

SIMG-782 Digital Image Processing

Homework 6

Ex. 1 — (Circular Convolution)

Let $\mathbf{f} = [1, 3, -1, 2, 0, -3]$ and $\mathbf{h} = [-1, 3, -2]$.

- Calculate the convolution $\mathbf{f} * \mathbf{h}$ assuming that both \mathbf{f} and \mathbf{h} are zero-padded to a length of 12.
- Calculate the convolution using a product of discrete Fourier transforms.
- Plot $F_p(u)$, $G_p(u)$ and $H_p(u)$ as points in the complex plane for $0 \leq u \leq 11$.
- Calculate and interpret $\sum g_p^2(n)$ and $\sum |G_p(u)|^2$.

Answer (ex. 1) — (Circular Convolution)

- The question is stated ambiguously. Let's calculate the *circular* convolution of the two padded sequences. To do that, write one of them in a ring and write the other in reverse order around the same ring, with the first elements in the same position (call this position 0). Sum the product of the elements around the ring. Move one of the rings by one step. Form the sum of all of the products. Continue until you have done this 12 times. The result is $g_p = [-1, 0, 8, -11, 8, -1, -9, 6, 0, 0, 0, 0]$. The result can be done in IDL by the following statements:

```
fp=[1,3,-1,2,0,-3,0,0,0,0,0,0]
hp=[-1,3,-2,0,0,0,0,0,0,0,0,0]
gp=convol(fp,hp,/edge_zero,center=0)
```

- The transform uses the padded sequences

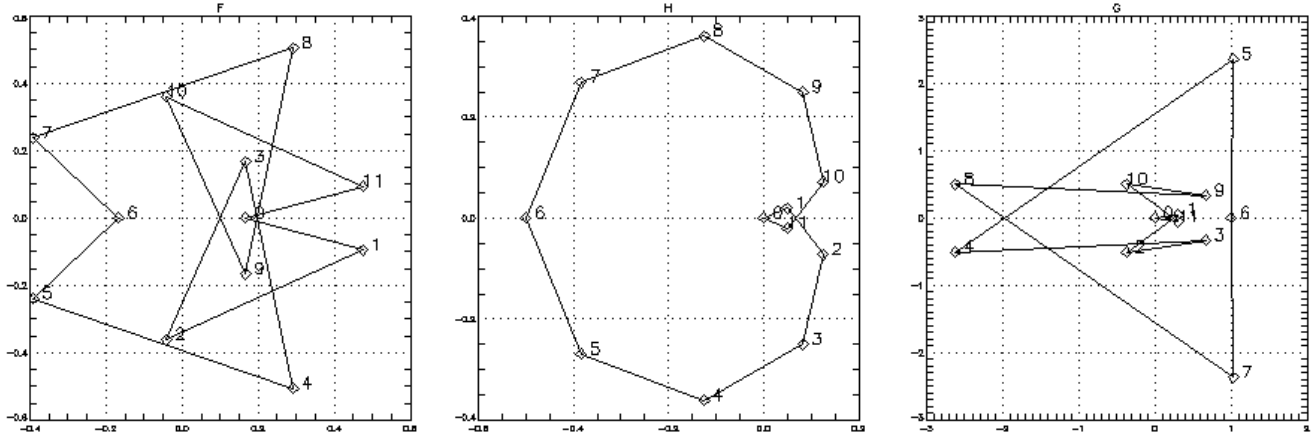
```
FFP=FFT(fp)
FHP=FFT(hp)
FGP=12*FFP*FHP
gp=REAL_PART(FFT(FGP,/INVERSE))
print,clip(gp)
```

```
gp = [ -1.0000  0.0000  8.0000 -11.0000  8.0000 -1.0000 -9.0000  6.0000  0.0000  0.0000  0.0000  0.0000 ]
```

Note 1: See below for a discussion of the N factor in the DFT and convolution.

Note 2: The inverse FFT (as well as the FFT) produces a complex result. The result is the real part. The "clip()" function simply removes extremely small values in the printout.

- Plots of the real and imaginary points in the complex plane are shown below. Note that $G_p(n) = 12F_p(n)H_p(n)$ for $n = 0, 1, 2, \dots, 11$.



