

# Image Sharpening

609410162 彭郁翔

Data due:11/16

Data handed in:11/16

Give two gray-level images, sharpen the two images using the Laplacian operator, unsharp masking, and high-boost filtering in the spatial and frequency domains. 這裡我分成 6 個 .mat 檔分別用 3 種方法和在 spatial domain 和 frequency domain 來執行。



有加上 fre 的是在 frequency domain 下的方法。

## Technical description

In spatial domain:

Laplacian operator :

根據定義及以下矩陣表示：

$$\nabla^2 f = \frac{\delta^2 f}{\delta x^2} + \frac{\delta^2 f}{\delta y^2},$$

$$\frac{\delta^2 f}{\delta x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

$$\frac{\delta^2 f}{\delta y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

$$\Rightarrow \nabla^2 f = \frac{\delta^2 f}{\delta x^2} + \frac{\delta^2 f}{\delta y^2} = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)$$

Filter mask :

0	1	0
1	-4	1
0	1	0

1	1	1
1	-8	1
1	1	1

這裡我有先把原本 image 做 zero-padding，再設定  $\text{mask} = [1 \ 1 \ 1; 1 \ -8 \ 1; 1 \ 1 \ 1]$ ，之後來和原本的 image 矩陣做 convolution，再經過跟原本 image 做相減， $g(x, y) = f(x, y) - \nabla^2 f(x, y)$ ， $g(x, y)$ :sharped image。

可得到一張 sharped image。

```

17 -   img1 = padarray(img1,[1,1]);
18 -   img1 = double(img1);
19 -   img2 = padarray(img2,[1,1]);
20 -   img2 = double(img2);
21
22   %設定Masks
23 -   mask = [1 1 1; 1 -8 1; 1 1 1];
24
25   %利用Laplacian equation
26 -   for i=1:size(img1,1)-2
27 -       for j=1:size(img1,2)-2
28 -           img1_t(i,j) = sum(sum(mask.* img1(i:i+2,j:j+2)));
29 -       end
30 -   end
31
32 -   for i=1:size(img2,1)-2
33 -       for j=1:size(img2,2)-2
34 -           img2_t(i,j,1) = sum(sum(mask.* img2(i:i+2,j:j+2)));
35 -           img2_t(i,j,2) = sum(sum(mask.* img2(i:i+2,j:j+2)));
36 -           img2_t(i,j,3) = sum(sum(mask.* img2(i:i+2,j:j+2)));
37 -       end
38 -   end
39

```

Unsharp masking :

Sharpening images can be implemented by subtracting a blurred version of an image from the image itself.

$$f_s(x, y) = f(x, y) - \bar{f}(x, y)$$

$f_s(x, y)$  : sharpened image

$\bar{f}(x, y)$  is a blurred version of  $f(x, y)$ .

1. Subtract the blurred version from the original (called the mask).
2. Add the mask to the original.

這裡我使用 **高斯模糊** 來模糊原圖，再用原圖減去模糊後的圖片，最後再加上原圖得到 sharpened image。

```

8      %建立gaussian filter
9      Gauss_filter = fspecial('gaussian',[3 3],1);

25     %convolution
26     for i=1:size(img1,1)-2
27         for j=1:size(img1,2)-2
28             img1_t(i,j) = sum(sum(Gauss_filter.* img1(i:i+2,j:j+2)));
29         end
30     end
31
32     sharpl = I1 - uint8(img1_t);
33     sharpl = I1 + sharpl;
34
35     for i=1:size(img2,1)-2
36         for j=1:size(img2,2)-2
37             img2_t(i,j,1) = sum(sum(Gauss_filter.* img2(i:i+2,j:j+2)));
38             img2_t(i,j,2) = sum(sum(Gauss_filter.* img2(i:i+2,j:j+2)));
39             img2_t(i,j,3) = sum(sum(Gauss_filter.* img2(i:i+2,j:j+2)));
40         end
41     end
42
43     sharp2 = I2 - uint8(img2_t);
44     sharp2 = I2 + sharp2;

