

Họ và tên: Lê Quang Duy

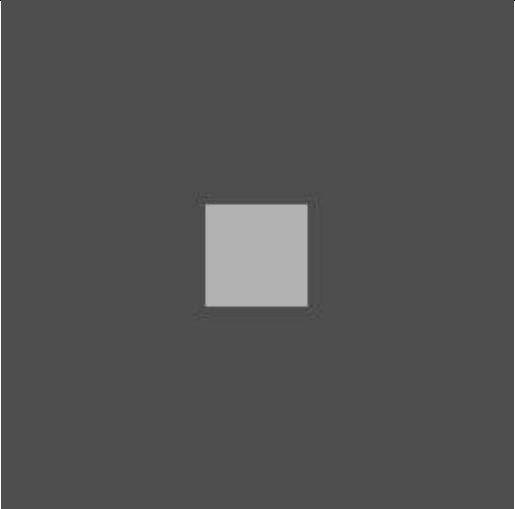
Lớp: CNTT1-K59

Mã sinh viên: 18120346

Báo cáo thực hành 2 - Xử lý ảnh

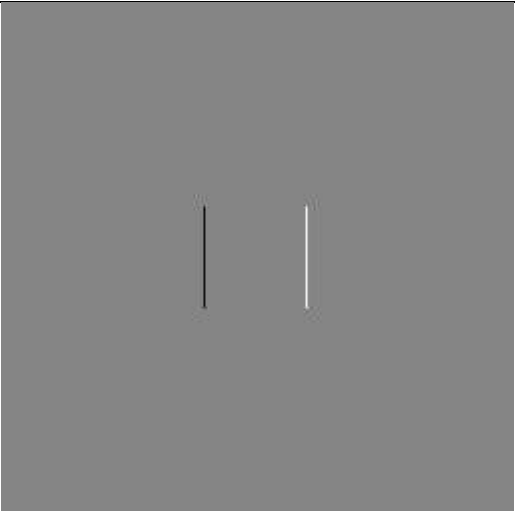
Phần 2.2:

Câu 1:

Code	Hình ảnh
<pre>X = ones(256,256)*(128-50) [w,h] = size(X) box = 50 X(w/2-(box/2):w/2+(box/2),h/2-(box/2):h/2+(box/2)) = 128+50 imshow(X/255) imwrite(X/255, 'e1.jpg')</pre>	

Câu 2:

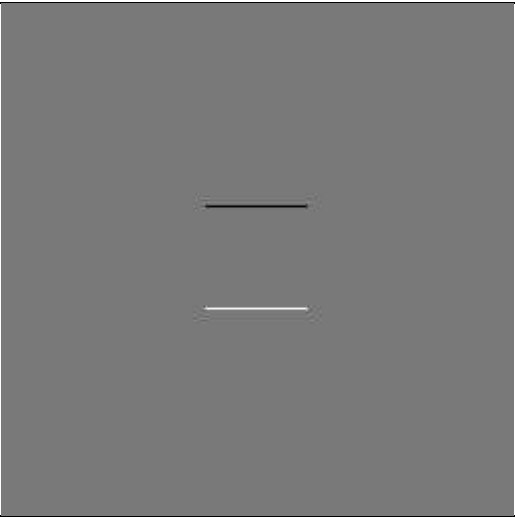
a.

Code	Hình ảnh
<pre>X = imread('/home/duy/port/Lab02/thuchanh2/e1.jpg') X = double(X) h1 = [1 -1] Y1 = imfilter(X,h1,'replicate') imwrite(Y1,'e2.png') imshow(mat2gray(Y1)) imwrite(mat2gray(Y1),'e2show.png')</pre>	

- Nhận xét: bộ lọc h1 làm nổi đường viền theo hướng dọc, bộ lọc này tốt cho phát hiện thay đổi ảnh từ sáng ang tối và cả tối sang sáng.

b.

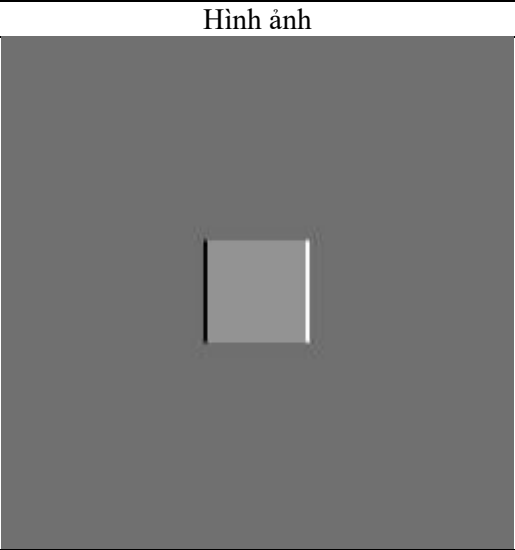
Code	Hình ảnh
------	----------

<pre> X = imread('/home/duy/port/Lab02/thuchanh2/e1.jpg ') X = double(X) h2 = [1 -1]' Y2 = imfilter(X,h2,'replicate') imwrite(Y2,'e2_.png') imshow(mat2gray(Y2)) imwrite(mat2gray(Y2),'e2show_b.png') </pre>		
---	--	---

- Nhận xét: bộ lọc h2 làm nổi đường viền theo hướng ngang, bộ lọc này tốt cho phát hiện thay đổi ảnh từ sáng ang tối và cả tối sang sáng.

Câu 3:

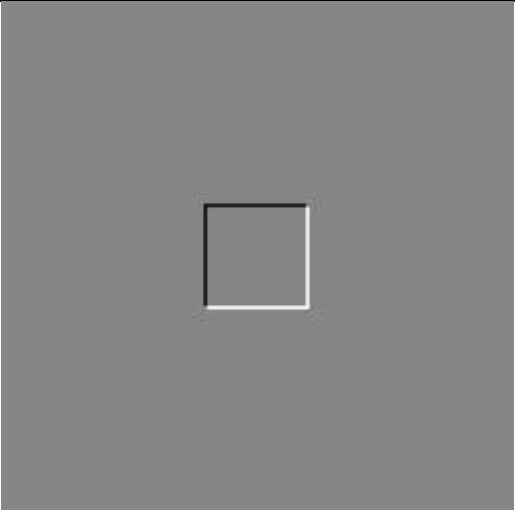
a.

