

# FILTERS AND FREQUENCIES 3

CSE 152: INTRO TO COMPUTER VISION

October 19, 2018

## 1 Filters

1. What does it mean for a filter to be shift-invariant?

**Solution:** If a filter is shift-invariant, it will perform the same high-level computation (e.g. taking a median of a neighborhood) no matter which part of the signal it is being applied to. The filter will do the same thing no matter where we put it.

Mathematically, this means  $[I(x, y) * f](x - n, y - k) = [I(x - n, y - k) * f](x, y)$ .

2. Convolve the image  $F$  with filter  $h$ . Assume ones beyond the boundaries.

$F$

1	0	5	2
0	0	3	5
0	0	5	4
2	2	0	1

$h$

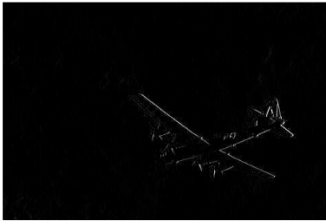
0	0	0
0	0	0
0	0	1

**Solution:**

$F * h$

1	1	1	1
1	1	0	5
1	0	0	3
1	0	0	5

## 3. Match the images to the descriptions.



original image

filtered with  
[[1, -1]] kernelfiltered with  
[[1], [-1]] kernelfiltered with  
Gaussian kernel

**Solution:** From top to bottom: *filtered with*  $[[1, -1]]$  *kernel, filtered with Gaussian kernel, filtered with*  $[[1], [-1]]$  *kernel, original image.*

## 4. One application of filtering/convolution is template matching: finding regions in an image that are similar to a given patch. How would you imagine that this is done?

**Solution:** Convolve the image with the (flipped) patch. In the resulting image, a high brightness at a pixel means that the patch is very similar to the neighborhood around that pixel.

5. Give a  $3 \times 3$  linear filter that shifts an image one pixel to the right and increases the image brightness by 50%.

**Solution:**

0	0	0
1.5	0	0
0	0	0

assuming correlation,

0	0	0
0	0	1.5
0	0	0

assuming convolution.

6. How do you obtain an edge image if you're only allowed a blurring filter?

**Solution:** Apply the blurring filter, then subtract the result from the original image. (The blurred image represents low frequencies. After subtracting these out, we are left with the high frequencies.)

## 2 Frequencies

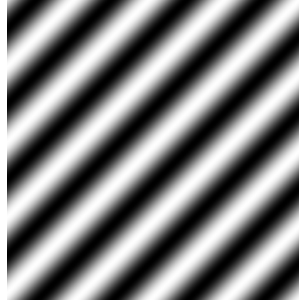
1. Say you have a linear filter in the form of a convolution kernel. How can you apply your filter to an image in frequency space? Why might it be advantageous to do so?

**Solution:** Zero-pad the filter (or the image, if it is smaller) to make the filter and the image the same shape. Use the fast Fourier transform (FFT) to convert the filter and image into their frequency representations. Perform an elementwise multiplication between these frequency representations. Convert the result back into the spatial domain using the inverse FFT.

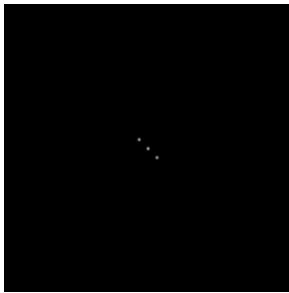
The main reason to filter in frequency space is efficiency. A sliding-window convolution is  $O((\text{image size}) \cdot (\text{filter size}))$ . A frequency-space convolution is  $O(n \log n)$ , where  $n = \max\{(\text{image size}), (\text{filter size})\}$ . If the filter size is greater than the log of the image size, filtering in frequency space is probably faster.

### 3 Fourier Domain Visualizations

1. Sketch the Fourier domain image for this function. For convenience, when drawing you can pretend that brightnesses are inverted (i.e. draw black as white, and white as black). Assume that the peak-to-peak distance in the direction of the sinusoid is 64 pixels. Label significant points on the Fourier domain image with their frequencies.



**Solution:**



frequency magnitudes from top left to bottom right:  $\frac{1}{64}$ ,  $0$ ,  $\frac{1}{64}$

2. These are frequency representations of image filters. What does each filter do?



**Solution:** They are both low-pass (blurring) filters. On the right is a box filter.