ECE 285 – Assignment #7

Writing part

1 Exercise 1 (Wiener-Khinchin theorem)

The power spectral density of a discrete and periodical image f is defined as the square modulus of its Fourier transform:

$$(S_f)_{u,v} = |\hat{f}_{u,v}|^2 = \left| \sum_{k=0}^{n_1-1} \sum_{l=0}^{n_2-1} f_{k,l} e^{-i2\pi\left(\frac{uk}{n_1}+\frac{vl}{n_1}\right)} \right|^2 . \tag{1}$$

Show that the power spectral density of f is the Fourier transform of its auto-correlation

$$(S_f)_{u,v} = (\hat{r}_{ff})_{u,v} \quad \text{where} \quad r_{ff} = f \star f . \tag{2}$$

Practical part

2 Exercise 2 (Spectral deconvolution)

An important ingredient for spectral deconvolution is the concept of mean power spectral density. The mean power spectral density is the expectation of the power spectral density for the given population (subset) of images that we are targeting. The formal definition is

$$S_{u,v} = \mathbb{E}[(S_f)_{u,v}] \tag{3}$$

where the expectation $\mathbb{E}$ is relative to a probability space on which images f are seen as random vectors. In theory, to determine the mean power spectral density, one has to choose a probability space for f and evaluate the above expectation (which usually leads to compute an integral).

Instead, we will assume that $S_{u,v}$ follows a power law of the form

$$S_{u,v} = n_1 n_2 e^{\beta} \omega_{u,v}^{\alpha} \quad \text{for} \quad \omega_{u,v} \neq 0 \quad \text{where} \quad \omega_{u,v} = \sqrt{\left(\frac{u}{n_1}\right)^2 + \left(\frac{v}{n_2}\right)^2}. \tag{4}$$

In a first part, we are going to estimate α and β on a subset of three training images. Once α and β have been determined, we will use them to deconvolve a test image in the second part.

2.1 Part I (mean power spectral density)

1. Create in `average_power_spectrum_density.m`, a function

```
function Savg = average_power_spectrum_density(f)
```

that takes a cell array of images `f{1}`, `f{2}`, ..., `f{K}`, and compute the average of their power spectral density

$$S^{(\text{avg})}_{u,v} = \frac{1}{K} \sum_{k=1}^{K} |\hat{f}^{(k)}_{u,v}|^2 \tag{5}$$

2. Download `assignment7.zip` and extract the data:

   - `assignment7/eagle.png`
   - `assignment7/owls_blur.png`
   - `assignment7/plane.png`
   - `assignment7/sheeps.png`

   Create a script `test_power_spectrum_density.m`, that runs your new function on the three images `f{1} = eagle`, `f{2} = plane` and `f{3} = sheeps`, and display the logarithm of $S^{(\text{avg})}$ as

```
imagesc(fftshift(log(Savg)), [0 30]),
colormap(gray);
axis image;
```

3. Show that except for the 0 frequency ($\omega_{u,v} \neq 0$), we have $\alpha x_{u,v} + \beta = y_{u,v}$ where $x_{u,v} = \log \omega_{u,v}$ and $y_{u,v} = \log S_{u,v} - \log n_1 - \log n_2$.

4. We now assume that $S \approx S^{(\text{avg})}$, and based on the above equation, we are going to estimate α and β from $S^{(\text{avg})}$ by least square linear regression. In our case, the least square linear regression consists in looking for α and β that minimizes the following sum of square errors (SSE)

$$\text{SSE}(\alpha, \beta) = \frac{1}{2} \sum_{u=0}^{n_1-1} \sum_{v=0}^{n_2-1} (\alpha x_{u,v} + \beta - y_{u,v}^{(\text{avg})})^2 \ . \tag{6}$$

   where $y_{u,v}^{(\text{avg})} = \log S_{u,v}^{(\text{avg})} - \log n_1 - \log n_2$. Find the mathematical expression of α and β that minimizes the SSE.

   Hint: they simultaneously cancel the partial derivatives of SSE with respect to α and β.

5. Based on these expressions, write in `mean_power_spectrum_density.m`, a function

```
function [S, alpha, beta] = mean_power_spectrum_density(Savg)
```

   that returns the estimated mean power spectral density S and the values of α and β.
   Set `S(1,1) = Inf`.

6. Complete the script `test_power_spectrum_density.m`, to call your new function in order to estimate S from $S^{(\text{avg})}$. What are the values of α and β?