Code	Hình ảnh
<pre> X = imread('/home/duy/port/Lab02/thuchanh2/e1.jpg ') X = double(X) h3 = [1 0 -1; 2 0 -1; 1 0 -1] Y3 = imfilter(X,h3,'replicate') imwrite(Y3,'e3.png') imshow(mat2gray(Y3)) imwrite(mat2gray(Y3),'e3show.png') </pre>	

- Nhận xét: bộ lọc h3 phát hiện đường viền theo hướng ngang và dọc, bộ lọc này tốt để phát hiện đường viền thay đổi từ cả sáng sang tối và cả tối sang sáng.

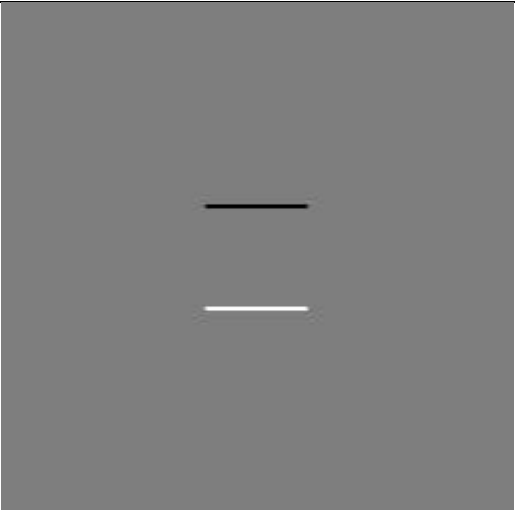
b.

Code	Hình ảnh
------	----------

<pre> X = imread('/home/duy/port/Lab02/thuchanh2/e1.jpg ') X = double(X) h4 = [2 1 0; 1 0 -1; 0 -1 -2] Y4 = imfilter(X,h4,'replicate') imwrite(Y4,'e4.png') imshow(mat2gray(Y4)) imwrite(mat2gray(Y4),'e4show.png') </pre>	
---	---

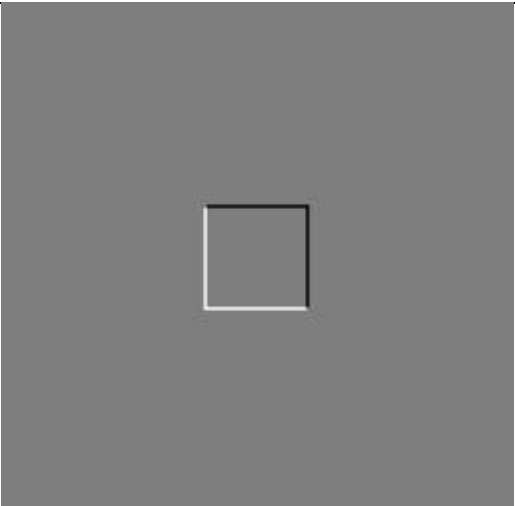
- Nhận xét: bộ lọc h4 phát hiện đường viền theo hướng ngang và dọc, bộ lọc này tốt để phát hiện đường viền thay đổi từ cả sáng sang tối và cả tối sang sáng.

c.

Code	Hình ảnh
<pre> X = imread('/home/duy/port/Lab02/thuchanh2/e1.jpg ') X = double(X) h5 = [1 2 1; 0 0 0; -1 -2 -1] Y5 = imfilter(X,h5,'replicate') imwrite(Y5,'e5.png') imshow(mat2gray(Y5)) imwrite(mat2gray(Y5),'e5show.png') </pre>	

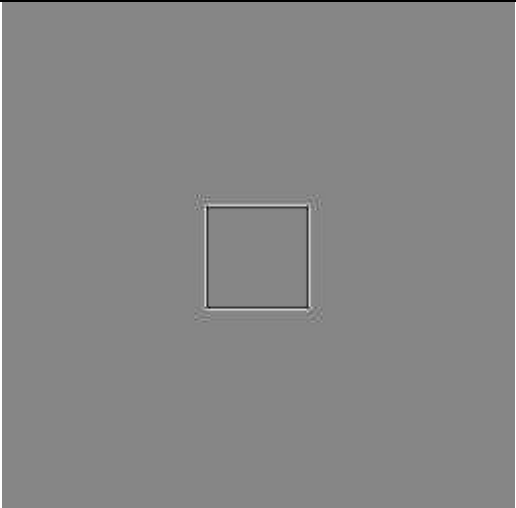
- Nhận xét: Bộ lọc h5 phát hiện đường viền theo hướng ngang, bộ lọc này tốt cho cả thay đổi từ sáng sang tối và từ tối sang sáng.

d.

Code	Hình ảnh
<pre> X = imread('/home/duy/port/Lab02/thuchanh2/e1.jpg ') X = double(X) h6 = [0 1 2; -1 0 1; -2 -1 0] Y6 = imfilter(X,h6,'replicate') imwrite(Y6,'e6.png') imshow(mat2gray(Y6)) imwrite(mat2gray(Y6),'e6show.png') </pre>	

- Nhận xét: Bộ lọc h4 làm thay đổi đường viền theo hướng ngang và dọc, bộ lọc này tốt cho thay đổi từ sáng sang tối và cả tối sang sáng.