```

### High-boost filtering :

A high-boost filtered image,  $f_{hb}$ , is defined as:

$$f_{hb}(x, y) = Af(x, y) - \bar{f}(x, y),$$

where  $A \geq 1$  and  $\bar{f}$  is a blurred version of  $f$

$$f_{hb}(x, y) = (A - 1)f(x, y) + f(x, y) - \bar{f}(x, y).$$

$$f_{hb}(x, y) = (A - 1)f(x, y) + f_s(x, y)$$

這裡我把 A 代 4:

```

25 %convolution
26 for i=1:size(img1,1)-2
27     for j=1:size(img1,2)-2
28         img1_t(i,j) = sum(sum(Gauss_filter.* img1(i:i+2,j:j+2)));
29     end
30 end
31
32 sharpmask1 = I1 - uint8(img1_t);
33 sharpI1 = 3*I1 + sharpmask1;
34
35 for i=1:size(img2,1)-2
36     for j=1:size(img2,2)-2
37         img2_t(i,j,1) = sum(sum(Gauss_filter.* img2(i:i+2,j:j+2)));
38         img2_t(i,j,2) = sum(sum(Gauss_filter.* img2(i:i+2,j:j+2)));
39         img2_t(i,j,3) = sum(sum(Gauss_filter.* img2(i:i+2,j:j+2)));
40     end
41 end
42
43 sharpmask2 = I2 - uint8(img2_t);
44 sharp2 = 3*I2 + sharpmask2;
45

```

## In frequency domain:

Frequency domain filtering operation :

Input image → pre-processing → Fourier transform → Filter function →

Inverse Fourier transform → post-processing → enhanced image

### Laplacian operator :

這裡我使用一個網路上參考的 paddedsize.m 檔，呼叫 paddedsize function。

```

8 PQ1=paddedsize(size(img1),1);
9 PQ2=paddedsize(size(img2),1);
10

```

根據講義上去把原圖和 filter 做 fourier transform，再經過 convolution，最後再做 inverse fourier transform。

$$H(u, v) = [1 + 4\pi^2[(u - M/2)^2 + (v - N/2)^2]].$$

$$g(x, y) = \mathfrak{F}^{-1}\{[1 + 4\pi^2((u - M/2)^2 + (v - N/2)^2)]F(u, v)\}$$

```

20 %進行fourier transform
21 - fft_img1 = fft2(double(img1),PQ1(1),PQ1(2));
22 - fft_img2 = fft2(double(img2),PQ2(1),PQ2(2));
23
24 %設定filter
25 - H = fspecial('laplacian');
26
27 %把filter也做傅立葉轉換
28 - H1=fft2(double(H),PQ1(1),PQ1(2));
29 - H2=fft2(double(H),PQ2(1),PQ2(2));
30
31 %convolution
32 - img1_t = fft_img1.*(H1+1);
33 - img2_t = fft_img2.*(H2+1);
34
35 %inverse fourier transform
36 - g1 = real(ifft2(img1_t));
37 - g2 = real(ifft2(img2_t));
38
39 - sharp1 = uint8(g1);
40 - sharp2 = uint8(g2);

```

### Unsharp masking :

這裡我一樣使用 **高斯模糊** 來模糊原圖，再將原圖及 gaussian filter 做 fourier transform，做 convolution 後再 inverse fourier transform 得到 g，再用原圖減去 g，最後再加上原圖得到 sharpened image。

```

24 - fft_img1 = fft2(double(img1),PQ1(1),PQ1(2));
25 - fft_img2 = fft2(double(img2),PQ2(1),PQ2(2));
26
27
28 %把filter也做傅立葉轉換
29 - H1=fft2(double(H),PQ1(1),PQ1(2));
30 - H2=fft2(double(H),PQ2(1),PQ2(2));
31
32 %convolution
33 - img1_t = fft_img1.*(H1+1);
34 - img2_t = fft_img2.*(H2+1);
35
36 %inverse fourier transform
37 - g1 = real(ifft2(img1_t));
38 - g2 = real(ifft2(img2_t));
39
40 - sharpmask1 = I1 - uint8(g1);
41 - sharp1 = I1 + sharpmask1;
42
43 - sharpmask2 = I2 - uint8(g2);
44 - sharp2 = I2 + sharpmask2;

```

### High-boost filtering :

這裡我把 A 代 2.7(投影片上寫的), b 代 3:

$$H_{hb}(u, v) = (A - 1) + H_{hn}(u, v),$$

with  $A \geq 1$ .