(d) The DFT transform pair is

$$F(u) = \frac{1}{N} \sum_{n=0}^{N-1} f(n)e^{-i2\pi nu/N} \quad f(n) = \sum_{u=0}^{N-1} F(u)e^{-i2\pi nu/N}$$

Then

$$\begin{aligned} \sum_{n=0}^{N-1} |f(n)|^2 &= \sum_{n=0}^{N-1} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u)F^*(v)e^{i2\pi(u-v)n/N} \\ &= \sum_{u=0}^{N-1} F(u) \sum_{v=0}^{N-1} F^*(v) \sum_{n=0}^{N-1} e^{i2\pi(u-v)n/N} \\ &= N \sum_{u=0}^{N-1} |F(u)|^2 \end{aligned}$$

where we have used

$$\sum_{n=0}^{N-1} e^{i2\pi(u-v)n/N} = \begin{cases} 0 & u \neq v \\ N & u = v \end{cases}$$

The factor of N on the right side is brought about by the definition of the DFT. The direct and inverse transforms are not symmetrical in scaling. (Some definitions put $\frac{1}{\sqrt{N}}$ on both sides.) This also shows up in “frequency-domain convolution”.

$$\begin{aligned} g(n) &= \sum_{m=0}^{N-1} f(m)h(n-m) = \sum_{m=0}^{N-1} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u)H(v)e^{i2\pi um/N} e^{i2\pi v(n-m)/N} \\ &= \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u)H(v)e^{i2\pi vn/N} \sum_{m=0}^{N-1} e^{i2\pi(u-v)m/N} \\ &= \sum_{u=0}^{N-1} NF(u)H(u)e^{i2\pi vn/N} = \sum_{u=0}^{N-1} G(u)e^{i2\pi vn/N} \end{aligned}$$

Hence,

$$G(u) = NF(u)H(u)$$

This can be used to calculate

```
sum1=total(gp^2)
print,'sum of g^2=',sum1
sum2=total(abs(FGP)^2)*12
```

```
print,'sum of 12*|G|^2=',sum2

sum of g^2=      368.000
sum of 12*|G|^2=  368.000
```

Ex. 2 — (2D Filtering Exercise)

An image called 'urban_image_3.png' is contained in the images directory at the course web site. Use this image as the “subject” for this problem. This image has dimensions 256×256 .

- (a) We have seen in the previous homework that a one-dimensional gaussian filter can be represented by $K \exp(-\pi(u/a)^2)$ where u is a suitable frequency variable, a is a parameter that controls the bandwidth and K is a scale factor. Now let u be an array of size 256×256 whose rows are all equal to the sequence $[-128, -127, -126, \dots, 126, 127]$. Construct the function

$$H_1(u) = e^{-\pi(u/10)^2}$$

Use frequency-domain filtering to find the output image produced by filtering the input image through this filter. You may have to scale the output image for display purposes.

- (b) Repeat the above with the filter

$$H_2(u) = e^{-\pi(u/50)^2}$$

Compare the results.

- (c) In the previous homework we found that $\exp(-\pi u^2)$ and $\exp(-\pi x^2)$ form a Fourier transform pair. Use this fact to determine the impulse response of $H_1(u)$. Can you get the same result by taking the inverse FFT of $H_1(u)$? Note that the impulse response $h_1(x, y)$ is now a function of two variables.
- (d) Produce the result of part (a) by doing convolution.
- (e) Repeat for $H_2(u)$.
- (f) Let $v = u^T$, so that its columns are all the sequence $[-128, -127, -126, \dots, 126, 127]$. Construct the function

$$H_3(v) = e^{-\pi(v/10)^2}$$

Use frequency-domain filtering to find the output image produced by filtering the input image through this filter. Compare the results with those of (a).

- (g) Repeat the above with the filter

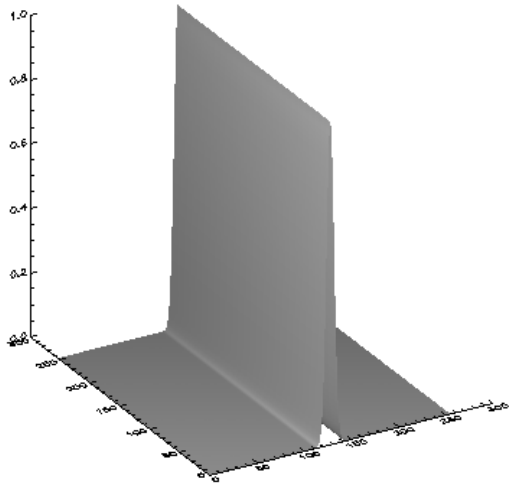
$$H_4(u) = e^{-\pi(v/50)^2}$$

Compare the results.