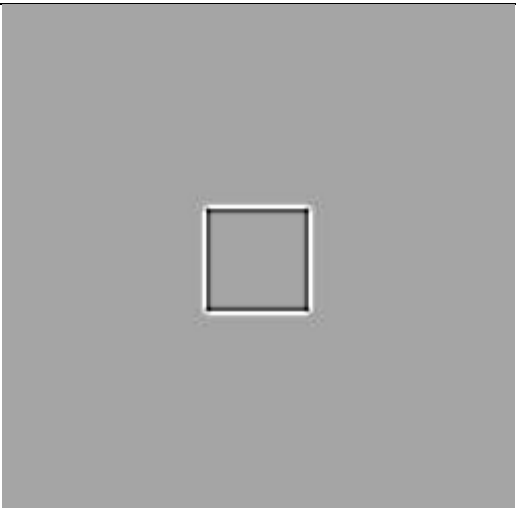
Câu 4:

Code	Hình ảnh
<pre>X = imread('/home/duy/port/Lab02/thuchanh2/e1.jpg ') X = double(X) h7 = 1/8 * [0 1 0; 1 -4 1; 0 1 0] Y7 = imfilter(X,h7,'replicate') imwrite(Y7,'e7.png') imshow(mat2gray(Y7)) imwrite(mat2gray(Y7),'e7show.png')</pre>	

- Nhận xét: bộ lọc h7 làm nổi đường viền theo cả hướng ngang và dọc, bộ lọc này tốt cho cả thay đổi từ sáng sang tối.

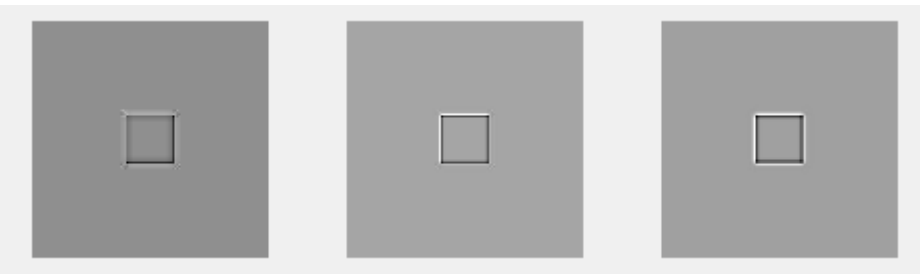
Câu 5:

a.

Code	Hình ảnh
<pre>X = imread('/home/duy/port/Lab02/thuchanh2/e1.jpg ') X = double(X) h8 = fspecial('log', [15 15], 1) Y8 = imfilter(X,h8,'replicate') imwrite(Y8,'e8.png') imshow(mat2gray(Y8)) imwrite(mat2gray(Y8),'e8show.png')</pre>	

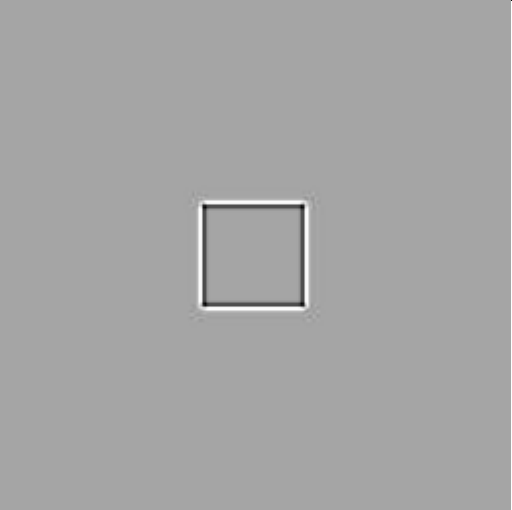
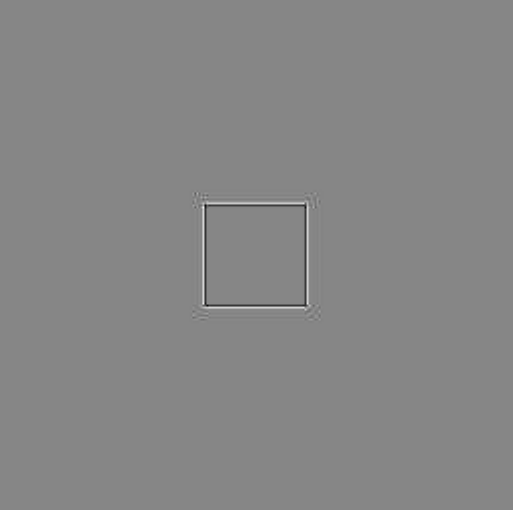
- Nhận xét: bộ lọc h8 phát hiện đường viền theo cả hướng ngang và dọc, tốt cho phát hiện thay đổi từ sáng sang tối và từ tối sang sáng.

b. Thay đổi phương sai 0.5 -> 1 -> 1.5

Hình ảnh


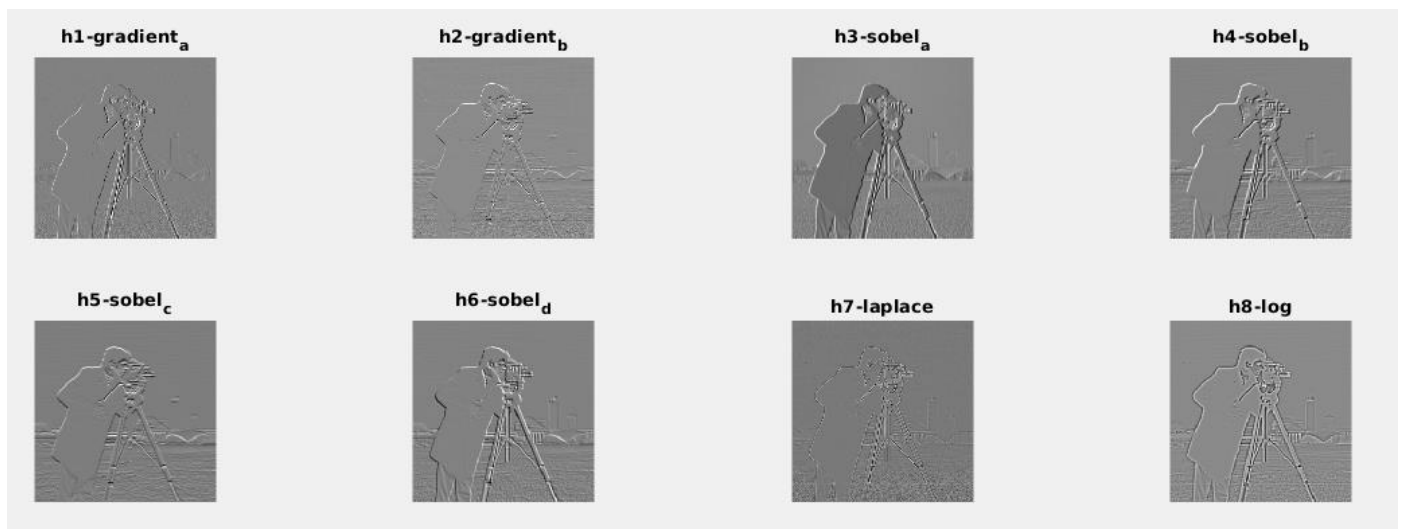
- Nhận xét; Khi tăng phương sai, phát hiện thay đổi càng nhạy cảm hơn, khi giảm phương sai, phát hiện thay đổi độ sáng ít đi.

