Department of Computer Science

Digital Image Processing

CS-371

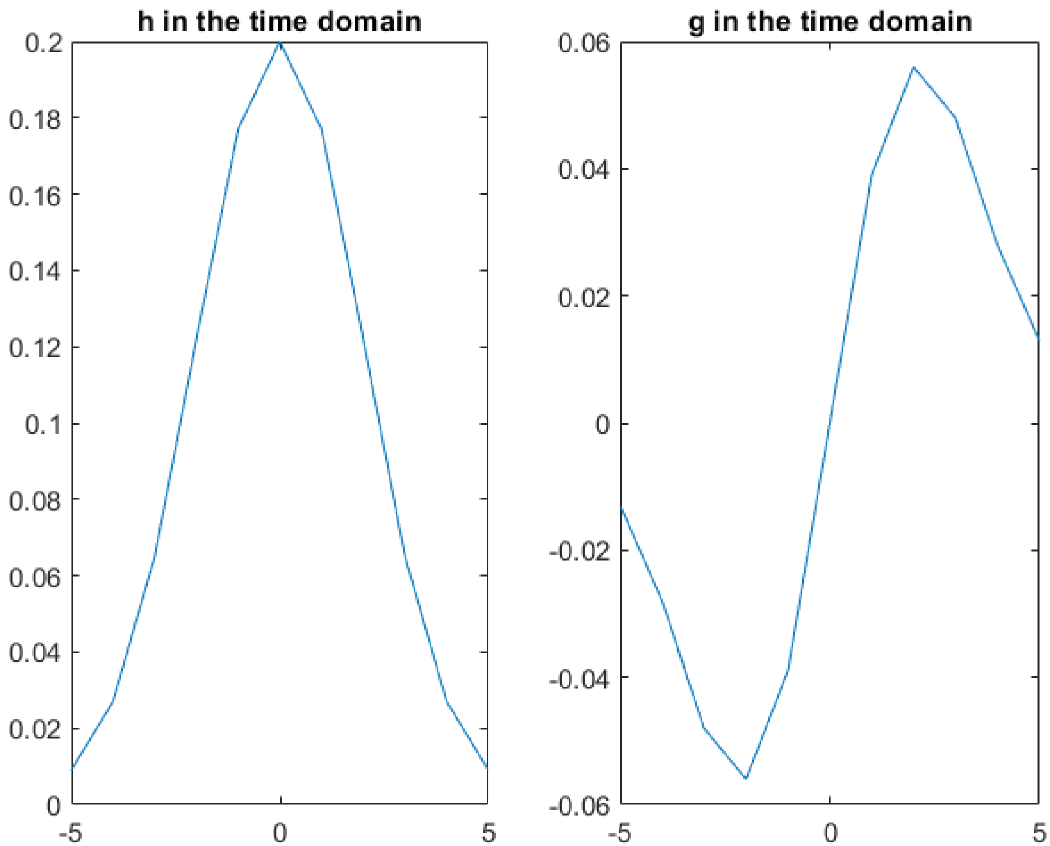
Assignment 4 - Playing with Images in the Frequency Domain

Vasileios Papageridis -csd4710

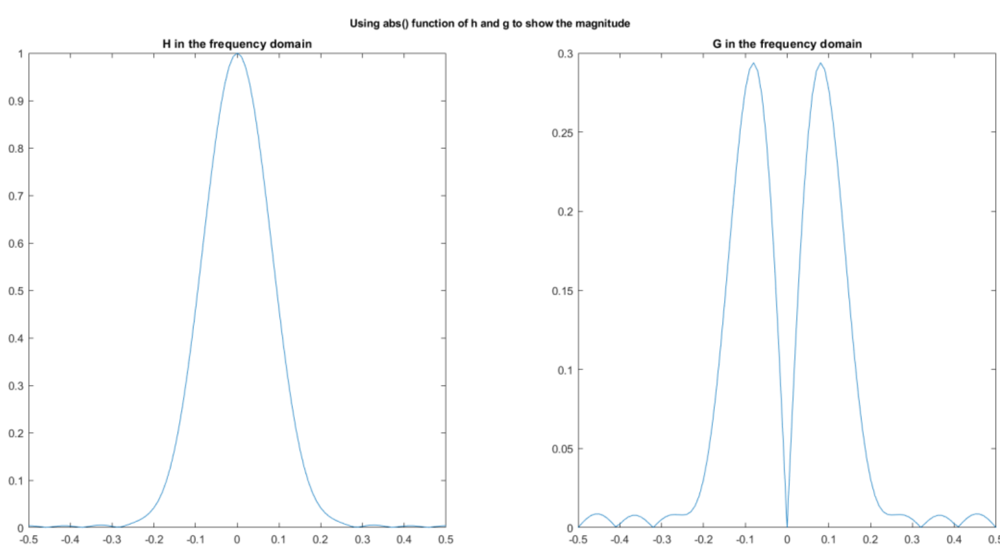
Phase 1 of the Assignment
Discrete Space Fourier Transform & Image Segmentation

