Image Processing - Assignment 3

Kelvin Likollari

1. Theoretical Exercise 1

We're given the fact that f(x, y) = g(x) · g(y). By the definition of the 2D Fourier Transform we have:

$$\mathcal{F}(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-i2\pi(ux+vy)}dxdy,$$

Given that f(x, y) = g(x) · g(y):

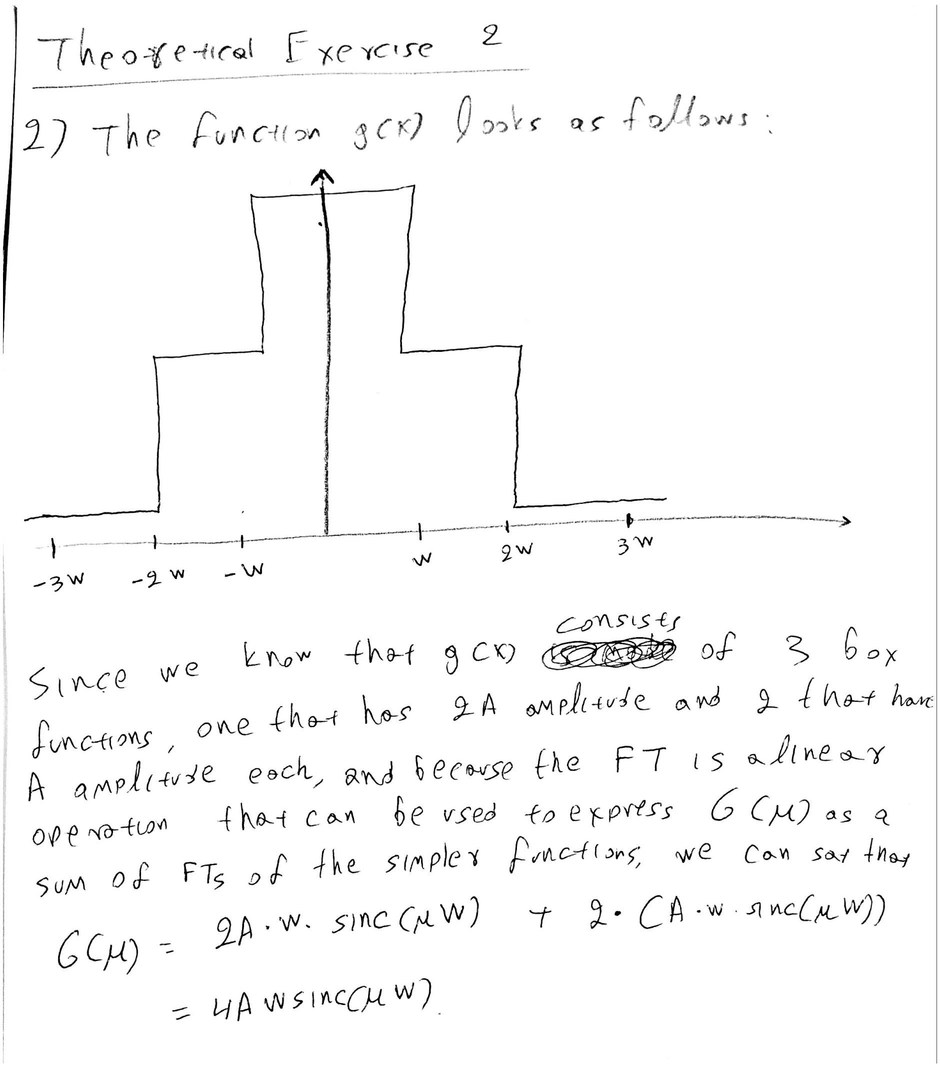
$$\mathcal{F}(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x)\cdot g(y)e^{-i2\pi(ux+vy)}dxdy,$$

Taking the integral over x first, we get:

$$\mathcal{F}(u,v) = \int_{-\infty}^{\infty} g(x)e^{-i2\pi(ux)}dx \cdot \int_{-\infty}^{\infty} g(y)e^{-i2\pi(vy)}dy,$$

We can notice that the above equation depicts the 1D FT of g(x) and g(y) with respect to u and v. Hence we can write the original F(u, v) as Fourier Transformations of g(x) and g(y), that is, F(u, v) = G(u) · G(v).