c.

Bộ lọc log (5a)		Bộ lọc laplace	
			

- Bộ lọc log phát hiện sự thay đổi cả từ sáng sang tối và từ tối sang sáng, bộ lọc laplace chỉ phát hiện sự thay đổi từ sáng sang tối.

Câu 6: Hình ảnh từ bộ lọc 1 -> 8

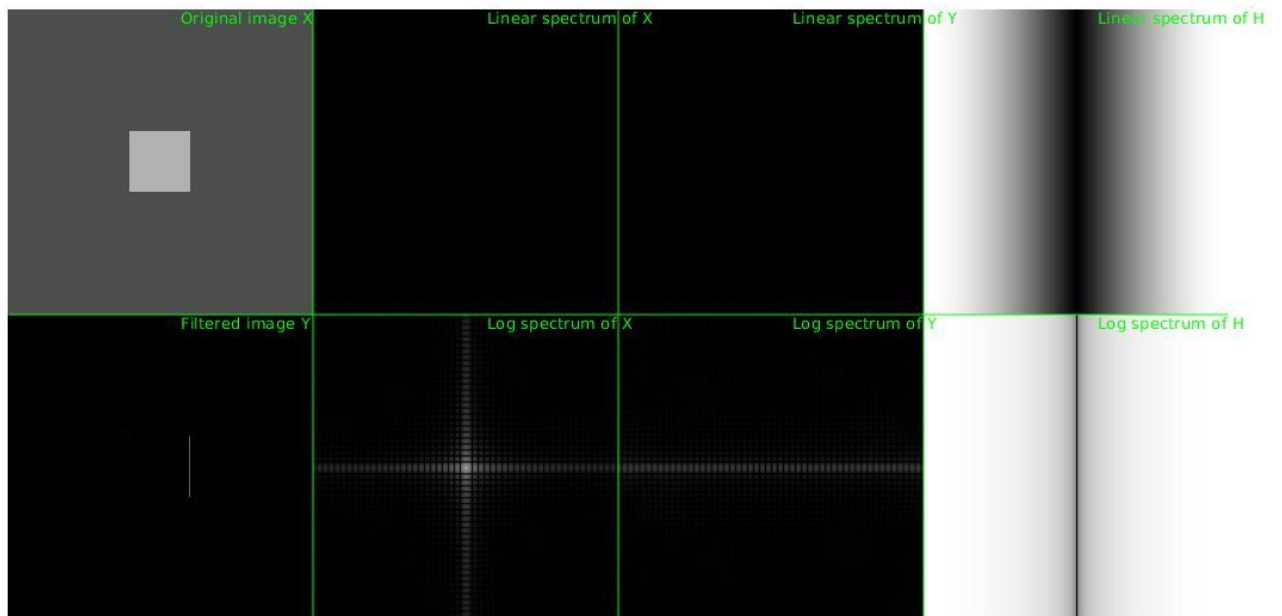


- Nhận xét:
- Với bộ lọc gradient, làm nổi bật các cạnh theo phương ngang (h1), theo phương dọc (h2).
- Với bộ lọc sobel, làm nổi bật các cạnh, tìm biên trong ảnh.
- Với bộ lọc laplace, log cũng phát hiện cạnh.

Câu 7:

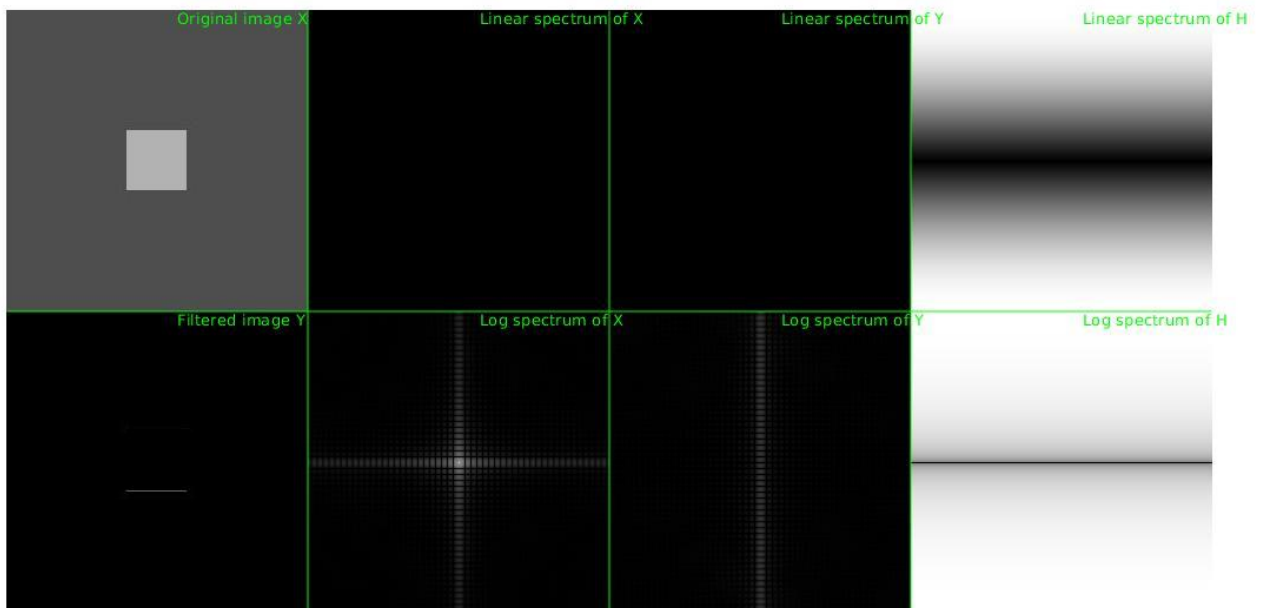
Filter 1:

