

## BÀI 3: XỬ LÝ LÂN CẬN (LỌC ẢNH TRONG MIỀN KHÔNG GIAN)

### I. Lọc tuyến tính

#### 1. Hàm `imfilter` và hàm `fspecial`

##### 1.1 Lọc tuyến tính với `imfilter`

- Hàm `imfilter` (<https://www.mathworks.com/help/images/ref/imfilter.html>) của MATLAB IPT cho phép thực hiện lọc ảnh tuyến tính bằng tương quan (mặc định) hoặc tích chập.

- Cú pháp:

```
g = imfilter(f, h, mode, boundary_options, size_options);
```

Trong đó:

- + `g`: ảnh đầu ra
- + `f`: ảnh đầu vào
- + `h`: bộ lọc tuyến tính
- + `mode` có thể là `'conv'` hoặc `'corr'`, tương ứng, cho biết việc lọc sẽ được thực hiện bằng cách sử dụng tích chập hoặc tương quan (mặc định);
- + `boundary_options` đề cập đến cách thuật toán lọc sẽ xử lý các giá trị đường viền ảnh. Có bốn khả năng:

- ✓ **X**: Các biên của mảng đầu vào (hình ảnh) được mở rộng bằng cách đệm với giá trị X. Đây là tùy chọn mặc định (với X = 0).
- ✓ **'symmetric'**: Các biên của mảng đầu vào (hình ảnh) được mở rộng bằng cách ánh xạ hình ảnh qua đường viền của nó.
- ✓ **'replicate'**: Các biên của mảng đầu vào (hình ảnh) được mở rộng bằng cách sao chép các giá trị gần nhất với đường viền hình ảnh.
- ✓ **'circular'**: Các biên của mảng đầu vào (hình ảnh) được mở rộng bằng cách giả định rằng mảng đầu vào là tuần hoàn, nghĩa là coi hình ảnh là một chu kỳ của hàm tuần hoàn 2D.

+ **size\_options**: Có hai tùy chọn cho kích thước của ảnh kết quả: `'full'` (hình ảnh đầu ra là kết quả được lọc đầy đủ, nghĩa là kích thước của ảnh được mở rộng / đệm) hoặc `'same'` (hình ảnh đầu ra có cùng kích thước với hình ảnh đầu vào hình ảnh). Tùy chọn thứ hai là tùy chọn mặc định.

##### 1.2 Tạo bộ lọc với hàm `fspecial`

- `fspecial` (<https://www.mathworks.com/help/images/ref/fspecial.html>) là một hàm IPT được thiết kế để đơn giản hóa việc tạo những bộ lọc ảnh 2 chiều phổ biến.

- Cú pháp: `h = fspecial(type, parameters)`

Trong đó:

+ **h** là mặt nạ lọc.

+ **type** là một trong những giá trị sau đây:

- ✓ **'average'**: bộ lọc trung bình
- ✓ **'disk'**: Bộ lọc trung bình tròn
- ✓ **'gaussian'**: Bộ lọc thông thấp Gaussian
- ✓ **'laplacian'**: Toán tử Laplace 2D
- ✓ **'log'**: Bộ lọc Laplacian of Gaussian (LoG)
- ✓ **'motion'**: Xấp xỉ chuyển động tuyến tính của máy ảnh
- ✓ **'prewitt'** và **'sobel'**: Bộ lọc nhân mạnh biên ngang
- ✓ **'unsharp'**: bộ lọc tăng cường tương phản unsharp

**parameters** là các tham số tùy chọn khác nhau tùy thuộc vào kiểu bộ lọc (type), ví dụ: kích thước mặt nạ, độ lệch chuẩn (đối với bộ lọc 'gaussian')

