Digital Image Processing

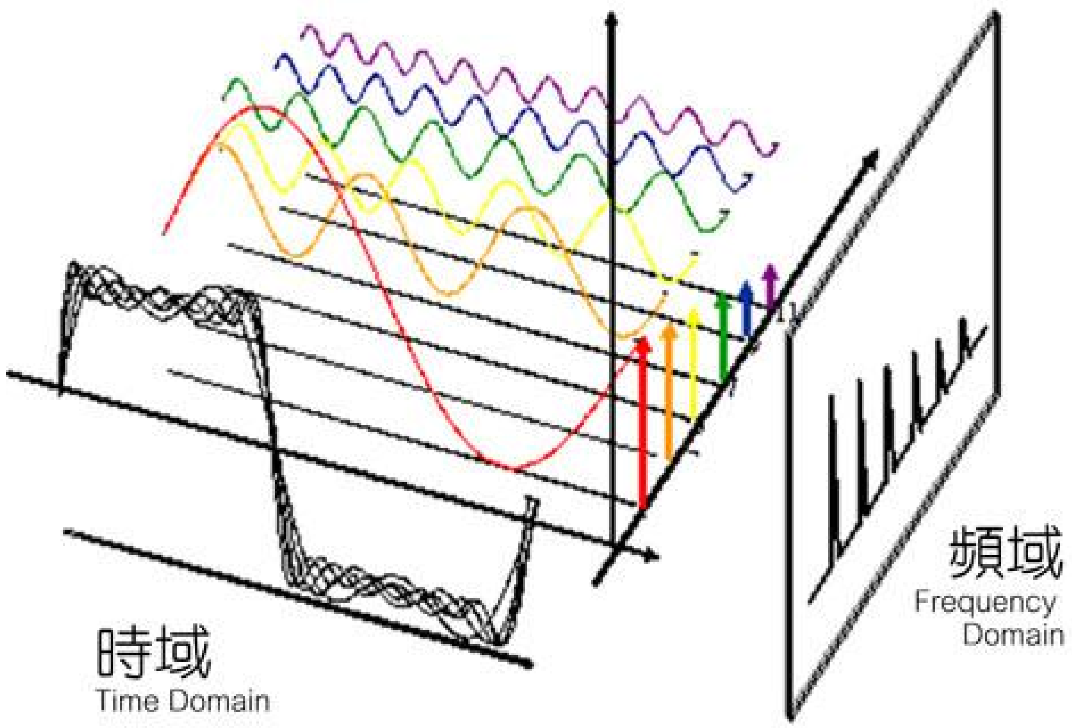
Exercise 03

Jiaqiao Peng 406038

Part 1: Theory

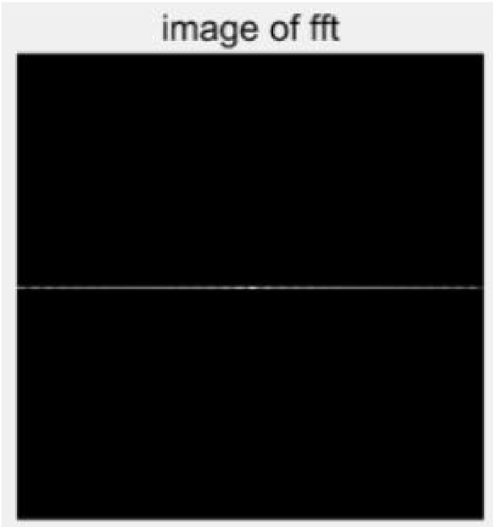
i) When adding borders to the image, it padded the image with a constant value, probably 0. So the white part will be influenced by black borders (greyscale is 0).

From the image shown below we can understand. The original image is like a square wave along horizontal direction. Average filter is a low-pass filter. Without high frequency sine wave, the fitting result is much worse than square wave, especially on the edge of square wave. So we can see the left and right is almost half of the amplitude of square wave, as well as the gray edge of image c).



ii) When adding borders to the top and bottom of the image, the row or column at the very edge of the original is replicated to the extra border. When adding borders to the left, we using constant value 255 (white). And 0 for the right borders.

iii) The frequency domain graph of original image is shown below:

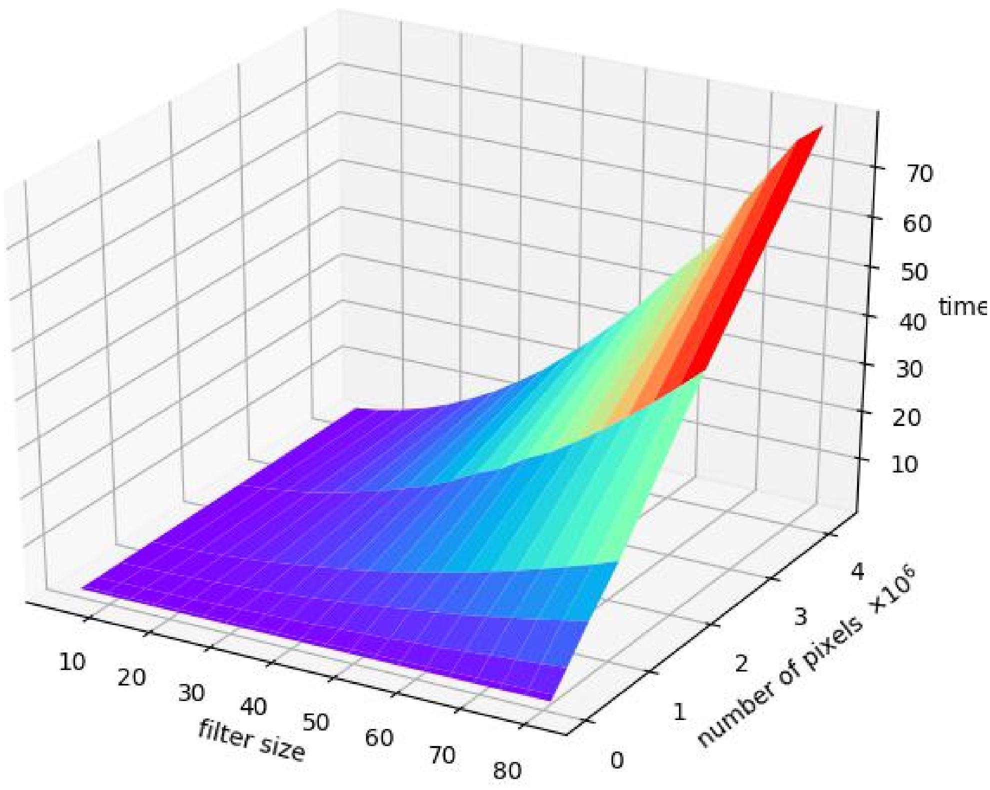


It doesn’t contain component along vertical direction. So we should pad the top and bottom of original image first.

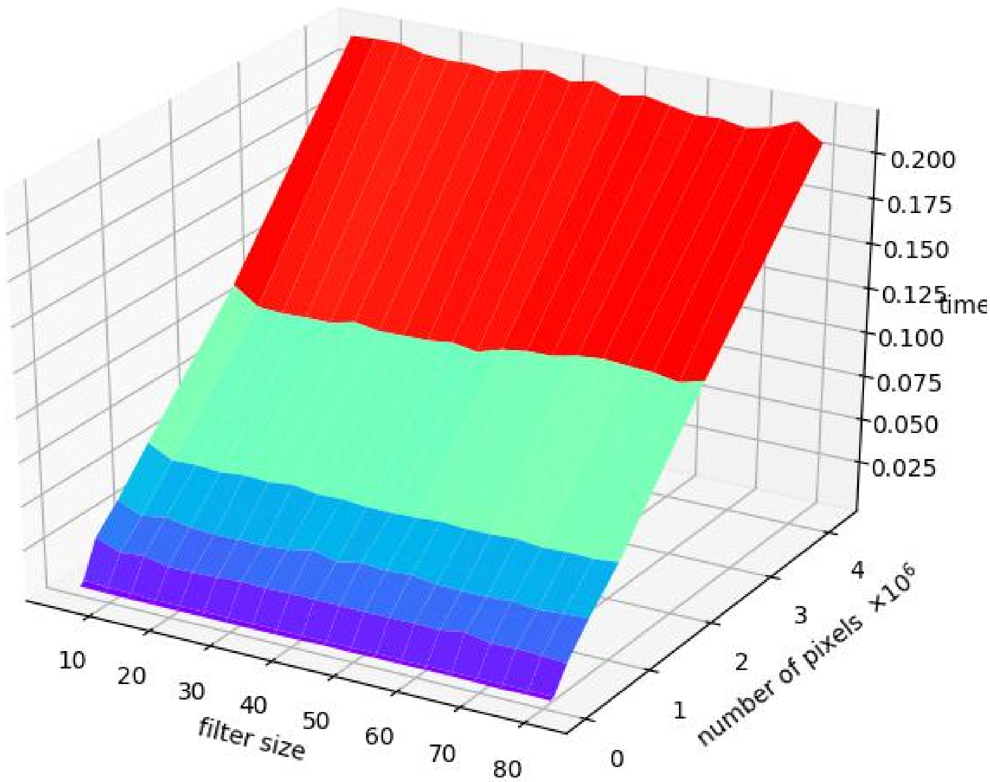
Part 2: Practical

The parameters for the output images are: thresh = 0, scale = 5.

The time behaviour of convolution by usage of spatial domain is shown below:



The time behaviour of convolution by usage of frequency domain is shown below:



There are 7 different size of input image: 128*128, 256*256, 512*512, 768*768, 1024*1024, 1536*1536 and 2048*2048, along with 20 different size of filter from 5*5 to 81*81. The plot tool is python with matplotlib.

The time complexity of spatial domain convolution is $O(MN^2)$, where M is the number of pixels and N is the filter size. So when the filter and image size get larger, the time consumption increases sharply. But the time complexity of frequency domain convolution is only relative to the number of pixels. So the time consumption grows equally along with the growth of number of pixels.

The time consumption of frequency domain convolution is much less than that of spatial domain frequency. When the size of image and filter become larger, time saved is considerable. Besides, after implementation of expanding the image to an optimal size (the performance of a DFT is dependent of the image size. It tends to be the fastest for image sizes that are multiple of the numbers two, three and five), it can save 1/3 more time.

After implementation of separable filters, the time performance improved a lot. For filter size N*N, the time complexity of spatial domain convolution reduced from O(N*N) to O(2N). Bigger filter means more time saved.

The time performance of box filter with integral image is only relative to the number of pixels, in a general way. Because for any filter size, it still needs to compute add and substract of values in integral image.