- Phổ tần số bộ lọc là tần số cao theo phương dọc vì vậy nó làm mất hầu như các tần số theo phương dọc.



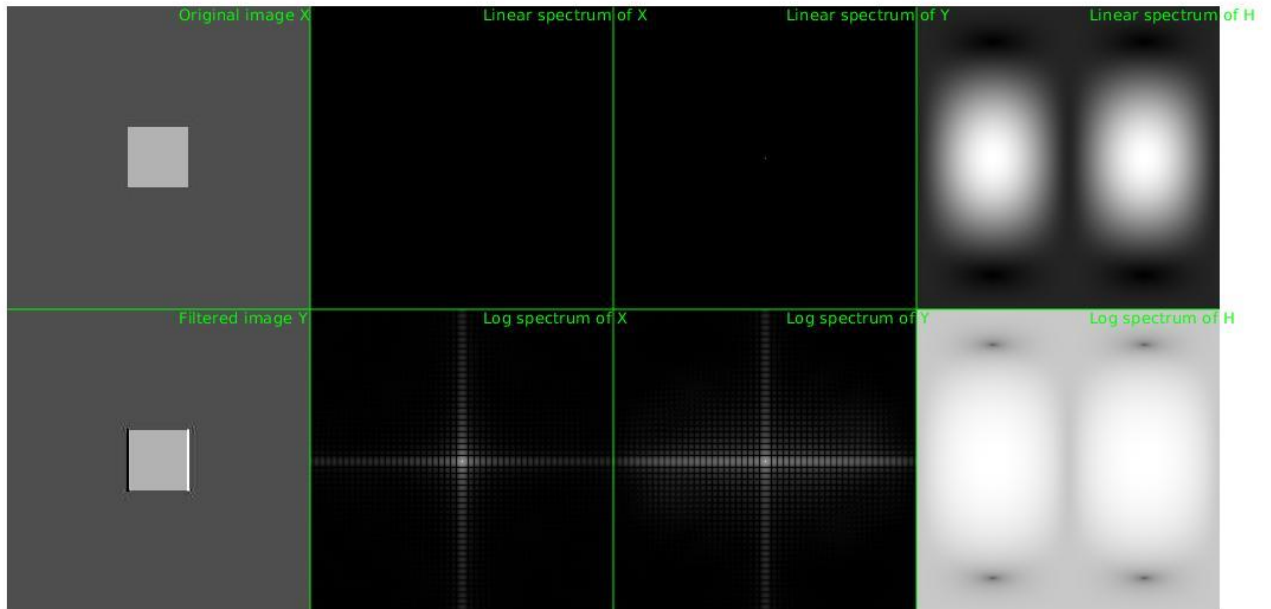
Filter 2:

- Nhận xét: phổ tần số bộ lọc tần số cao nằm ngang vì vậy nó trung hòa tất cả các tần số theo phương ngang.



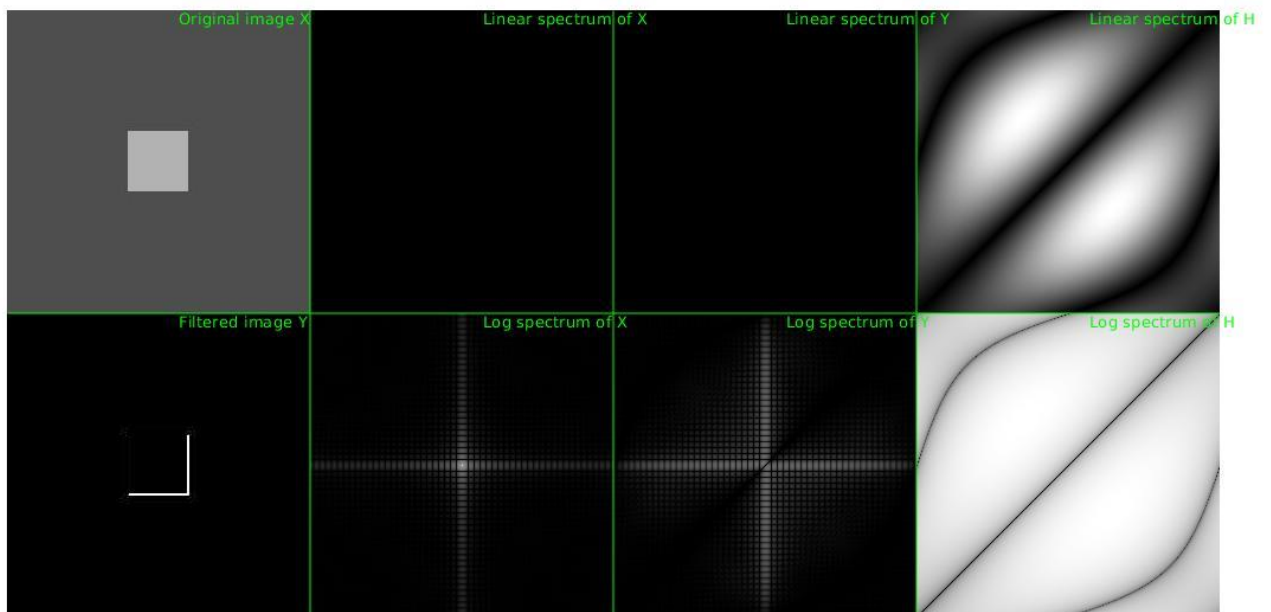
Filter 3:

- Phổ tần số bộ lọc làm sắc nét hơn cạnh (vì nó nhân phần nhiều với một số gần 0 khá cao).