1.1 Discrete Space Fourier Transform:

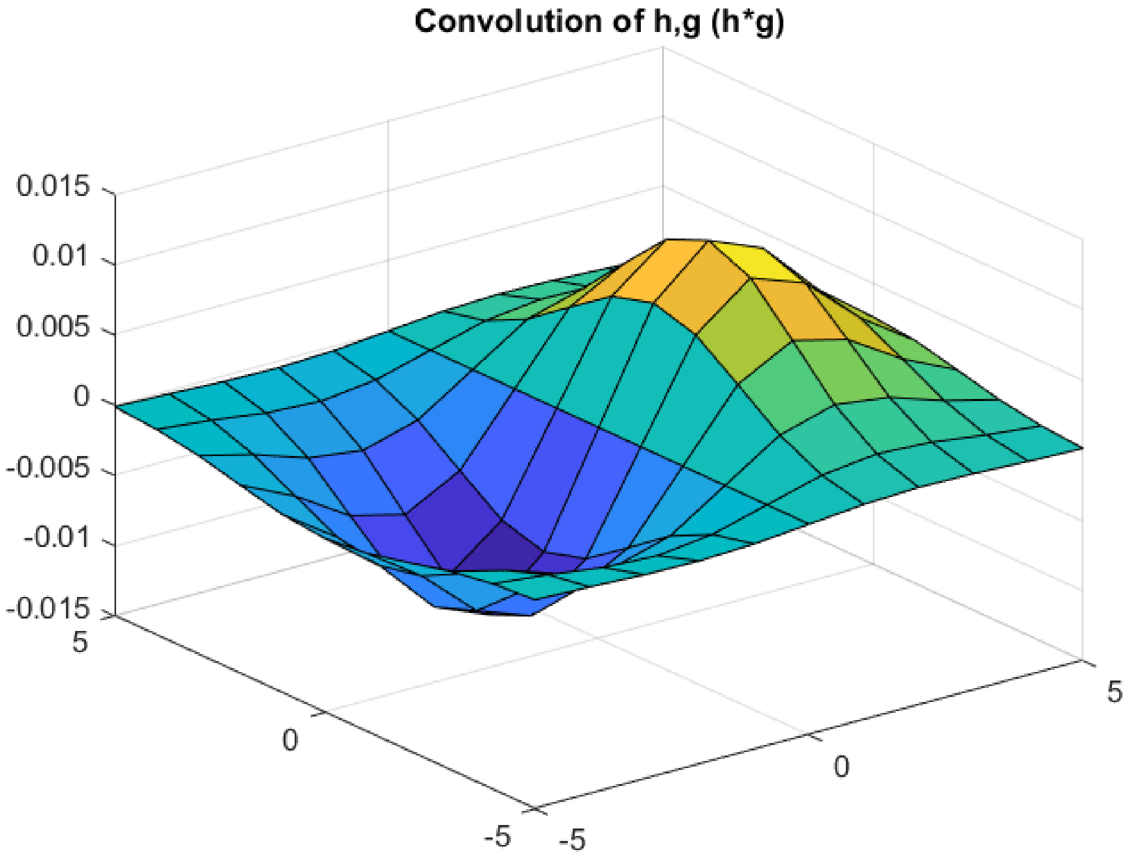
As we can observe from the graphs below (1.1.2) in which our filters are in the frequency domain, h(m) filter is a Gaussian low-pass filter. This means that it allows low frequencies to pass and it will help apply a smoothing effect on the image. Also, g(m) is a band-pass filter that allows frequencies in an area that will help us to detect edges in our image.