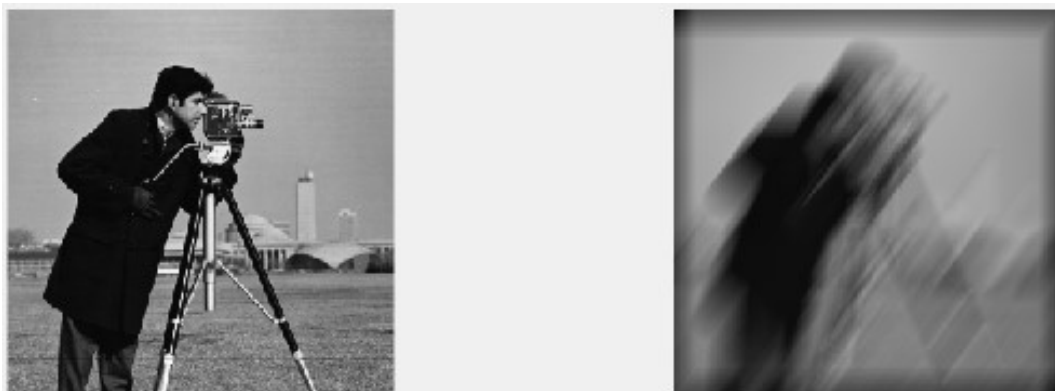
### 1.3 Ví dụ, kết hợp giữa hàm *imfilter* và *fspecial*

- Tạo bộ lọc motion (tương tự hiệu ứng chuyển động mờ), với chiều dài (len) của chuyển động là 50 và góc của chuyển động (theta) theo hướng ngược chiều kim đồng hồ là 54. Bộ lọc trở thành một vector của các chuyển động ngang và dọc. Giá trị mặc định của len là 9 và theta là 0, tương ứng với chuyển động ngang của 9 pixels.

```
>> img=imread('B3_images\cameraman.tif');
>> h=fspecial('motion',50,54);
>> f=imfilter(img,h,'conv');
>> subplot(121);
>> imshow(img);
>> subplot(122);
>> imshow(f);
```

Ở đây, tùy chọn cho `boundary_options` của `imfilter` đang ở chế độ mặc định, có nghĩa thay thế giá trị các điểm ảnh trên đường viền ảnh bằng giá trị 0.

Kết quả thu được:



Yêu cầu:

- Lần lượt tạo ra các bộ lọc khác nhau bằng hàm **fspecial** và sử dụng các bộ lọc này cho việc lọc các ảnh trong gói dữ liệu ảnh B2\_images.
- Điều chỉnh giá trị cho các tùy chọn `boudary_options` và `size_options` của hàm **imfilter** để thấy được những thay đổi có thể có trên ảnh đầu ra.

## 2. Hàm **conv2** và **filter2**

- Hàm **conv2** (<https://www.mathworks.com/help/matlab/ref/conv2.html>): tính toán tích chập 2D giữa hai ma trận. Tham số thứ ba của hàm **conv2** chỉ định kích thước của đầu ra:
  - + `full`: Trả về tích chập 2D đầy đủ (mặc định).
  - + `same`: Trả về phần trung tâm của tích chập có cùng kích thước với A.
  - + `valid`: Chỉ trả về những phần đã tính tích chập, không bao gồm các đường viền ảnh được đệm bằng không.
- Hàm **filter2** (<https://www.mathworks.com/help/matlab/ref/filter2.html>): Nó xoay mặt nạ tích chập (được coi là bộ lọc FIR 2D) 180° theo mỗi hướng để tạo nhân tích chập và sau đó gọi **conv2** để thực hiện thao tác tích chập.

Cú pháp: `Y = filter2(H,X)`

`Y = filter2(H,X,shape)`

Trong đó: **H**: bộ lọc; **X**: ảnh đầu vào; **shape**: tùy chọn kích thước đầu ra của **conv2**.