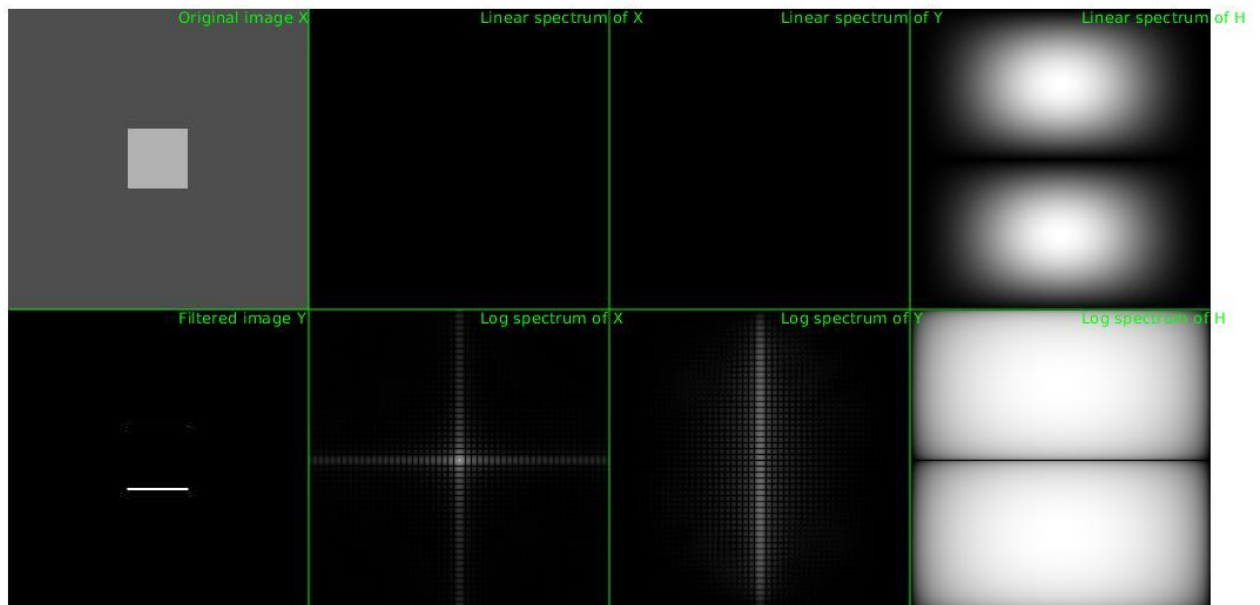
Filter 4:

- Nhận xét: Bộ lọc với các tần số cao màu đen theo dạng hình chiếu làm biến mất những nhiễu theo các phương ngoại trừ phương của tần số cao bộ lọc.



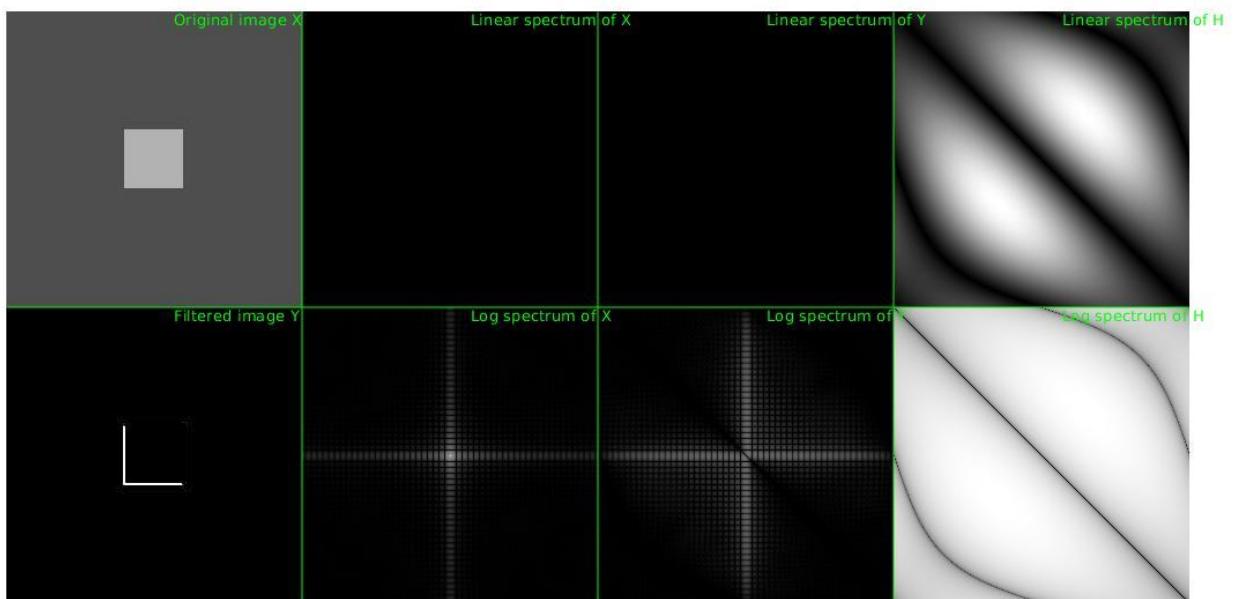
Filter 5:

- Nhận xét: Bộ lọc với tần số cao màu đen làm biến mất các vùng nhiễu theo phương dọc của ảnh.



