Lab

Chương 1 BIỂU DIỄN ẢNH

1.1 Dùng MATLAB *đọc, hiển thị ảnh* cùng với *truy xuất* vị trí của pixel.

- Chẳng hạn: `I(i,j) = I(i,j)+25` hoặc `I(i,j) = I(i,j)+25`
- Bắt đầu với việc đọc ảnh "`cell.tif`" và vị trí pixel (100,20). Ảnh hưởng như thế nào trên mức xám của ảnh đối với phép toán "+" và "-".

1.2 Dựa trên câu trả lời của câu 1.1, sử dụng cấu trúc lặp for để lặp lại phép tính trên đối với tất cả pixel của ảnh I để tạo ra hai ảnh I_1, I_2

`I1(i,j) = I(i,j)+25;` hoặc `I2(i,j) = I(i,j)+25;`

- ***Chú ý***: Anh/chị chắc chắn rằng chương trình của mình không làm giá trị của pixel lớn hơn 255 hay nhỏ hơn 0.

1.3 Đọc ảnh "`cameraman.tif`" và dùng hàm imwrite để ghi lại một ảnh có định dang JPEG (chẳng hạn: `Ijpg.jpg`) và một ảnh có định dạng PNG (chẳng hạn: `Ipng.png`). Sau đó đọc hai ảnh `Ijpg.jpg` và `Ipng.png` vào hai biến tương ứng `Ijpg` và `Ipng`.

- Điều chúng ta mong đợi là hai ảnh trên giống nhau hoàn toàn? Nếu chúng ta so sánh chúng bằng cách trừ tương ứng từng pixel của hai ảnh cho nhau thì liệu này có đúng không?

```
X=imabsdiff(Ijpg,Ipng);
imagesc(x);
```

- Chúng ta thấy rằng sự khác nhau giữa chúng không phải là zero như ta mong đợi. Vì, định dạng JPEG là nén mất thông tin (lossy compression) còn định dạng png là nén không mất thông tin (lossless compression).

Chương 2 NÂNG CAO CHẤT LƯỢNG ẢNH

2.1 Cho một ảnh đa mức xám r='Fig0304(a)(breast_digital_Xray).tif'. Yêu cầu thực hiện các phép xử lý sau:

a. Phân tích sự phân bố mức xám của ảnh

b. Viết code để biến đổi âm bản ảnh trên

s_a=L-1-r. Trong đó r ảnh đầu vào, L cấp mức xám.

c. so sánh lược đồ xám ảnh S_a và r, giải thích.

d. chọn ngưỡng t=127. Viết code để tạo ra ảnh nhị phân từ ảnh r.

2.2 Cho ảnh đa mức xám i='Fig0122(a)(fractal-iris).tif'. Yêu cầu thực hiện các phép xử lý làm mỏng mặt phẳng bit:

a. Tạo ảnh i3 từ bit plane thứ 3 của ảnh I (dùng hàm bitget)

b. Tạo ảnh i6 từ bit plane thứ 6 của ảnh i

c. Tạo ảnh i78 từ bit plane thứ 7, thứ 8 của ảnh I (dùng hàm bitget và bitset)

Ví dụ: *– tạo ảnh i4 từ bit plane thứ 4 của ảnh I ảnh*

```
i=imread('..\images\Fig0122(a)(fractal-iris).tif');
i4=bitget(i,4);%lấy bit plane thứ 4
```

– tạo ảnh i67 từ bit plane thứ 6 và 7 của ảnh I

```
i67=zeros(size(i));
i67=bitset(i67,6,bitget(i,6));%% lấy  bit thứ 7 của
                               a đặt vào bit thứ 7
                               của b78
i67=bitset(i67,7,bitget(i,7));%% lấy  bit thứ 8  của
                               a đặt vào bit thứ 8
                               của b78
```

2.3 Cân bằng mức xám, cho ảnh 256 cấp xám i= 'Fig0316(4)(bottom_left).tif':

a. Xem ảnh, phân tích lược đồ xám ảnh i.

b. Cân lược đồ xám ảnh I, dùng hàm histeq(i,gray-scale level)

c. xem ảnh vừa được cân bằng mức xám và lược đồ xám của ảnh.

2.4 Các phép toán số học trên ảnh

	Matlab code	Ý nghĩa

1	A=imread('cameraman.tif ');
	subplot(1,2,1), imshow(A);
	B=imadd(A, 100);
	subplot(1,2,2), imshow(B);
		%đọc ảnh nạp vào biến A
		%hiển thị ảnh A
		%cộng 100 vào mỗi giá trị pixel của A
		% hiển thị ảnh B

2	A=imread('cola1.png');
	B¼imread('cola2.png');
	subplot(1,3,1), imshow(A);
	subplot(1,3,2), imshow(B);
	Output = imsubtract(A, B);
	subplot(1,3,3),imshow(Output);
	
