CS371 Digital Image Processing
Exercise 4 part 1
Discrete Space Fourier Transform & Image Segmentation

Alexandros Angelakis
`csd4334@csd.uoc.gr`

December 23, 2022

1 Discrete Space Fourier Transform

In this section, we are going to look at two different 1D filter, h(m) and g(m), where m is an integer vector from -5 to 5. We will compute the Discrete Space Fourier Transform of the two 1D signals and their product, in the interval $[-\frac{1}{2}, \frac{1}{2}]$, and plot their magnitude.

First of all, let's look at the plots of the 1D signals in the time domain.

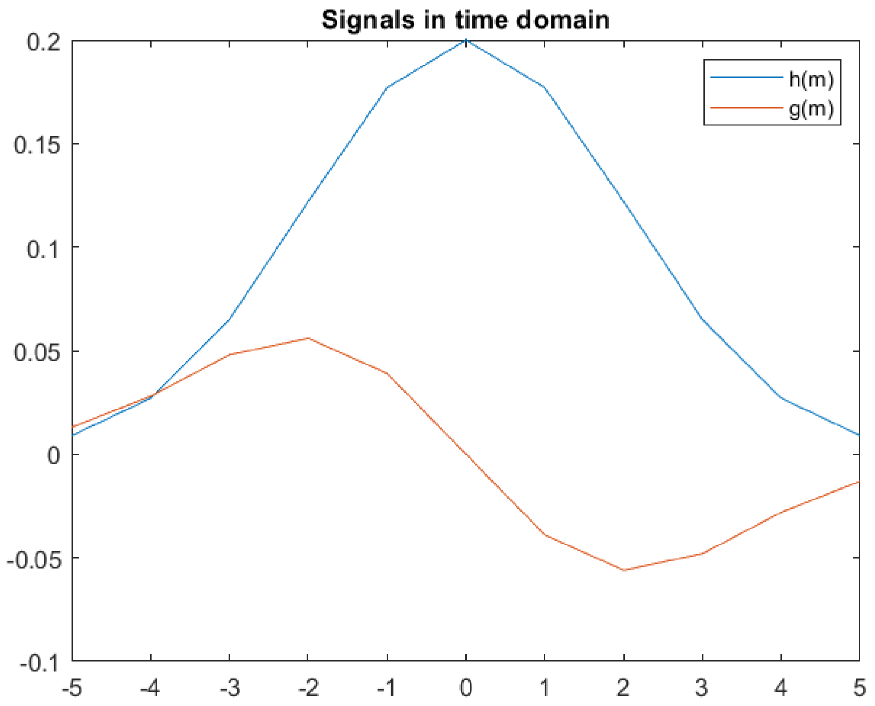

Figure 1: The 1D signals in the time domain

Let's now take a look at the 1D signals in the frequency domain.