High-frequency emphasis has a filter transfer function given by:  
(4.4-20)

where  $a \geq 1$  and  $b > a$ .

```
23      %進行fourier transform
24 -    fft_img1 = fft2(double(img1),PQ1(1),PQ1(2));
25 -    fft_img2 = fft2(double(img2),PQ2(1),PQ2(2));
26
27
28      %把filter也做傅立葉轉換
29 -    H1=fft2(double(H),PQ1(1),PQ1(2));
30 -    H2=fft2(double(H),PQ2(1),PQ2(2));
31
32      %convolution
33 -    img1_t = fft_img1.*(H1+1);
34 -    img2_t = fft_img2.*(H2+1);
35
36      %inverse fourier transform
37 -    g1 = real(ifft2(img1_t));
38 -    g2 = real(ifft2(img2_t));
39
40 -    sharpmask1 = I1 - 3*uint8(g1);
41 -    sharpl = 2.7*I1 + sharpmask1;
42
43 -    sharpmask2 = I2 - 3*uint8(g2);
44 -    sharp2 = 2.7*I2 + sharpmask2;
```

## Experimental results

In spatial domain:

### Laplacian operator :

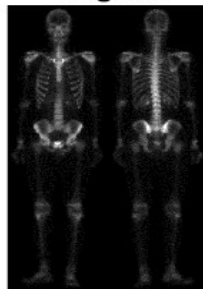
**Original**



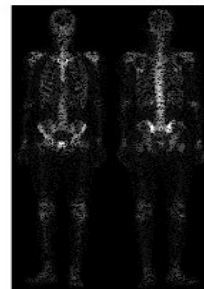
**transformed**



**Original**



**transformed**



Unsharp masking :

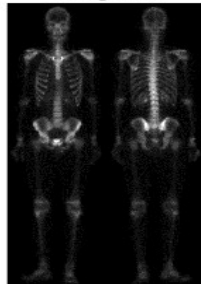
**Original**



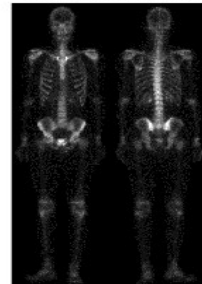
**transformed**



**Original**



**transformed**



High-boost filtering :

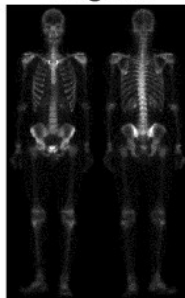
Original



transformed



Original



transformed



In frequency domain:

Laplacian operator :



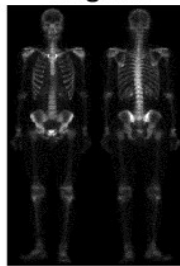
**Original**



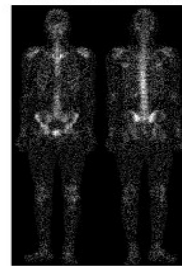
**transformed**



**Original**



**transformed**



Unsharp masking :

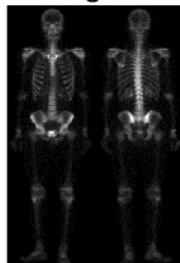
**Original**



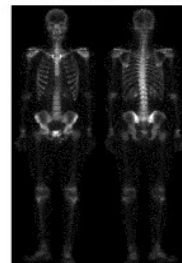
**transformed**



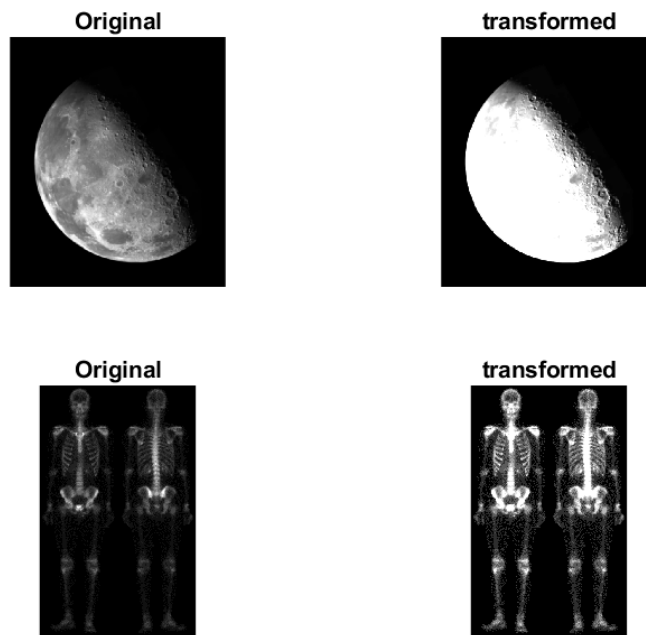
**Original**



**transformed**



High-boost filtering :



## Discussions

Frequency domain 的流程及作法不太懂，做出來不知道是對還錯。  
希望助教能提供範例程式給大家參考。

## References and Appendix

- Chapter 3 IMAGE ENHANCEMENT IN THE SPATIAL DOMAIN
- Chapter 4 IMAGE ENHANCEMENT IN THE FREQUENCY DOMAIN