Bijan Fakhri
2/16/17

EEE 508 – Digital Image and Video Processing & Compression
HandsOn 2

Problem 2.1

Question 3 – Display the Image using imshow()

Question 5 – Image output of y1

Question 6 - Compare y1 and y2 and Conv2() and Filter2()

The images y1 and y2 are visibly nearly identical. Image y1 though is defined with single floating point values while y2 double. Also, when taking the mean square error of the difference between the images you get a very low but nonzero error: 4.7651e-11, proving they are in fact not identical. This is likely due to implementation differences. Conv2 implements classic 2D convolution while filter2 implements 2D correlation as per the MATLAB documentation.

- Image output of y2

Question 7 – Show that 'h' is separable

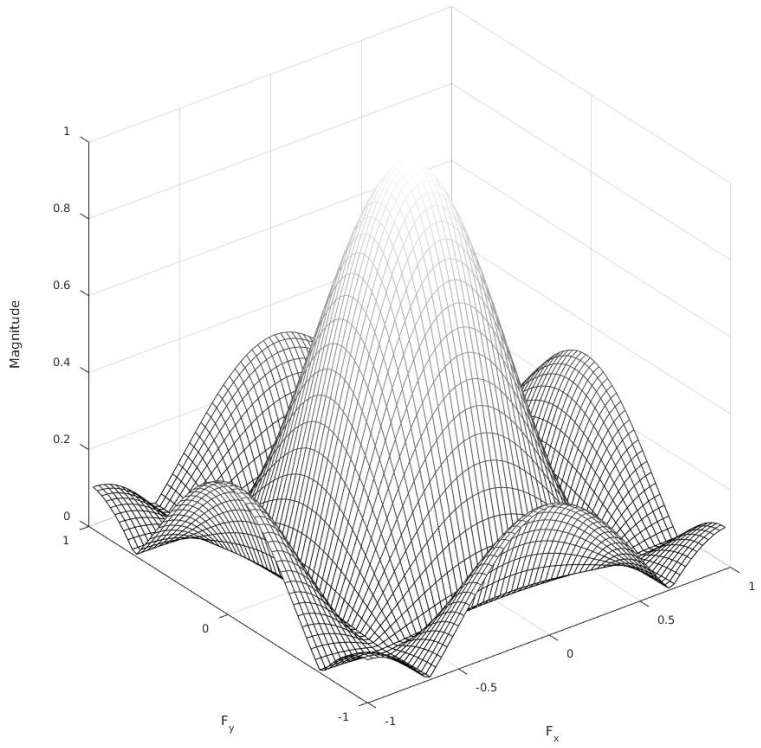
The 'h' can be separated into two 1D matrices as follows:
h1(n1) = [1,1,1]/3 and h2(n2) = [1;1;1]/3
h1h2 = h = [1,1,1
1,1,1
1,1,1] / 9

y1 and y3 are identical.

- Image output of y3

Question 8 – Low Pass or High Pass?

The filter h(n1, n2) is a LOW PASS filter. There are several ways to deduce this but in terms of the image applying the filter produces a smoothing effect, reducing high frequency content. It is also deducible that the filter is a low pass filter because of its frequency response plot:

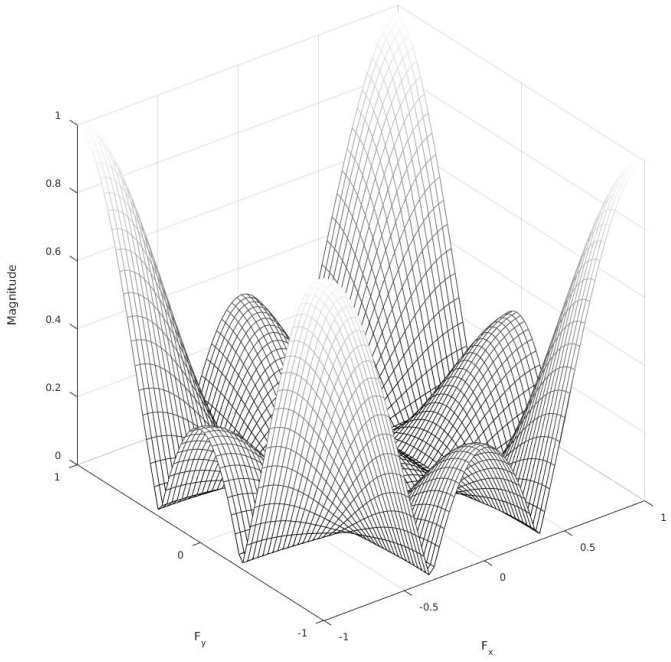


As you can see, the center (DC) components pass through the easiest while the higher frequencies are attenuated. The filter was converted into a high pass filter by “pushing” the frequency domain representation of the filter in both the n1 and n2 directions until the center of the frequency response plot became the low point while the corners became the high points. Converting back from the frequency domain we get the high pass filter ‘little_h’.

The following code produces these results:

```
H = fft2(h, 128, 128);
l1 = circshift(H, 64, 2);
l2 = circshift(l1, 64, 1);
little_h = circshift(circshift(ifft2(l2), 64, 2), 64, 1);
freqz2(little_h)
```

A graph of the frequency response of the high pass filter is below:



Question 9 – Results of filtering with the high pass filtering

Filtering with our new high pass filter we get this image (y4):

You can see that the high frequency components were retained, mostly edges!