- (h) Construct the impulse response functions $h_3(x, y)$ and $h_4(x, y)$. Using them, repeat the results found from filtering in the frequency domain.
- (i) A filter of the form $H(u, v) = \exp(-[(u/a)^2 + (v/b)^2])$ can be factored into something of the form $H(u, v) = S(u)T(v)$. How can this fact be exploited in building a faster filter algorithm?

Answer (ex. 2) — (2D Filtering Exercise)

- (a) The filter response and the resulting image are shown below. The filter is selective in the u -direction but broadband in the v -direction. Thus, we expect only low frequencies to pass in the horizontal direction but all frequencies to pass in the vertical direction.

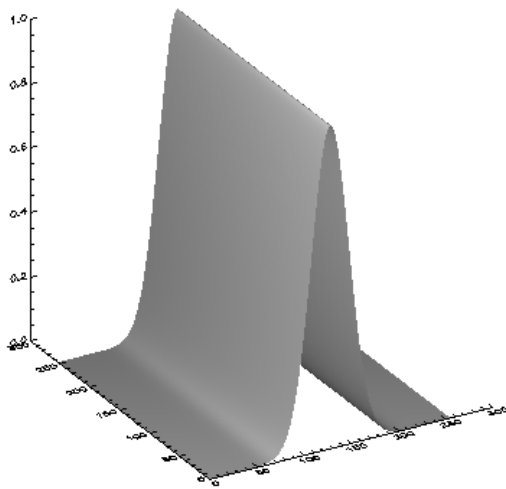


(a) $W_u = 10$

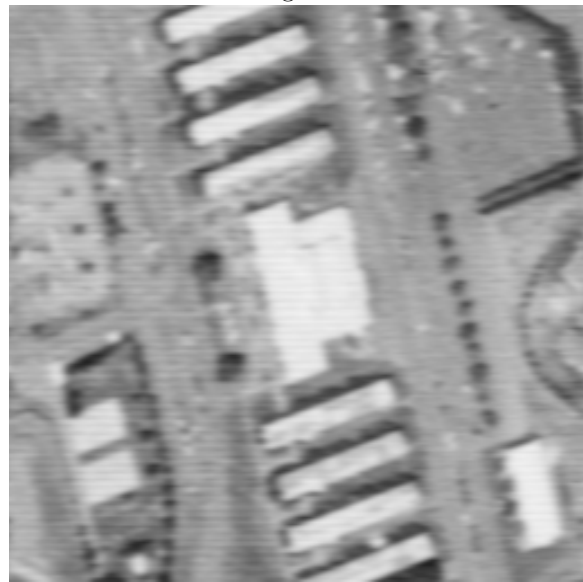


(b) Filtered with $W_u = 10$

(b) The effect of broadening the filter is shown below. There is now much less smearing in the horizontal direction.



(c) $W_u = 50$



(d) Filtered with $W_u = 50$

(c) In general, if $f(x) \leftrightarrow F(u)$ then $f(ax) \leftrightarrow \frac{1}{a}F(\frac{u}{a})$. Thus,

$$e^{-\pi(u/10)^2} \leftrightarrow 10e^{-\pi(10x)^2}$$

In the discrete representation we need to take the periodicity in the two domains into account. Since the image has dimensions $N \times N$ where $N = 256$, the Fourier transforms shown in (a) and (c) above repeat with period $N\Delta u = N\Delta v = 256$ in both the u and v directions, where, implicitly, $\Delta u = \Delta v = 1$. Therefore, the spatial grid has $\Delta x = \Delta y = \frac{1}{256}$ and spatial period $N\Delta x = N\Delta y = 1$. The impulse response of the filter at

$(x, y) = (n\Delta x, m\Delta y)$ is

$$\begin{aligned} h(n, m) &= \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u, v) e^{i2\pi un/N} e^{i2\pi vm/N} \\ &= \left(\sum_{v=0}^{N-1} e^{i2\pi vm/N} \right) \left(\sum_{u=0}^{N-1} e^{-\pi((u-N/2)/a)^2} e^{i2\pi un/N} \right) \\ &= 256\delta(m) \left(\frac{1}{\Delta u} \sum_{u=0}^{N-1} e^{-\pi((u-N/2)/a)^2} e^{i2\pi un/N} \Delta u \right) \end{aligned}$$