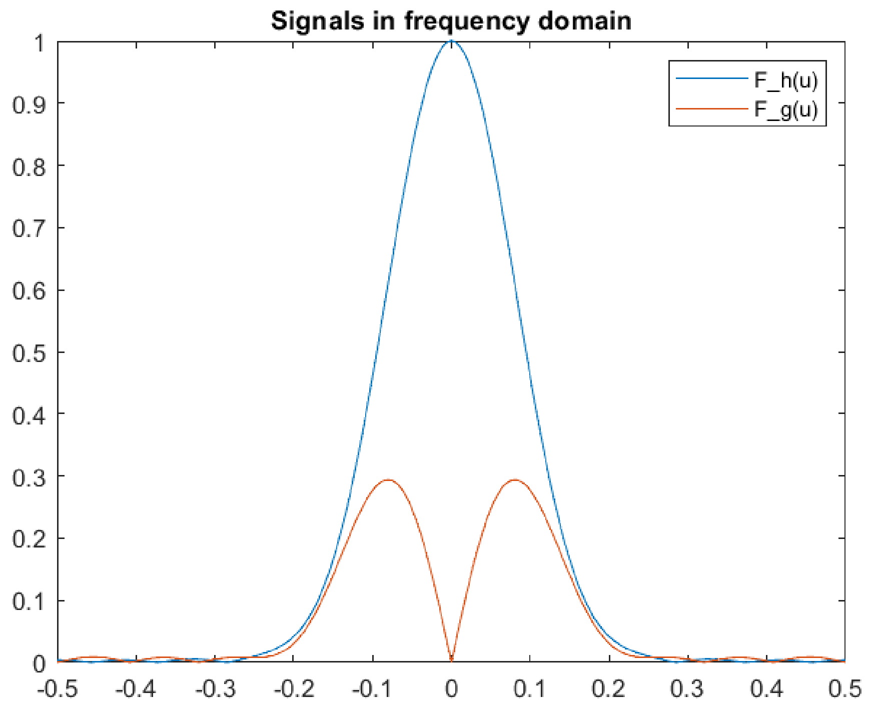

Figure 2: The 1D signals in the frequency domain

As we can see from the figure, the two signals are pretty different in the frequency domain. The blue filter takes all the low frequencies of the image and ignore the high frequencies. To be even more clear, the blue filter, $F_h(u)$, is a low pass filter, meaning that it only allows the low frequencies to pass through, without changing the magnitude of the zero frequencies. It cuts all the frequencies that are greater than $\approx 0.28 = \frac{\pi}{11}$, so the cutoff frequency is $\frac{\pi}{11}$. The red filter, $F_g(u)$, behaves like a band-stop filter, meaning that it passes most frequencies in the range $(0, \frac{\pi}{11})$ and $(0, \frac{\pi}{11})$, lowering their magnitude, but cuts completely off all the zero frequencies and all outside the frequencies $-\frac{\pi}{11}$ and $\frac{\pi}{11}$
To sum up, the blue filter acts like a low pass filter, with cutoff frequency equal to $\frac{\pi}{11}$, and the red filter acts like a stop-band filter, it allows a specific range of frequencies to pass through the filter, lowering their magnitude, and it completely cuts off the frequencies outside $-\frac{\pi}{11}$ and $\frac{\pi}{11}$.

Now let's see the 2D filter h(m)g(n), meaning the convolution of those two 1D signals in the time domain and their product in the frequency domain. I calculated the 2D signal in the time domain with two methods. For the first method, I just did the convolution between the two 1D signals, and for the second method, I computed the Fourier transform for both the 1D signals, multiplied them and then got the inverse Fourier transform.

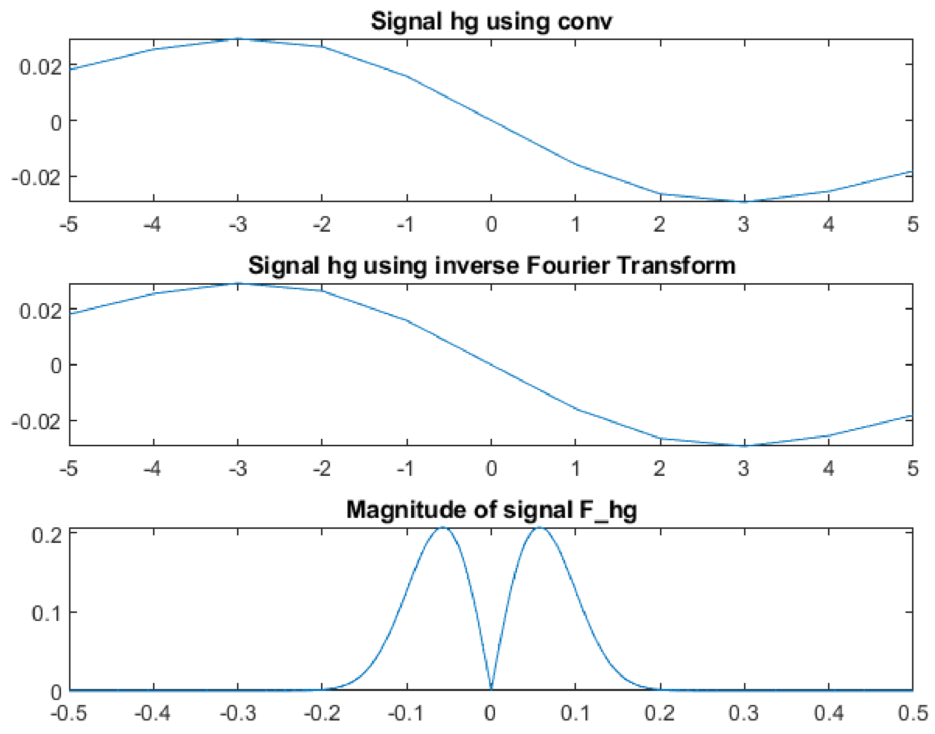

Figure 3: The 2D signal in the time and frequency domain

As we can see, in the frequency domain our 2D filter is similar to the F_g filter (red filter) in the frequency domain. That happens because we multiply those two signals, F_h and F_g in the frequency domain and if we look at their plot, the result of this multiplication will be the F_g but slightly "smaller".

2 Image Segmentation

In this section we are going to segment our image based on the unsigned direction of the vector θ (we'll talk about it later) using the K-means clustering algorithm. The image we are going to cluster is the following:

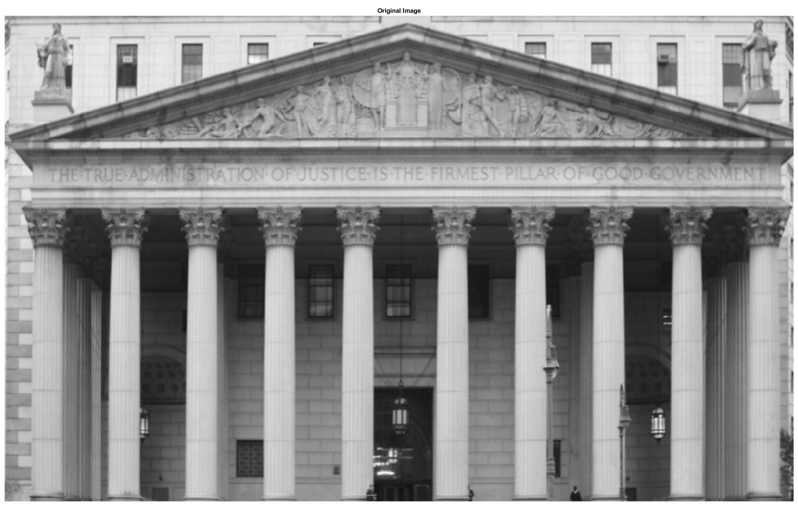
Original Image
THE·TRUE·ADMINISTRATION·OF·JUSTICE·IS·THE·FIRMEST·PILLAR·OF·GOOD·GOVERNMENT

Figure 4: The original image

We'll start by utilizing the filter $h_1 = h(m)g(n)$ and the filter $h_2 = g(m)h(n)$, where h, g are the 1D filters from the previous section and m refers to rows, whereas n refers to columns. We then compute the filter responses $y1, y2$ of the image I with the filters h_1, h_2 respectively. Let's take a look at the images after filtering them.

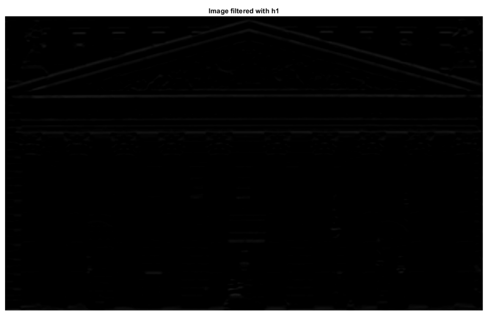

Figure 5: Filter response $y1$ of image I

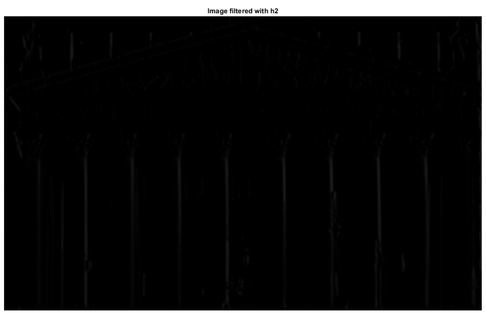

Figure 6: Filter response $y2$ of image I

As we can see, both filters are doing an edge detection for the image, not so visible, where the filter h_1 is doing a horizontal and diagonal edge detection, unlike h_2, which does a vertical edge detection.
Now let's go to the segmentation part of the image. As I mentioned before, we are going to segment our image using the K-means clustering algorithm with 5 clusters. The first cluster has all the values of $A = y_1^2(m,n) + y_2^2(m,n)$ that satisfy the relation $A(m,n) \leq \mu$, and the rest clusters will have as initial cluster centers the values $0, 0.15\pi, 0.35\pi, 0.5\pi$. After we run the K-means algorithm for the image I, we get the following result:

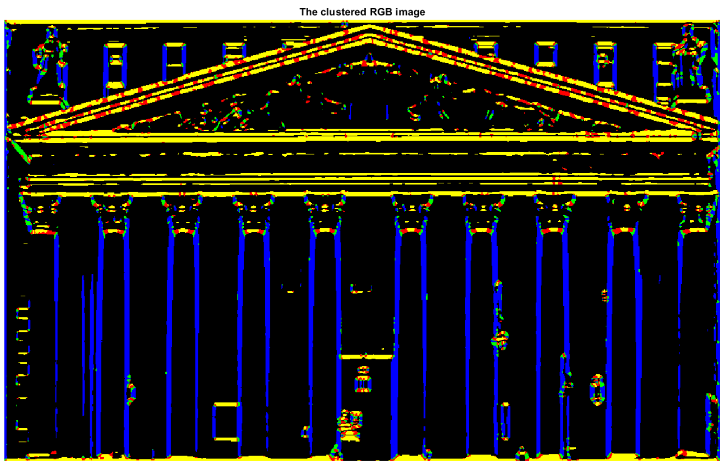

Figure 7: The clustered image

First things first, the obvious thing we can see from this clustered image is that every horizontal and diagonal edge is coloured yellow and every vertical edge is coloured blue. The thing that is less obvious is why some pixels are red or green. If we look closely, we can see that a pixel is coloured red when the angle of that facade is kind of acute and green when is more obtuse. This is not always the case tho, we can see on the roof of that building that there are red spots which is strange, maybe the K-means algorithm did not clustered those pixels correctly. I think we can say that the clustering algorithm has erroneously grouped together those pixels. Other than that, it did a pretty good job clustering the image, always based on the unsigned direction vector θ, we can clearly see the outline of the building's facade .

To sum up, the image is coloured black if the square magnitude of of the two responses is less or equal than its mean, coloured blue and yellow if the image has vertical and horizontal/diagonal edges respectively and finally coloured red for acute and green for obtuse angles in the image.