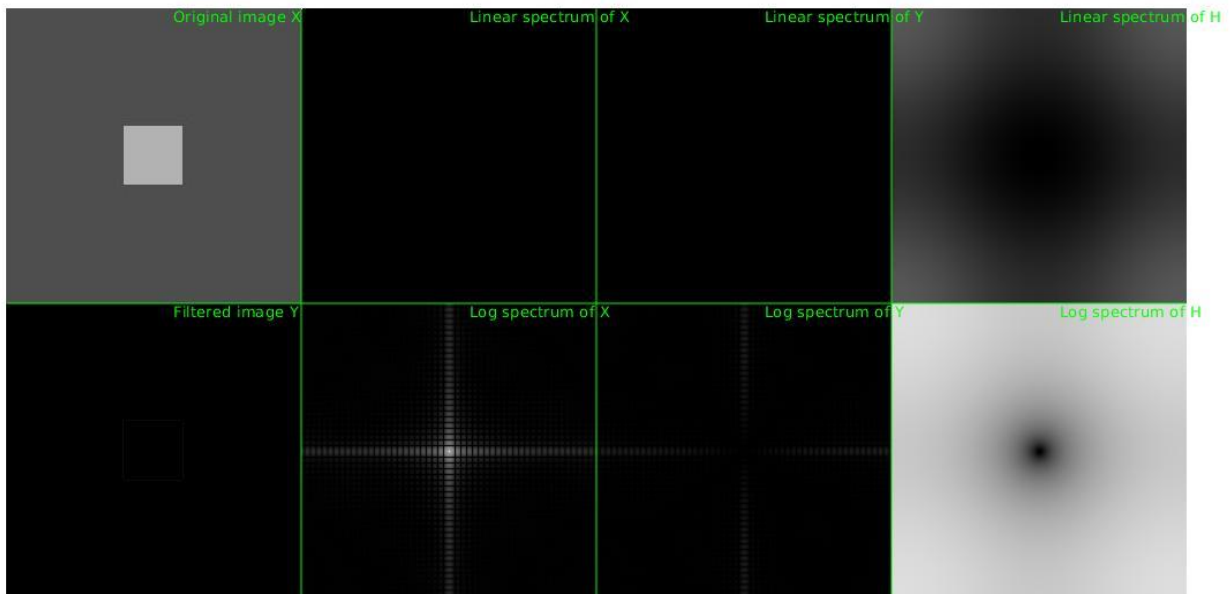
Filter 6:

- Nhận xét: Tần số cao màu đen của bộ lọc rất mạnh cho những đường chéo.



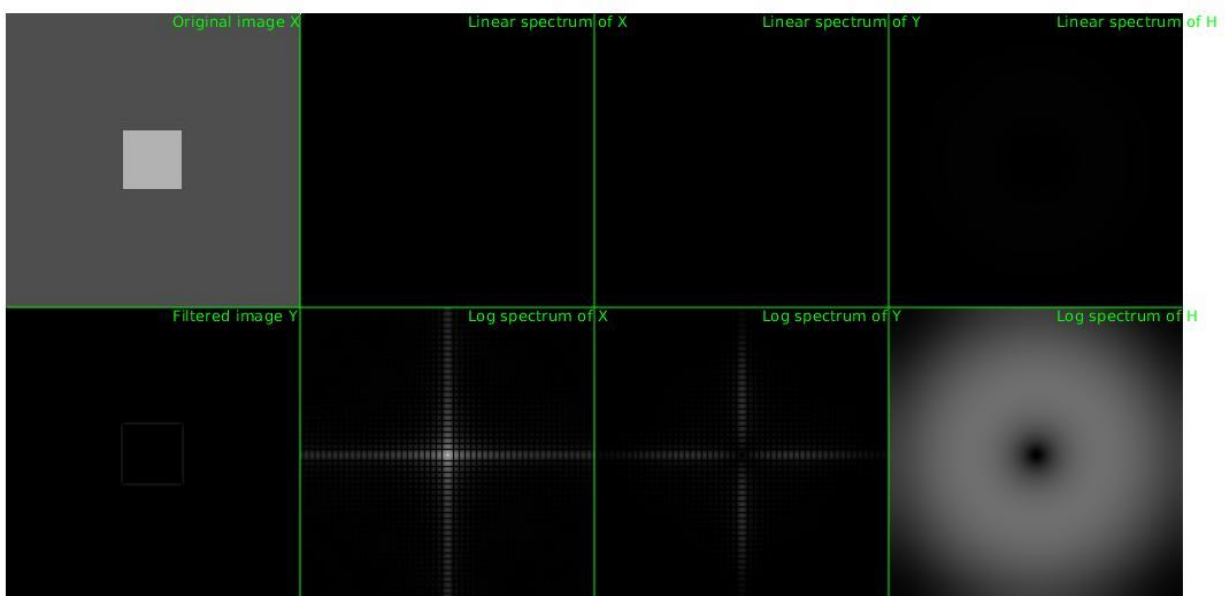
Filter 7:

- Nhận xét: Vì các tần số cao của bộ lọc (màu đen) nên khi nhân với ảnh gốc đã biến đổi, phần đó sẽ bị nhân nhiều hơn dẫn đến giảm bớt nhiễu.



Filter 8:

Nhận xét: phổ tần số bộ lọc các tần số thấp tập trung ở tâm và các góc của hình, nhưng chủ yếu bộ lọc hiệu quả vì các tần số cao ở tập, khi nhân element-wise với ảnh đã biến đổi f thì sẽ giảm các nhiễu sáng ở chính giữa đi.