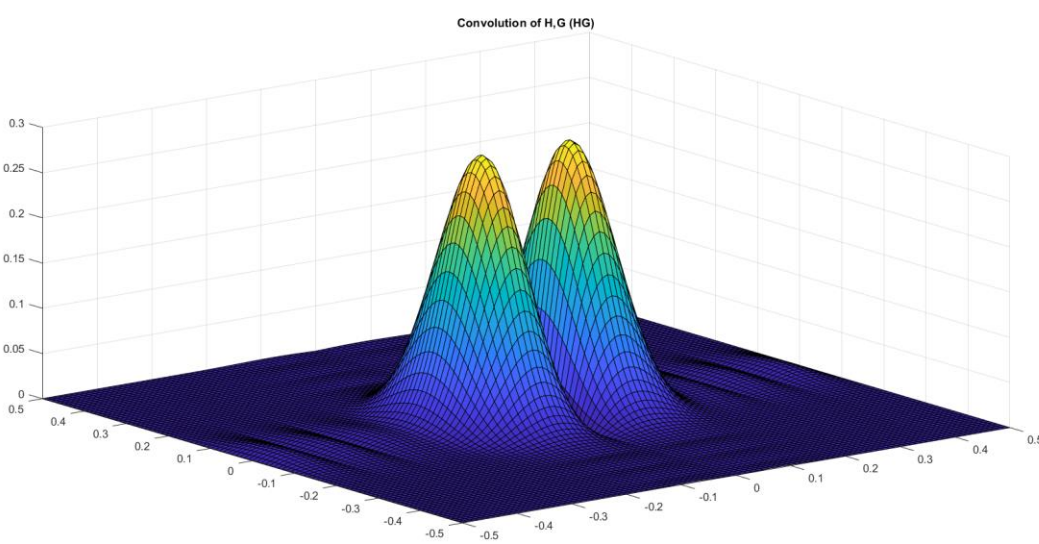
(1.1.1). h & g signals in the time domain.



(1.1.2). h & g signals in the frequency domain.



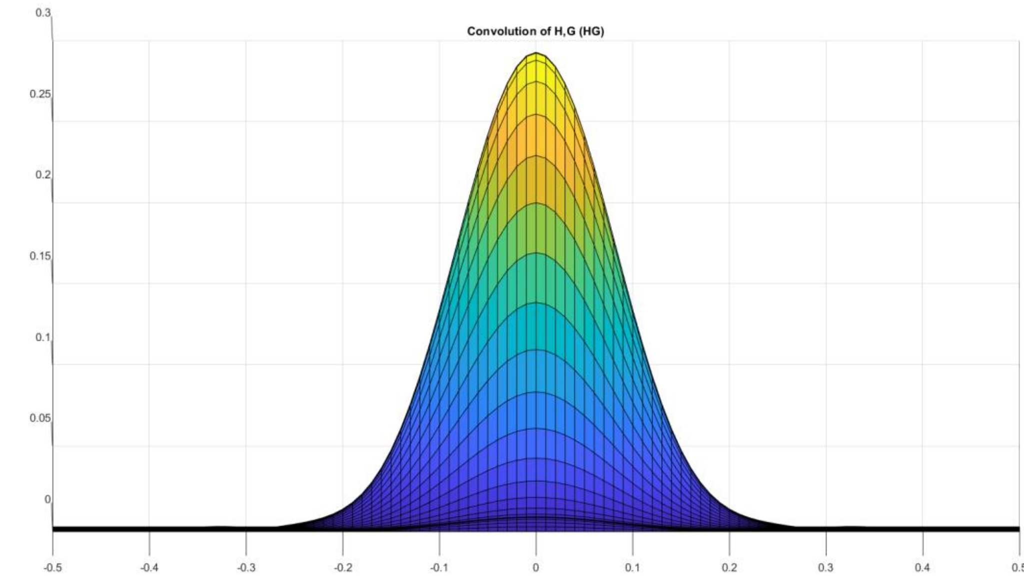
(1.1.3). Convolution of h,g in the time domain.