	Output1 = imabsdiff(A, B);
	subplot(1,3,3),imshow(Output1);
		%đọc ảnh thứ nhất
		% đọc ảnh thứ hai
		%hiển thị ảnh thứ nhất
		%hiển thị ảnh thứ hai
		%imsubtract(A,B) ~ A-B
		%hiển thị ảnh kết quả
		
		%trừ ảnh
		%hiển thị ảnh kết quả 1

3	Output = immultiply(A,1.5); subplot(1,3,3), imshow(Output); Output = imdivide(A,4); subplot(1,3,3), imshow(Output);	%nhân ảnh với 1.5 %hiển thị ảnh kết quả %chia gia trị pixel với 4 %hiển thị ảnh kết quả
Ghi chú: Ta có thể dùng hàm imagesc() để xem ảnh kết quả		

4	A=imread('toycars1.png');
	B=imread('toycars2.png');
	Abw=im2bw(A);
	Bbw=im2bw(B);
	subplot(1,3,1), imshow(Abw);
	subplot(1,3,2), imshow(Bbw);
	Output = xor(Abw, Bbw);
	subplot(1,3,3), imshow(Output);
		%Read in 1st image
		%Read in 2nd image
		%convert to binary
		%convert to binary
		%Display 1st image
		%Display 2nd image
		%xor images images
		%Display result

Ghi chú: • im2bw(): hàm convert ảnh sang ảnh nhị phân dùng ngưỡng tự động		

5*	I=imread('trees.tif');
	T=im2bw(I, 0.1);
	subplot(1,3,1), imshow(I);
	subplot(1,3,2), imshow(T);
		%Read in 1st image
		%Perform thresholding, ngưỡng 0->1
		%Display original image
		%Display thresholded image

2.5 Biến đổi logarit

	Matlab code	Ý nghĩa

1	I=imread('cameraman.tif ');
	subplot(2,2,1), imshow(I);
	Id=im2double(I);
	
	Output1=2*log(1+Id);
	Output2=3*log(1+Id);
	Output3=5*log(1+Id);
	
	subplot(2,2,2),imshow(Output1);
		%biến đổi logarit

	subplot(2,2,3),imshow(Output2); subplot(2,2,4),imshow(Output3);	

2.6 Phân tích ngưỡng

	Matlab code	Ý nghĩa

1	I=imread('coins.png');
	subplot(1,2,1), imshow(I);
	subplot(1,2,2), imhist(I);
		%Vẽ lược đồ xám ảnh I

	I=imread('coins.png'); [counts,bins]=imhist(I); counts(60)	%đếm số mức xám %truy vấn giá trị xám 60

2.7 Sử dụng kỹ thuật chọn ngưỡng ở đoạn code 5* và ví dụ ở câu 2.6. xác định và áp ngưỡng đối với ảnh `pillsetc.png`. Tương tự cho những ảnh '`tape.png`', '`coins.png`' và '`eight.tif`'.

2.8 Phép trừ ảnh $g(x,y)=f(x,y)-h(x,y)$

a. cho ảnh `f='Fig0122(a)(fractal-iris).tif'`,soạn code file `.m` tạo ảnh **`h`** bằng cách đặt 4 plane bit thấp của ảnh `f` bằng 0.

b. Tạo ảnh `g` từ việc trừ hai ảnh `f` và `h` cho nhau theo từng pixel tương ứng.

c. Tạo ảnh `i` từ việc cân bằng mức xám của ảnh `g`.

2.9 Lọc tuyến tính: tổng của các tích giữa hệ số của bộ lọc với cấp xám của điểm ảnh tương ứng trong vùng áp dụng lọc.

$$R = w_1 z_1 + w_2 z_2 + \ldots + w_{mn} z_{mn} = \sum_{i=1}^{mn} w_i z_i$$

Trong MATLAB, sử dụng hàm sau để lọc tuyến tính

`g=imfilter(f , w , filtering_mode , boundary_options , size_options)`

Trong đó: **`f`** là ảnh đầu vào, **`w`** là mặt nạ lọc, **`g`** ảnh đầu ra (kết quả), `filtering_mode , boundary_options , size_options` như bảng sau:

Lựa chọn	Mô tả
Filtering_mode	
'corr'	Lọc dùng ***Correlation***
'conv'	Lọc dùng ***Convolution***
Boundary options	
p	Mở rộng biên cho ảnh khi chập với giá trị p, mặc nhiên 0
'replicate'	Mở rộng biên cho ảnh bằng việc lập lại giá trị biên
'symmetric'	Kích thước ảnh được mở rộng bằng cách phản chiếu ngang qua biên
'circular'	Kích thước ảnh được mở rộng bằng cách xem ảnh như hàm chu kỳ
Size option	
'full'	ảnh đầu ra có cùng kích cở với ảnh được mở rộng
'same'	ảnh đầu ra có cùng kích cở với ảnh đầu vào

☞ ***Yêu cầu: lọc nhiễu ảnh:***

`i='Fig0333(a)(test_pattern_blurring_orig).tif'`

với mặt nạ lọc:

$$w = \frac{1}{3}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

2.10 Lọc phi tuyến – lọc trung vị

Vì tính chất quan trọng của phép lọc MATLAB phát triển sẵn hàm lọc phi tuyến:

```
g=medfilter2(f, [m n], padopt);
```

Trong đó: **[m n]** kích thước lân cận *m*×*n*, padopt chỉ định 1 trong 3 kiểu biên: '*zeros*' (mặc định), '*symmetric*'(ảnh *f* được mở rộng bằng việc phản chiếu những giá trị pixel ngang qua biên), '*indexed*'(ảnh *f* được mở rộng bằng việc thêm ngoài biên những giá trị 1).