- Ví dụ: lần lượt áp dụng **filter2** với các bộ lọc sau:

Lọc thông thấp	Lọc thông cao	Phát hiện biên ngang
$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & 1 \end{bmatrix}$

**Lọc thông thấp:**

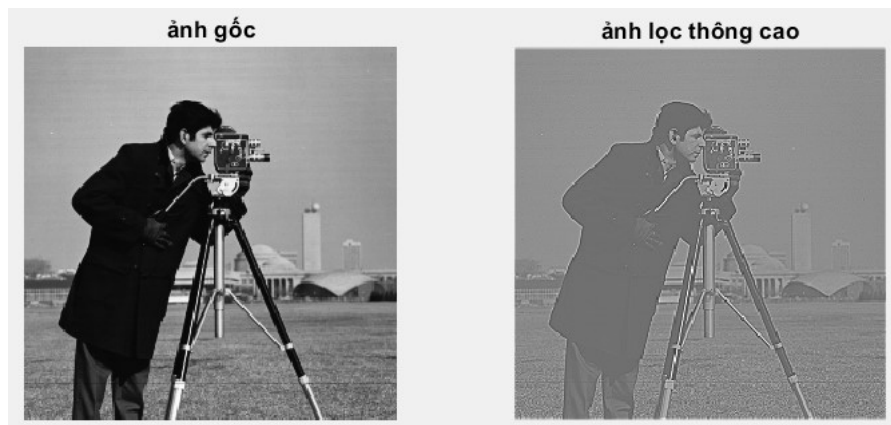
```
>> img=imread('B3_images\cameraman.tif');
>> low=ones(3)/9;
>> f2=filter2(low,img);
>> subplot(121);
>> imshow(img);
>> title('ảnh gốc');
>> subplot(122);
>> imshow(f2,[]);
>> title('ảnh lọc thông thấp');
```

Kết quả thu được:

**Lọc thông cao:**

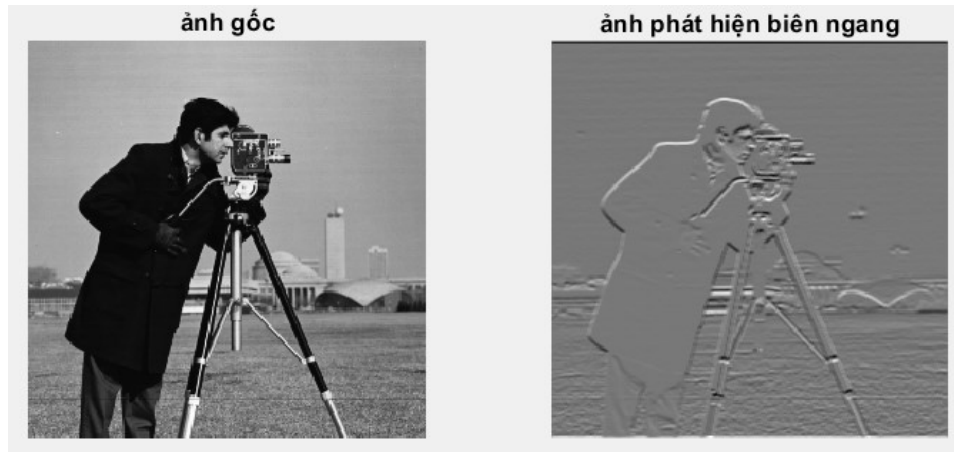
```
>> img=imread('B3_images\cameraman.tif');  
>> high=[0 -1 0;-1 5 -1; 0 -1 0];  
>> f2=filter2(high,img);  
>> subplot(121);  
>> imshow(img);  
>> title('ảnh gốc');  
>> subplot(122);  
>> imshow(f2, []);  
>> title('ảnh lọc thông cao');
```

Kết quả thu được:

**Phát hiện biên ngang:**

```
>> img=imread('B3_images\cameraman.tif');  
>> hor=[1 1 1;0 0 0; -1 -1 -1];  
>> f2=filter2(hor,img);  
>> subplot(121);  
>> imshow(img);  
>> title('ảnh gốc');  
>> subplot(122);  
>> imshow(f2, []);  
>> title('ảnh phát hiện biên ngang');
```