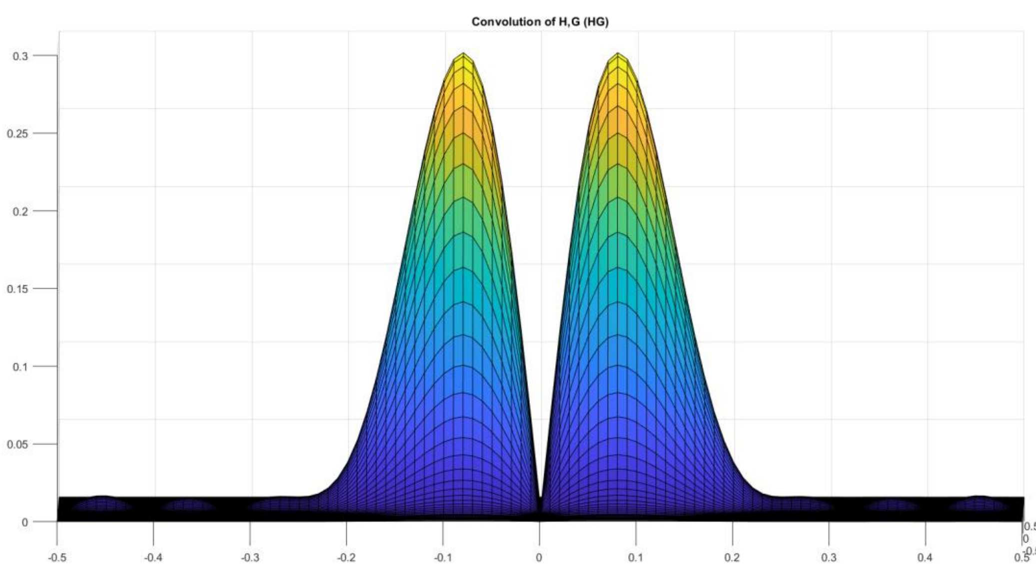
(1.1.4). Convolution of H,G in the frequency domain.

When we calculate the convolution of the 2 filters h and g in the time domain and as a multiplication in the frequency domain, we have as a result a 2D filter. As we can see in the figures we can observe the effect of every filter in the corresponding axis.

In the figures below we make some clear observations. The first figure (1.1.5) is the convolution of h and g in the frequency domain, rotated properly on the X axis and we can see that it represents the effect of filter h. On the other hand, when we rotate the figure properly on the Y axis(1.1.6) we can see that it represents the effect of the filter g.



(1.1.5). Convolution of H,G rotated on the X axis.



(1.1.6). Convolution of H,G rotated on the Y axis.

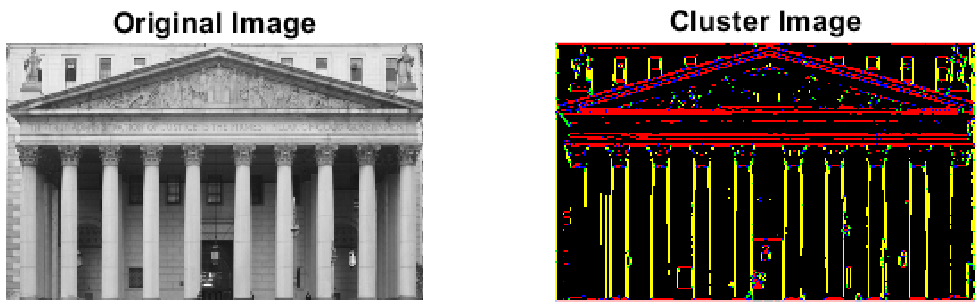
The conclusion is that the filter is detecting edges horizontally and applies smoothing vertically.

1.2 Image Segmentation:

In this part of the assignment, we applied the 2D filter that we created earlier so we can achieve the edge detection that we want. In particular, as we mentioned earlier the Gaussian filter h is used to smooth the image, to block the high frequencies pass and allow the details that we are interested to. The band-pass filter g is used to detect the edges that we want, so we can cluster the image.

After applying clustering and color mapping for the groups we can observe that every time that there is a big difference between the intensity vertically, those spots are colored yellow, which means that the value of θ is low (close to 0). Big difference between the intensity horizontally means that the value of θ is high (close to $\frac{\pi}{2}$) and pixels colored red. Edges that are slightly blunt are colored green, while slightly sharp edges colored blue and this happens if θ is between the average of those 2 edges we discussed. Finally, when the intensity doesn’t change which means that the square magnitude is lower than the average, those pixels are colored black.

Lastly, we can observe that lines with different orientations have been erroneously grouped together. We can clearly see that at the roof of the building (especially there but also in some smaller parts of the image) there are some blue points mixed with the red ones. This happens because there are some edges that are not so sharp with the vertical axis and some other are really sharp with the vertical axis. We can also see the same problem with a mix of green and yellow points in some parts of the image. This happens because, same as before, there are some edges that are not too blunt with the horizontal axis and there are some that are really blunt to the horizontal axis. In order to avoid this problem we could add some more classes with more strict conditions so we could achieve better groups.



(2.1). Original Image and Image with colored edges.