2. Theoretical Exercise 2



Theoretical Exercise 2

2) The function g(x) looks as follows:

$-3W \quad -2W \quad -W \quad W \quad 2W \quad 3W$

Since we know that g(x) consists of 3 box functions, one that has 2A amplitude and 2 that have A amplitude each, and because the FT is a linear operation that can be used to express $G(\mu)$ as a sum of FTs of the simpler functions, we can say that

$$G(\mu) = 2A \cdot w \cdot \mathrm{sinc}(\mu W) + 2 \cdot (A \cdot w \cdot \mathrm{sinc}(\mu W))$$

$$= 4AW\,\mathrm{sinc}(\mu W)$$

Figure 1: Second Exercise

3. Theoretical Exercise 3

3) Reduce image size (factor of c) $\longrightarrow$ aliasing

$\downarrow$

Mitigated using a low-pass filter bef.

$$\sigma_f = 1/2 \cdot \sigma_s \cdot \pi$$

, 75% attenuation

1D - signal $s(t)$

$$FT(s(t)) = S(w)$$

As appling a Gaussian filter with σ_s in spatial domain is equivalent to point-wise multiplication of $S(w)$ with a Gaussian function in the freq. domain

$$\sigma_f = 1/(2 \cdot \sigma_s \cdot \pi)$$

$$Nyquist\_freq = \frac{sampling\_freq}{2}$$

$$Nyquist\_freq = c \cdot \pi$$

To find desired cut-off frewency

$$G(cutoff) = e^{\left(-\frac{1}{2} \cdot cutoff / \sigma_f\right)^2} \Leftrightarrow$$

$$cutoff = \sigma_f \cdot \sqrt{-2 \cdot \log(.75)}$$

$$cutoff = \frac{1}{2\sigma_s\pi} \cdot \sqrt{-2\log(0.75)}$$

But cutoff $<$ Nyquist_freq $\Leftrightarrow$

$$\sigma_s = \frac{1}{2 \cdot c \cdot \sqrt{-2\log(0.75)}}$$

But because $\sigma_s < 0$, just take its abs value or multiply by -1.

Figure 2: Third Exercise

Bonus: Show that the Fourier transform of a 1D Gaussian filter is also a Gaussian

3: Bonus: Prove that the FT of a 1D-Gaussian filter is also a Gaussian function.

By definition, the Gaussian function in the spatial domain,

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{\left(\frac{-x^2}{2\sigma^2}\right)}, \text{ with } \sigma = \text{standard deviation}$$

$$G(f) = \int \left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(\frac{-x^2}{2\sigma^2}\right)} \cdot (-2\pi i f x)\right) dx$$

$$G(f) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \int \left(e^{\left(\frac{-(x - i\sigma^2 f)^2}{2\sigma^2}\right)}\right) dx$$

$$= \frac{-(x - i\sigma^2 f)^2}{2\sigma^2} = \frac{-x^2 + 2i\sigma^2 f x - i^2\sigma^4 f^2}{2\sigma^2}$$

$$= \frac{-x^2 + 2i\sigma^2 f x + \sigma^4 f^2}{2\sigma^2}$$

$$G(f) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(\frac{\sigma^4 f^2}{2\sigma^2}\right)} \cdot \sqrt{2\pi\sigma^2} = e^{\frac{\sigma^4 f^2}{2\sigma^2}}$$

thus the FT of a 1D-Gaussian filter is also a Gaussian function

Figure 3: Third Exercise

4. Theoretical Exercise 4

4)

$$F(w) = A \cdot \left(\mathrm{sinc}\left(w \cdot \frac{W}{2}\right)\right)^2$$

According to the convolution theorem:

$$F(w) = A \left(\mathrm{sinc}\left((w - w_0) \cdot \frac{W}{2}\right)\right) \cdot \mathrm{sinc}\left((w + w_0) \cdot \frac{W}{2}\right)$$