☞ ***Yêu cầu:***

a. nạp ảnh `i='coins.png'`, gây nhiễu 'muối -tiêu'cho ảnh 3% (hàm *imnoise*), xem ảnh.

b. lọc trung vị hàm *medfilt2*. Xem ảnh.

c. Gây nhiễu '*gaussain*' 2%, lọc ảnh, xem ảnh

2.11 Áp dụng bộ lọc sau với `'Fig0335(a)(ckt_board_saltpep_prob_pt05).tif'`:

a. Lọc tuyến tính.

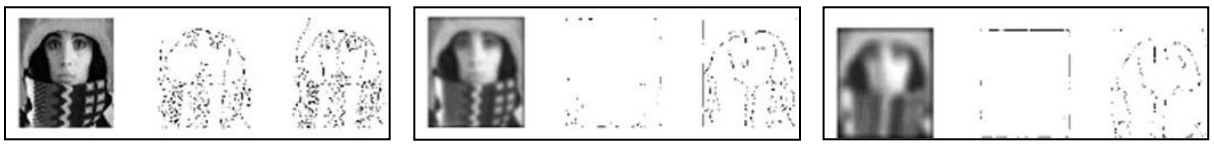
b. Lọc phi tuyến ➔ so sánh 2 phương pháp lọc, nhận xét.

Chương 3&4 PHÁT HIỆN BIÊN VÀ PHÂN ĐOẠN ẢNH

3.1 Phát hiện biên Canny

Matlab code	Ý nghĩa
A=imread('trui.png');	
subplot(3,3,1), imshow(A,[]);	
h1=fspecial('gaussian',[15 15],6);	
h2=fspecial('gaussian',[30 30],12);	
subplot(3,3,4), imshow(imfilter(A,h1),[]);	%Xem ảnh với lọc sigma=6
subplot(3,3,7), imshow(imfilter(A,h2),[]);	%Xem ảnh với lọc sigma=12
[bw,thresh]=edge(A,'log');	%Phát hiện biên với bộ lọc logarit
subplot(3,3,2), imshow(bw,[]);	
[bw,thresh]=edge(A,'canny');	%Lọc Canny
subplot(3,3,3), imshow(bw,[]);	%hiển thị ảnh
[bw,thresh]=edge(imfilter(A,h1),'log');	%Lọc logarit với sigma=6
subplot(3,3,5), imshow(bw,[]);	
[bw,thresh]=edge(imfilter(A,h1),'canny');	%Lọc Canny với sigma=6
subplot(3,3,6), imshow(bw,[]);	
[bw,thresh]=edge(imfilter(A,h2),'log');	%Lọc logarit với gigma=12
subplot(3,3,8), imshow(bw,[]);	
[bw,thresh]=edge(imfilter(A,h2),'canny');	%Lọc Canny với sigma=6
subplot(3,3,9), imshow(bw,[]);	

***Kết quả**



không lọc | logarit | Canny | σ=6 | logarit | Canny | σ=12 | logarit | Canny

3.2 Phát hiện biên

- Các mặt nạ phát hiện biên

-1	-1	-1
2	2	2
-1	-1	-1

ngang

-1	-1	2
-1	2	-1
2	-1	-1

$+45^0$

2	-1	-1
-1	2	-1
-1	-1	2

dọc

2	-1	-1
-1	2	-1
-1	-1	2

-45^0

- Hàm: `imfilter(f,w)`, trong đó *f*: ảnh đầu vào; *w*: mặt nạ

```
w = [-1 -1 -1;2 2 2;-1 -1 -1] % tạo ma trận w(3×3)
f = imread('Fig1005(a)(wirebond_mask).tif');
g = imfilter(f,w);
imshow(f);
figure, imshow(g);
```

- Thực hiện tương tự cho các mặt nạ $+45^0$, dọc, -45^0
- Nhận xét các ảnh thu nhận được?

3.3 Phát hiện biên bằng phương pháp 'sobel', 'prewitt', 'roberts'

- Hàm [g, t]=edge(f, 'method', parameters);
- Trong đó:
  - *f* ảnh đầu vào,
  - method= {sobel, prewitt, roberts};
  - g ảnh đầu ra
  - t ngưỡng
- ví dụ:

```
f=imread('Fig1016(a)(building_original).tif');
[g, t]=edge(f, 'sobel', 'vertical');
imshow(f);
figure, imshow(g);
```

- Thực hiện tương tự cho những phương 'prewitt', 'roberts'.