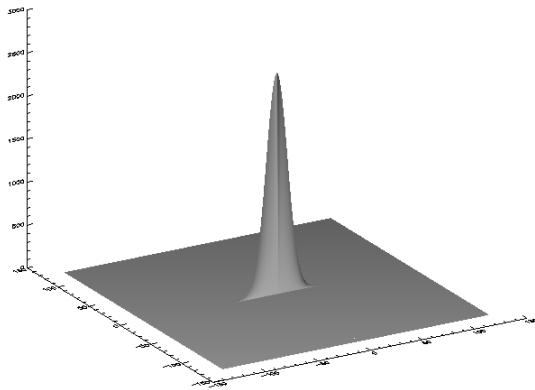
The last term is closely approximated by

$$\int_{-\infty}^{\infty} e^{-\pi(u/a)^2} e^{i2\pi un/N} du = ae^{\pi(an/N)^2}$$

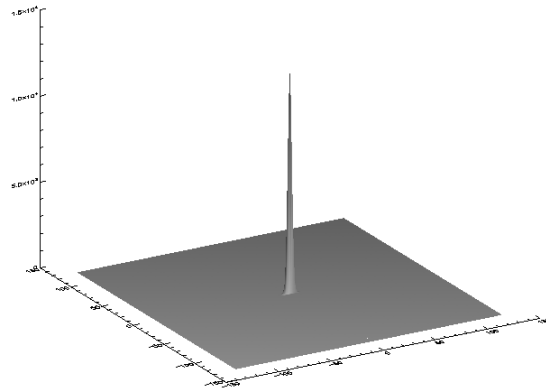
Therefore,

$$h(n, m) = 256a\delta(m)e^{-\pi(an/N)^2}$$

with periodic repetition of period $(N, N) = (256, 256)$. A plot of one period of h is shown below.



(e) $a = 10$



(f) $a = 50$

- (d) Since the filter is a delta function in the y direction, we can do 1-D filtering on each row of the image. To maintain the same general brightness we normalize the impulse response to have unit “area.”

```
x=(findgen(256)-128)/256
ha=exp(-!pi*(10*x)^2)
ha=ha/total(ha)
B=fltarr(256,256)
for k=0,255 do B[* ,k]=convol(float(A[* ,k]),ha,/edge_zero)
disp_image,B
```

The results are shown below. Except for some pixels on the edges of the image the results are the same as obtained in the frequency domain. The edge effects are probably due to the implementation of the computations.



(g) $a = 10$ by convolution



(h) $a = 50$ by convolution

(e) See figure (h) above.

(f) Filtering in the vertical direction produces the images shown below. The difference is especially apparent by comparing (i) with (b).



(i) $a = 10$ in v direction



(j) $a = 50$ in v direction

(g) See (j) above.

(h) Transpose the filter analysis of parts (c), (d), (e).

(i) This enables $h(x, y) = s(x)t(y)$. The separable filter can be used to do filtering with $s(x)$ on the rows and then $t(y)$ on the columns. Separable filters are widely used in image processing because of the decrease by order N in the number of computations required.

Ex. 3 — (Blurred Image Restoration)

Images `urban0.png` and `urbanblur.png` are in the images directory at the course web site. Use Wiener filtering to attempt to recover the blurred image.



urban0.png



urbanblur.png

Answer (ex. 3) — We want to construct a filter of the form

$$T(u, v) = \frac{H^*(u, v)}{|H(u, v)|^2 + K}$$

This requires knowledge of the sensor frequency response $H(u, v)$ and an estimate of K . After some experimentation the restoration below was done with a symmetric Gaussian filter with bandwidth $w = 50$ and $K = 0.0001$

```
A=read_image('urbanblur.png')
```

```
disp_image,A,xp=0,yp=0
```

```
t=findgen(256)-128
```

```
u=t#replicate(1,256)
```

```
v=transpose(u)
```

```
w=50
```

```
H=exp(-((u/w)^2+(v/w)^2))
```

```
window,/fr & shade_surf,H,u,v
```

```
H=shift(H,-128,-128)
```

```
K=0.0001
```

```
T=H/(H^2+K)
```

```
AF=FFT(A)
```

```
BF=T*AF
```

```
B=FFT(BF,/inverse)
```

```
disp_image,real_part(B),xp=260,yp=0
```



urban0.png



urbanblur.png



Restored