Kết quả thu được:



**Yêu cầu:** Xây dựng ứng dụng cho phép đọc một ảnh, xuất kết quả lọc ảnh đầu vào với các bộ lọc trong phần ví dụ trên.

### 3. Bộ lọc trung bình

- Lọc làm mịn ảnh (có độ nhòe lớn).
- Là bộ lọc có kích thước lẻ ( $3 \times 3, 5 \times 5, 7 \times 7, \dots$ ) với trọng số của các phần tử bằng nhau và là số dương. Tổng của các trọng số bằng 1.

- Tạo bộ lọc trung bình bằng:

+ Hàm `fspecial` theo cú pháp:

```
h = fspecial('average', hsize)
```

+ Hàm `ones` theo cú pháp: `h=ones(hsize)/(hsize*hsize)`

- Lọc ảnh bằng hàm **`imfilter`** hoặc **`filter2`**

**Yêu cầu:** Thao tác với Command Window và Editor, sử dụng gói dữ liệu ảnh `B3_images`.

- Sử dụng hàm `fspecial` tạo bộ lọc trung bình kích thước  $3 \times 3$ , sử dụng bộ lọc kết hợp với hàm `filter2`. Kiểm tra kết quả với phần Lọc thông thấp trong ví dụ trong phần 2.

- Sử dụng hàm `fspecial` tạo bộ lọc trung bình kích thước  $3 \times 3$ , sử dụng bộ lọc kết hợp với hàm `imfilter`, mode: giá trị mặc định. Kiểm tra kết quả so với hàm `filter2`.

- Xây dựng hàm **`locTrungBinh(file, hsize)`** cho phép thực hiện chức năng lọc trung bình ảnh đầu vào. Kiểm tra tính đúng đắn của hàm trong Command Window.

### 4. -5Bộ lọc Gaussian

- Lọc làm mịn ảnh (ít nhòe hơn lọc trung bình).
- Là bộ lọc có kích thước lẻ ( $3 \times 3, 5 \times 5, 7 \times 7, \dots$ ) với trọng số của các phần tử là số dương và phân phối theo hàm số Gauss. Tổng của các trọng số bằng 1.

- Tạo bộ lọc làm mịn Gaussian bằng hàm **`fspecial`** theo cú pháp:

```
h = fspecial('gaussian', hsize, sigma)
```



- Sau đó, lọc ảnh bằng hàm **imfilter** hoặc **filter2** (không khuyến khích).
- Lọc làm mịn gaussian với hàm **imgaussfilt** theo cú pháp:

```
B = imgaussfilt(A)
B = imgaussfilt(A, sigma)
B = imgaussfilt(____, Name, Value)
```

Tham khảo liên kết: <https://www.mathworks.com/help/images/ref/imgaussfilt.html>

- Ví dụ: lọc gaussian với hàm **imgaussfilt**

```
>> img=imread('B3_images\cameraman.tif');
>> oimg=imgaussfilt(img,1.0,'FilterSize',3,'Padding',0);
>> subplot(121);
>> imshow(img);
>> title('ảnh gốc');
>> subplot(122);
>> imshow(oimg);
>> title('ảnh lọc thông thấp gaussian');
```

Kết quả thu được:



**Yêu cầu:** Thao tác với Command Window và Editor, sử dụng gói dữ liệu ảnh B3\_images.

- Dùng hàm **fspecial** tạo bộ lọc làm mịn Gaussian, dùng hàm **imfilter** hoặc **filter2** lọc ảnh đầu vào với bộ lọc vừa tạo.

- Điều chỉnh các thông số của hàm **imgaussfilt**, kiểm tra những thay đổi trên các ảnh đầu ra. So sánh kết quả lọc Gaussian làm mịn ảnh bằng hàm **imgaussfilt** với cách thực hiện trên.