Applying the IFT

$$f(x) = \frac{1}{2\pi} \cdot \int_{-\infty}^{+\infty} F(w) \cdot e^{iwx}\, dx$$

$$F1(w) = \mathrm{IFT}\left(\mathrm{sinc}\left((w - w_0) \cdot \frac{W}{2}\right)\right)$$

$$F2(w) = \mathrm{IFT}\left(\mathrm{sinc}\left((w + w_0) \cdot W/2\right)\right)$$

Replacing:

$$F1(w) = \frac{1}{2\pi} \cdot \int_{-\infty}^{+\infty} \mathrm{sinc}\left((w - w_0) \cdot \frac{W}{2}\right) \cdot e^{iwx}\, dw$$

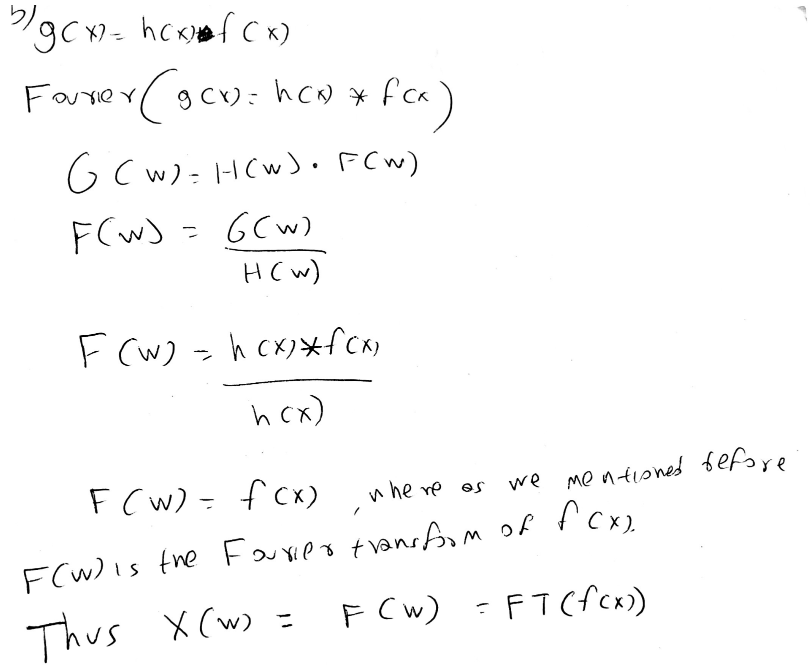
and

$$F2(w) = 1/2\pi \cdot \int_{-\infty}^{+\infty} \mathrm{sinc}\left((w + w_0) \cdot \frac{W}{2}\right) \cdot e^{iwx}$$

$$f(x) = F1(w) \underset{\uparrow\,\text{convolution}}{*} F2(w)$$

Figure 4: Fourth Exercise

5. Theoretical Exercise 5



5) $g(x) = h(x) * f(x)$

$$\text{Fourier}\left(g(x) = h(x) * f(x) \right)$$

$$G(w) = H(w) \cdot F(w)$$

$$F(w) = \frac{G(w)}{H(w)}$$

$$F(w) = \frac{h(x) * f(x)}{h(x)}$$

$F(w) = f(x)$, where as we mentioned before

$F(w)$ is the Fourier transform of $f(x)$.

Thus $X(w) = F(w) = FT(f(x))$

Figure 5: Fifth Exercise

Using a small kernel size is a bad idea as there are occasions in which important information present in the images might be lost. Smaller kernel size effectively means that we take into account a smaller number of pixels, compared to having a normal kernel size, and the details present on those pixels are not even considered, yielding strong information loss. A proper adjustment must be made so as to not use smaller than required kernel sizes, but also make sure that when the data given is relatively small, you should also choose a small kernel size. Appropriate toggling must be made, based on the given data, so as to ensure no information loss.