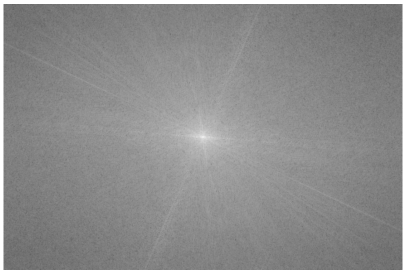
7. Display next to $S^{(\text{avg})}$, the logarithm of S (using the same colormap and range). Display also one-dimensional slices of $S^{(\text{avg})}$ and S (on top of each other), for the frequency $u = 0$ and $u = 50$. Check that your results are consistent with the ones on top of next page.

2.2 Part II (Wiener deconvolution)

1. Create a script `test_deconvolution.m`, that loads and displays the image `g = owls_blur` and the logarithm of its spectrum $\hat{g}$. This image is corrupted by a blur of spread $\tau = 2$, and a noise with standard deviation $\sigma = 1$. Our purpose is to recover the sharp underlying image.

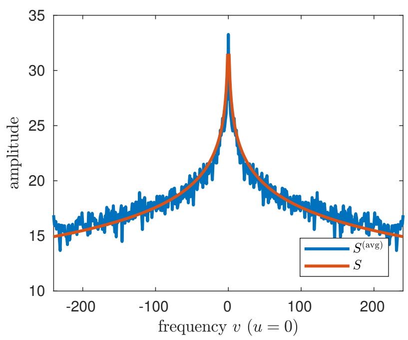
2. In the script `test_deconvolution.m`, use `imkernel2fft` to create the frequency response λ corresponding to an exponential blur with $\tau = 2$. In a first naive approach, we are going to deconvolve this image by simply dividing its spectrum by λ or, equivalently, multiplying it by the transfer function $\hat{h} = \lambda^*/|\lambda|^2$. Compute $\hat{h}$, display its modulus $|\hat{h}|$, display the logarithm of $|\hat{h} \cdot \hat{g}|$ and the deconvolved image as $f = \mathcal{F}^{-1}(\hat{h} \cdot \hat{g})$ (don't forget to take the real part). Interpret the results.

3. Repeat this experiment with the Gaussian and the box kernel. What is the condition number of each of these three deconvolution problems? Interpret the results.

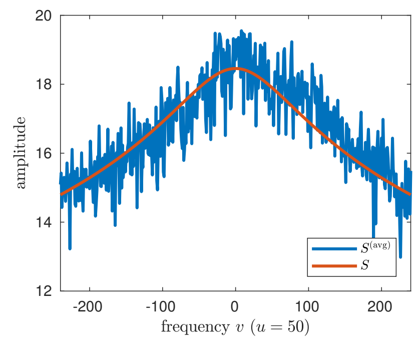


(a) Average $S^{(\text{avg})}$

(b) Estimated $S^{(\text{avg})}$ ($\alpha = -3.0$ and $\beta = .88$)



(c) First slice



(d) Second slice

4. The Wiener deconvolution consists in doing the same thing as the naive version, but instead with the following transfer function

$$\hat{h}_{u,v} = \frac{\lambda^*_{u,v}}{|\lambda_{u,v}|^2 + n_1 n_2 \sigma^2 / S_{u,v}} \tag{7}$$

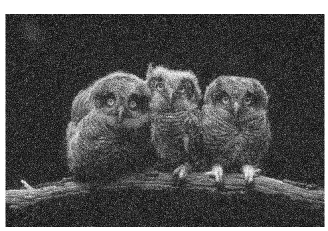
What does this filter tend to do when σ approaches 0? or $+\infty$?

5. In **wiener_deconvolution.m**, create a function

```
function [xdeconv, h] = wiener_deconvolution(x, S, lambda, sig)
```

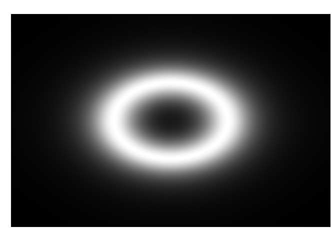
that implements the Wiener deconvolution, returns the result and the transfer function $\hat{h}$. Complete **test_deconvolution.m**, to deconvolve g using the mean power spectrum density S that was learned in the first part of this exercise, and assuming an exponential kernel. Display $|\hat{h}|$, $|\hat{h} \cdot \hat{g}|$ and the deconvolved image f. Check that your results are consistent with the following ones:

(e) Observation g



(f) Naive deconvolution

(g) Wiener deconvolution f



(h) Transfer function $\hat{h}$

6. What does Wiener deconvolution do?

7. Repeat for the Gaussian and the box kernel.
Which of these three kernel functions does fit the best our deconvolution problem?