## 5. Lọc làm nét với toán tử Laplacian

## II. Lọc phi tuyến

### 1. Hàm **ordfilt2**

- Không áp dụng tích chập.
- Thực hiện tính toán giá trị mới cho điểm ảnh đầu ra bằng một phương pháp phi tuyến tính.

- Hàm **ordfilt2** (<https://www.mathworks.com/help/images/ref/ordfilt2.html>) thực hiện lọc thống kê có thứ tự 2 chiều.

Cú pháp của **ordfilt2**:

$B = \text{ordfilt2}(A, \text{order}, \text{domain})$

$B = \text{ordfilt2}(A, \text{order}, \text{domain}, S)$

$B = \text{ordfilt2}(\_, \text{padopt})$

Trong đó:

+ **B**: ảnh đầu ra

+ **A**: ảnh đầu vào

+ **order**: phần tử để thay thế pixel mục tiêu, được chỉ định dưới dạng số nguyên.

+ **domain**: xác định vùng lân cận cần xét, được chỉ định dưới dạng ma trận các giá trị số hoặc luận lý chứa các số 1 và 0. Domain tương đương với phần tử cấu trúc được dùng trong các phép toán ảnh nhị phân. Một thành phần giá trị 1 xác định vùng lân cận cho phép lọc ảnh. Một số ví dụ về domain của các bộ lọc phổ biến.

Bộ lọc	Mã lệnh	Vùng lân cận	Ví dụ vdữ liệu ảnh và chỉ định thành phần được chọn																		
Bộ lọc Median (Trung vị)	B = ordfilt2(A,5,ones(3,3))	<table><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	1	1	1	1	1	1	1	1	1	<table><tr><td>88</td><td>16</td><td>56</td></tr><tr><td>5</td><td>3</td><td>30</td></tr><tr><td>21</td><td>63</td><td>42</td></tr></table>	88	16	56	5	3	30	21	63	42
1	1	1																			
1	1	1																			
1	1	1																			
88	16	56																			
5	3	30																			
21	63	42																			
Bộ lọc Minimum	B = ordfilt2(A,1,ones(3,3))	<table><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	1	1	1	1	1	1	1	1	1	<table><tr><td>88</td><td>16</td><td>56</td></tr><tr><td>5</td><td>3</td><td>30</td></tr><tr><td>21</td><td>63</td><td>42</td></tr></table>	88	16	56	5	3	30	21	63	42
1	1	1																			
1	1	1																			
1	1	1																			
88	16	56																			
5	3	30																			
21	63	42																			
Bộ lọc Maximum	B = ordfilt2(A,9,ones(3,3))	<table><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	1	1	1	1	1	1	1	1	1	<table><tr><td>88</td><td>16</td><td>56</td></tr><tr><td>5</td><td>3</td><td>30</td></tr><tr><td>21</td><td>63</td><td>42</td></tr></table>	88	16	56	5	3	30	21	63	42
1	1	1																			
1	1	1																			
1	1	1																			
88	16	56																			
5	3	30																			
21	63	42																			

Giá trị nhỏ nhất của vùng lân cận 4	$B = \text{ordfilt2}(A, 1, [0 \ 1 \ 0; 1 \ 0 \ 1; 0 \ 1 \ 0])$	<table><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr></table>	0	1	0	1	0	1	0	1	0	<table><tr><td>88</td><td>16</td><td>56</td></tr><tr><td>5</td><td>3</td><td>30</td></tr><tr><td>21</td><td>63</td><td>42</td></tr></table>	88	16	56	5	3	30	21	63	42
0	1	0																			
1	0	1																			
0	1	0																			
88	16	56																			
5	3	30																			
21	63	42																			

+ **padopt**: tùy chọn vùng đệm (đường viền ảnh), chọn 1 trong các giá trị: 'zeros': thiết lập bằng 0; 'symmetric': lấy giá trị phản xạ gương của chính nó.