The following image shows a visualization of the Gaussian filter h(x) and its windowed version:

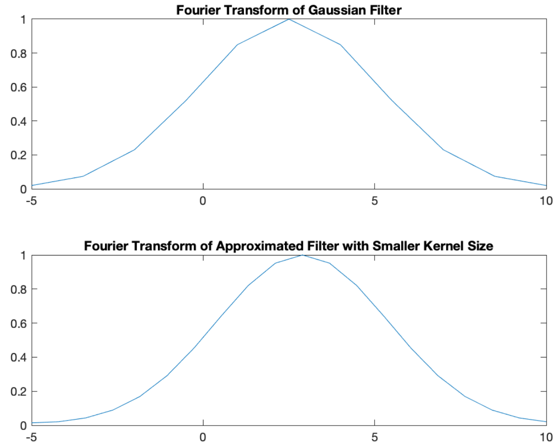


Figure 6: Gaussian filter vs windowed Gaussian filter

6. Gaussian Filtering

The 3 images, the original one, the spatial filtered, and the frequency filtered image, for σ=10 are shown below:

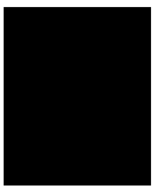


Figure 7: Original

Figure 8: Spatial filtered

Figure 9: Frequency filter

For σ=100, the images shall look like this:

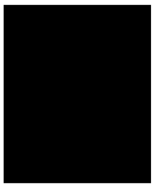


Figure 10: Original

Figure 11: Spatial filtered

Figure 12: Frequency filter

Bonus: As seen from the graph below, the performance of equivalent filtering in spatial and temporal domains depends on σ_s. The results are shown below:

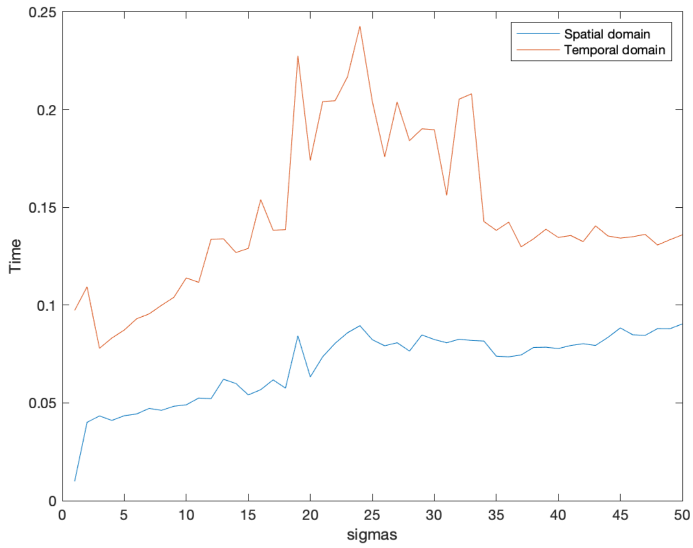


Figure 13: Filtering in Spatial-Temporal Domain

As seen from the plot, the temporal domain is greater than the spatial domain, as the spatial domain is computationally less expensive as it convolves a relatively small pixel neighborhood with a filtering kernel, unlike the temporal one which is also a convolution with a filtering kernel, but in this scenario, we are taking the entire image, and not just a neighborhood of pixels. Therefore this means that filtering in the temporal domain is performed faster than filtering in the spatial domain. Notice also that as we increase the number of sigmas, the time taken to execute also increases. This signifies that the number of sigmas is proportional to the time taken to perform the filtering, in any of the 2 domains, but each with relatively different time taken to execute.

7. Image Restoration

The filtering procedure I chose to remove this repetitive pattern from the image is Notch Filtering.

