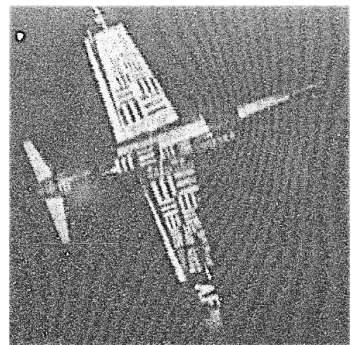
(a)



(b)



(c)



(d)

BÀI TẬP CHƯƠNG 4

- 4.1. Trình bày ma trận mặt nạ khác nhau có được từ hàm f_{special} . Vẽ đáp ứng tần số của các bộ lọc này và cho nhận xét.
- 4.2. Thực hiện lọc thông thấp Butterworth và Gaussian với nhiều tần số cắt khác nhau. Từ đó rút ra kết luận.
- 4.3. Ảnh được cho bị ảnh hưởng bởi nhiễu Gaussian. Thực hiện lọc thông thấp lý tưởng cho ảnh. Khảo sát chất lượng ảnh sau khi lọc dùng thông số PSNR.
- 4.4. Vẽ đáp ứng tần số của bộ lọc thông thấp và thông cao với cùng tần số cắt. Cho nhận xét.

TÀI LIỆU THAM KHẢO

1. Nguyễn Quang Hoan, *Xử lý ảnh*, lưu hành nội bộ, Học viện Công nghệ Bưu chính Viễn thông, 2006.
2. Đỗ Năng Toàn, Phạm Việt Bình, *Xử lý ảnh*, Giáo trình môn học, Đại học Thái Nguyên, Khoa Công nghệ thông tin, 2007.
3. Maria Petrou, Panagiota Bosdogianni, *Image Processing: The Fundamentals*, John Wiley & Sons Ltd, 1999.
4. William K. Pratt, *Digital Image Processing*, John Wiley & Sons, Inc., 2001.
5. Rafael C. Gonzalez, Richard E. Woods, *Digital Image Processing*, The Third Edition, Prentice Hall, 2008.
6. Rafael C. Gonzalez, Richard E. Woods, *Digital Image Processing Using Matlab*, Prentice Hall, 2004.
7. Kayvan N., Robert S., *Biomedical Signal and Image Processing*, Taylor and Francis Group, 2006.

GIÁO TRÌNH XỬ LÝ ẢNH

TS. NGUYỄN THANH HẢI

NHÀ XUẤT BẢN
ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
Khu Phố 6, Phường Linh Trung, Quận Thủ Đức, TPHCM
Số 3, Công trường Quốc tế, Quận 3, TP Hồ Chí Minh
ĐT: 38239171 – 38225227 – 38239172
Fax: 38239172 - Email: vnuhp@vnuhcm.edu.vn

PHÒNG PHÁT HÀNH NHÀ XUẤT BẢN
ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
Số 3 Công trường Quốc tế - Quận 3 – TPHCM
ĐT: 38239170 – 0982920509 – 0913943466
Fax: 38239172 – Website: www.nxbdhqghcm.edu.vn

Chịu trách nhiệm xuất bản:
NGUYỄN HOÀNG DŨNG

Chịu trách nhiệm nội dung:
NGUYỄN HOÀNG DŨNG

Tổ chức bản thảo và chịu trách nhiệm về tác quyền
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TPHCM

Biên tập:
PHẠM ANH TÚ

Sửa bản in:
THÙY DƯƠNG

Trình bày bìa
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TPHCM

Mã số ISBN: 978-604-73-2582-5

Số lượng 300 cuốn; khổ 16 x 24cm.

Số đăng ký kế hoạch xuất bản: 1007-2014/CXB/05-13/ĐHQGTPHCM.

Quyết định xuất bản số: 107 ngày 28/05/2014 của NXB ĐHQGTPHCM.

In tại Công ty TNHH In và Bao bì Hưng Phú.

Nộp lưu chiểu quý III năm 2014.