**Yêu cầu:** Thao tác với Command Window và Editor, sử dụng gói dữ liệu ảnh B3\_images.

- Đọc ảnh và lọc ảnh phi tuyến với hàm `ordfilt2` và sử dụng các domain như trong bảng trên. So sánh và đánh giá các kết quả thu được.

## 2. Bộ lọc trung vị (Median Filter) với hàm `medfilt2`

- Lọc trung vị là một phép phi tuyến tính thường được dùng trong xử lý ảnh để giảm nhiễu muối tiêu ("salt and pepper" noise). Bộ lọc trung vị hiệu quả hơn so với các bộ lọc tuyến tính trong trường hợp muốn loại bỏ nhiễu nhưng vẫn bảo toàn đặc tính các biên.

- IPT hỗ trợ hàm `medfilt2` (<https://www.mathworks.com/help/images/ref/medfilt2.html>) cho phép lọc trung vị trên ảnh.

- Cú pháp:

```
J = medfilt2(I)
J = medfilt2(I, [m n])
J = medfilt2(___, padopt)
```

- Mô tả:

`J = medfilt2(I)` lọc trung vị với kích thước vùng lân cận mặc định là  $3 \times 3$

`J = medfilt2(I, [m n])` lọc trung vị với kích thước vùng lân cận là  $m \times n$

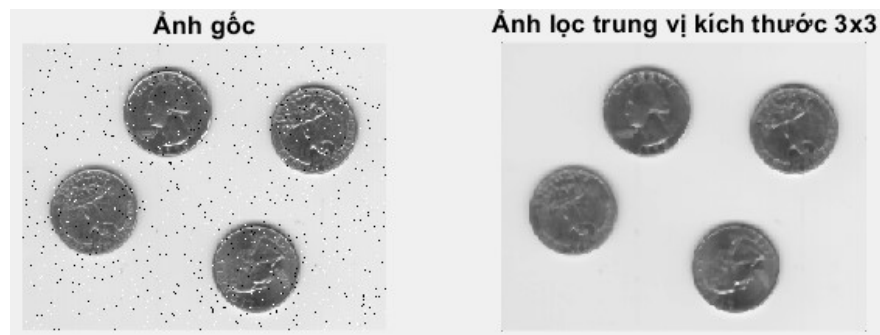
`J = medfilt2(___, padopt)` lọc trung vị với tùy chọn `padopt` chỉ định cách thức xử lý đường viền ảnh là một trong các giá trị 'zeros' (mặc định), 'symmetric' hoặc 'indexed'.

-Ví dụ: Nhập các lệnh sau trong Command Window

```
>> img=imread('B3_images\eight_salt&pepp_prob2.png');
>> oim=medfilt2(img);
>> subplot(121);
>> imshow(img);
>> title("Ảnh gốc");
>> subplot(122);
>> imshow(oim);
>> title("Ảnh lọc trung vị kích thước 3x3");
```

Kết quả thu được:





**Yêu cầu:** Thao tác với Command Window và Editor, sử dụng gói dữ liệu ảnh B3\_images.

- Lọc các ảnh nhiễu trong gói dữ liệu ảnh bằng hàm `medfilt2` với các kích thước bộ lọc khác nhau. So sánh đánh giá giữa các kết quả thu được.

- Xây dựng hàm **Loctrungvi(file, hsize)** thực hiện lọc trung vị ảnh theo giải thuật đã học. Kiểm tra tính đúng đắn của hàm trong Command Window.

### III. Xây dựng ứng dụng cho phép thực hiện các phép lọc ảnh trong miền không gian

- Phát họa giao diện và hành vi của ứng dụng.
- Thiết kế giao diện theo bản phát họa.
- Viết code thực thi hành vi của ứng dụng.
- Lưu ý, cần cho phép người dùng chọn các bộ lọc và điều chỉnh các tham số tùy chọn của bộ lọc (nếu cần) như kích thước bộ lọc, độ lệch chuẩn của bộ lọc Gaussian...