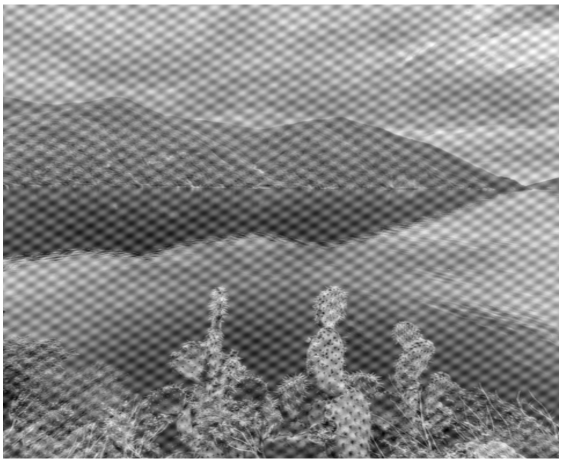


Figure 14: San Domenico

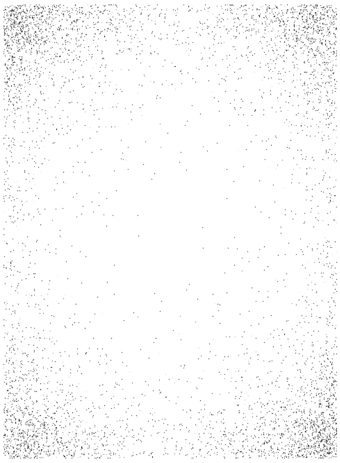


Figure 15: FFT Magnitude Spectrum

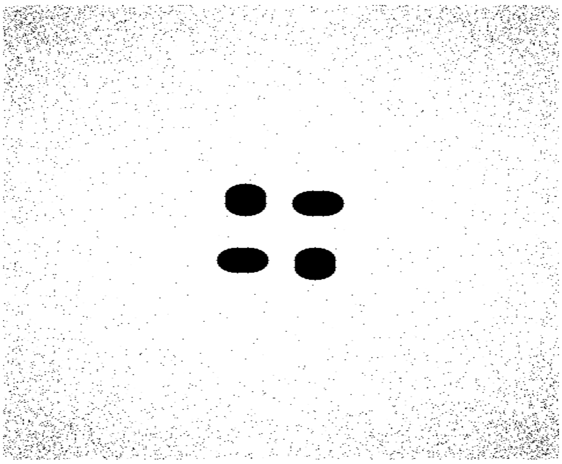


Figure 16: Notch Reject Filter

Figure 17: Restored image

Effectively we start off by applying 2D FFT to the provided image and then shift the low-frequency components (i.e. zero-frequency component) towards the center, as by default the low-frequency components are by default in the corners of the image. By shifting them towards the center, it is easier to apply filtering procedures to the image. Following we calculate the magnitude spectrum of the shifted 2D FFT. For the next step, we define a function that creates a notch reject filter, taking into account shape, notch width and notch frequencies. Effectively, it takes into account the Euclidean distances from the current point to two centers, one with positive and one with negative frequencies. It then uses as a threshold the aforementioned notch width and checks whether any of the two distances is less than or equal to the notch width. If that's the case, that element is set to 0 and thus rejected, otherwise, it is set to 1 and accepted. As a last step, we apply the inverse 2D FFT to the inverse shifted FFT of the element-wise multiplication of the 2D FFT and the notch reject filter, after shifting the spectrum to the center.

8. BONUS: Image Interpolation Analysis

Bonus: we want to double the size of an image.
Let I be the original image and $F(x,y)$ its fourier transform.
The FT of the NN Interpolated image is $G(x,y)$ and $H(x,y)$ the one of LI image.
Because we want to double the size of the original image, we can rewrite the FT of the upscaled image as:

$G(x,y) = F(x/2, y/2)$, with respect to the original img

and $H(x,y) = F(x/2, y/2)$

To estimate the missing values in NNI image, we use $(\text{sinc})^2$ and for LI image sinc function, they're represented using a box filter and a low-pass filter respectively.
Hence:

$$B(x,y) = \text{sinc}^2(x/2) \cdot \text{sinc}^2(y/2)$$

and $\text{LowPass}(x,y) = \text{sinc}(x/2) \cdot \text{sinc}(y/2)$

Taking the FT of NNI image

$$G(x,y) = F(x/2, y/2) \cdot B(x,y)$$

and for LI img

$$H(x,y) = F(x/2, y/2) \cdot \text{LowPass}(x,y)$$

the ratio is thus

$$\frac{G(x,y)}{H(x,y)} = \frac{\cancel{F(x/2, y/2)} \cdot B(x,y)}{\cancel{F(x/2, y/2)} \cdot \text{LowPass}(x,y)}$$

$$= \frac{\text{sinc}^2(x/2) \cdot \text{sinc}^2(y/2)}{\text{sinc}(x/2) \cdot \text{sinc}(y/2)} = \text{sinc}(x/2) \cdot \text{sinc}(y/2)$$

Thus ratio is the product of the sincs between x and y, and hence NNI performs better when applying a low-pass filter and maintains most significant details of an image. Linear Interpolation though is responsible for semantically important image features such as edges of objects, textures, patterns etc.