Phần 3.2

Câu 1:

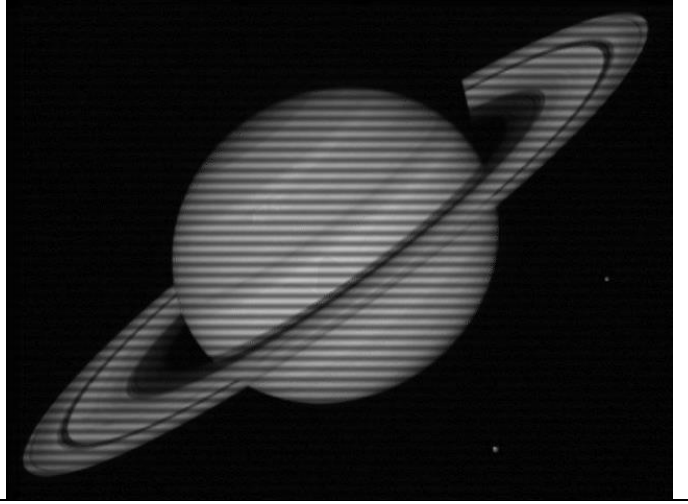
Code
<pre>X = imread('/home/duy/port/Lab02/thuchanh2/saturn.png') X = double(X) X = X/255</pre>

Câu 2:

Code	Hình ảnh
------	----------

```
X =
imread('/home/duy/port/Lab02/thuchanh2/saturn.png')
X = double(X)
X = X/255
```

```
Y = AddPeriodicNoise(X)
imshow(Y)
imwrite(Y,'e2.png')
PSNR = psnr(Y,X)
```



PSNR(Y,X) = 21.3357

Câu 3:

Code	Hình ảnh
<pre>X = imread('/home/duy/port/Lab02/thuchanh2/saturn.png') X = double(X) X = X/255 Y = AddPeriodicNoise(X) FY = fft2(Y) FY = fftshift(FY) imshow(mat2gray(10*log10(abs(FY)+0.00001))) % imwrite(Y,'e2.png')</pre>	

- Nhận xét: điểm nhiễu trên phổ tần số tập trung nhiều theo phương ngang nên khi quan sát phổ tần số sẽ thấy vùng sáng là phương dọc, như vậy đường kẻ dọc ở giữa và 2 chấm tròn ngoài là của nhiễu dọc đã thêm vào.

Câu 4:

Code
<pre>X = imread('/home/duy/port/Lab02/thuchanh2/saturn.png') X = double(X) X = X/255 Y = AddPeriodicNoise(X) FY = fft2(Y) FY = fftshift(FY) [w,h] = size(FY) Z = ones(w,h)</pre>

Câu 5:

Code	Hình ảnh
------	----------

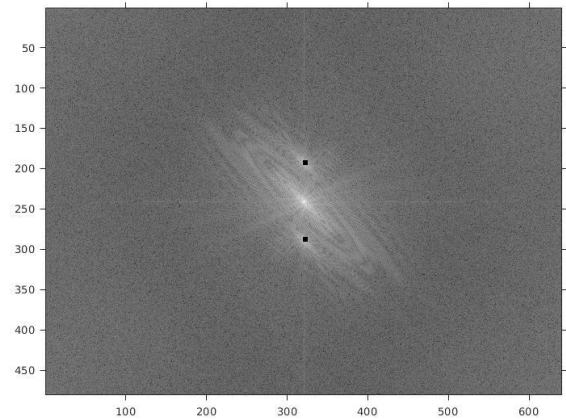
```
X =
imread('/home/duy/port/Lab02/thuchanh2/saturn.png')
X = double(X)
X = X/255
```

```
Y = AddPeriodicNoise(X)
FY = fft2(Y)
FY = fftshift(FY)
[w,h] = size(FY)
```

```
Z = ones(w,h)
```

```
Z(190:195,320:325) = 0
Z(285:290,320:325) = 0
```

```
imshow(Z)
```



Câu 6:

Code

```
X = imread('/home/duy/port/Lab02/thuchanh2/saturn.png')
X = double(X)
X = X/255
```

```
Y = AddPeriodicNoise(X)
FY = fft2(Y)
FY = fftshift(FY)
[w,h] = size(FY)
```

```
% FY_1 = ifftshift(FY)
% FY_1 = ifft2(FY_1)
% imshow(Y)
```

```
Z = ones(w,h)
```

```
Z(190:195,320:325) = 0
Z(285:290,320:325) = 0
```

```
FY2 = FY.*Z
```

Câu 7:

Code

Hình ảnh

```
X =
imread('/home/duy/port/Lab02/thuchanh2/saturn.png')
X = double(X)
X = X/255
```

```
Y = AddPeriodicNoise(X)
FY = fft2(Y)
FY = fftshift(FY)
[w,h] = size(FY)
```

```
% FY_1 = ifftshift(FY)
% FY_1 = ifft2(FY_1)
% imshow(Y)
```

```
Z = ones(w,h)
```

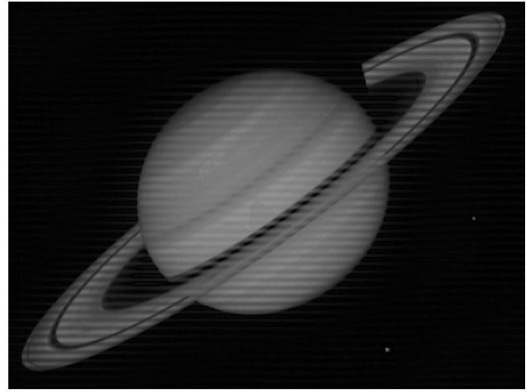
```
Z(190:195,320:325) = 0
Z(285:290,320:325) = 0
```

```
FY2 = FY.*Z
```

```
FY2 = ifftshift(FY2)
FY2 = ifft2(ifftshift(FY2))
```

```
PSNR = psnr(abs(FY2),X)
```

```
imshow(abs(FY2))
```



- PSNR(FY2,X) = 21.1823
- Nhận xét: Tần số thay đổi chủ yếu nằm theo phương dọc, nên khi che đi những vùng tần số mạnh thì sẽ giảm nhiều.