Lab 07

Basic Image Processing Fall 2019





"Degradation happens"

We don't want a reduced quality image but that's the best the system can capture. Quality loss happens due to motion, out-of-focus lenses, transmission and quantization errors etc.

Usually it is not possible (or affordable) to eliminate the causes of the degradation. Instead, we use restoration approaches to get back the original image from the low quality one.

Let's take it to the next level!



The system is LSI (*linear space-invariant*) so we can Fourier transform it to the frequency domain:

Frequency domain

$$X(\omega_1, \omega_2) \cdot H(\omega_1, \omega_2) + N(\omega_1, \omega_2) = Y(\omega_1, \omega_2)$$

Point spread function & Optical transfer function

We can describe H with an image-pair. Let's create a well-known image and send it through the imaging system. The response will be a modified image. If we assume no noise then the input-output pair perfectly describes the H.

an arbitrary input





the response of the system

If the input image contains only one point (a pixel) then we call the response the *point spread function* and we can use this single image to describe the system.





the response is the point spread function

Point spread function & Optical transfer function

The point spread function can be used to realize the system. An image convolved with the point spread function is the response of the *H* block.



We call the Fourier transform of the PSF "optical transfer function" (OTF) 6



Since we have the model of the degradation system, we can build its inverse.



Now we just have to find n and H^{-1} and we are ready :)



In practice it is impossible to find the exact $n \Rightarrow$ subtraction cannot be done. *However*, we can do some *noise statistics* or *assume* the type of noise.

In practice we don't know the exact *H* either.

However, we can do *measurements* or we can *estimate* the system. Also, we have to *deal with the noise* here.

Therefore, instead of using H^{-1} it is very common to create an R 'recovery' system.

1.



Depending on the restoration approach the *R* system can be...

Inverse Filter	$\operatorname*{argmin}_{x} \left(\left\ y - Hx \right\ ^2 \right)$	$R(\omega_1, \omega_2) = \frac{1}{H(\omega_1, \omega_2)}$
Constrained Least Square	$\underset{x}{\operatorname{argmin}} \left(\left\ y - Hx \right\ ^{2} + \alpha \left\ Cx \right\ ^{2} \right)$	$R(\omega_1, \omega_2) = \frac{H^*(\omega_1, \omega_2)}{ H(\omega_1, \omega_2) ^2 + \alpha C(\omega_1, \omega_2) ^2}$
Wiener Filter	$\operatorname*{argmin}_{x} \left(\mathbb{E}\left[\ x - \tilde{x}\ \right] \right)$	$R(\omega_1, \omega_2) = \frac{H^*(\omega_1, \omega_2)}{ H(\omega_1, \omega_2) ^2 + \frac{P_{NN}(\omega_1, \omega_2)}{P_{XX}(\omega_1, \omega_2)}}$

Now please

download the 'Lab 07' code package

from the

submission system

Exercise 1

Implement the function degradation in which:

- The function has one input and three output arguments.
- Inputs:
 - **x** the input image (grayscale, double, within range [0,1])

• Outputs:

- y the degraded output image
- h the kernel of the imaging system
- n the values of the additive noise layer

Exercise 1 – continued

The degradation kernel is a 'motion' type kernel with parameters len=42, and theta=30. Use fspecial to produce the kernel matrix.

The additive noise is a white noise; the n should be filled with normally distributed random numbers (use randn). The variance should be $\sigma^2 = 0.001$ (Multiply the result of randn by sqrt(0.001) to get the desired variance.)

After creating the appropriate h and n matrices, apply the kernel to the input using imfilter with the options 'replicate', 'same', 'conv', and after this, add the noise layer to the filtered image.

$$y = x \otimes h + n$$

Execute script1.m and check your results!

Original input image





Exercise 2

Implement the function restoration_CLS in which:

- The function has four inputs:
 - y is the degraded image,
 - h is the point spread function of the system,
 - alpha is a regularization parameter, and
 - c is the point spread function of a high pass filter
- The function have to return the restored image (x_tilde).

Exercise 2 – continued

First, move everything to the frequency domain. You have to compute the Y matrix (the degraded image in the frequency domain) using fft2.

After this, compute the optical transfer functions from the given point spread functions. For this use psf2oft(), where the first argument is the PSF to be converted, and the second argument is the desired size of the OTF.

In our case, we want the OTFs to be the same size as Y!

Exercise 2 – continued

After this, implement the formula for the **R** restoration matrix:

$$R(\omega_1, \omega_2) = \frac{H^*(\omega_1, \omega_2)}{|H(\omega_1, \omega_2)|^2 + \alpha |C(\omega_1, \omega_2)|^2}$$

in which H^* means complex conjugate (use the conj() function in MATLAB).

Finally, apply the restoration system to the input in the frequency domain. Multiply **Y** with **R** elementwise and transform back the result to the spatial domain.

$$\tilde{x} = \left| \text{IDFT} \left\{ R \cdot \text{DFT} \left\{ y \right\} \right\} \right|$$

Execute script2.m and check your results!



Degraded image (filter + noise)



Restored image (CLS)



Exercise 3

Implement the function restoration_wiener in which:

- The function has three inputs:
 - y is the degraded image,
 - h is the point spread function of the system,
 - n is a noise layer (an image containing noise only)
- The function have to return the restored image (x_tilde).

Exercise 3 – continued

First, move everything to the frequency domain (just as in Exercise 2).

Moreover, you have to compute two additional values, the power spectra. According to the formula (next slide) we need the P_XX and P_NN values. Use the expression

$$P_{XX}(\omega_1,\omega_2) = X(\omega_1,\omega_2) \cdot X^*(\omega_1,\omega_2)$$

in which X^* means complex conjugate (use the conj() function in MATLAB).

Ooops! We don't know the matrix X. (Well, it is impossible to know it, that's the point of the degradation.) Nevermind, let's approximate P_XX with P_YY .

Exercise 3 – continued

After this, implement the formula for the R restoration matrix:

$$R(\omega_{1}, \omega_{2}) = \frac{H^{*}(\omega_{1}, \omega_{2})}{|H(\omega_{1}, \omega_{2})|^{2} + \frac{P_{NN}(\omega_{1}, \omega_{2})}{P_{XX}(\omega_{1}, \omega_{2})}}$$

in which H^* means complex conjugate (use the conj() function in MATLAB).

Finally, apply the restoration system to the input in the frequency domain. Multiply **Y** with **R** elementwise and transform back the result to the spatial domain.

$$\tilde{x} = \left| \text{IDFT} \left\{ R \cdot \text{DFT} \left\{ y \right\} \right\} \right|$$

Execute script3.m and check your results!

Original input image



Degraded image (filter + noise)



Restored image (Wiener)



Exercise 4

Implement the function restoration_wiener_white in which:

- The function has three inputs:
 - y is the degraded image,
 - h is the point spread function of the system,
 - var_n is the variance of the noise (and we assume white noise)
- The function have to return the restored image (x_tilde).

Exercise 4 – continued

First, move everything to the frequency domain (just as in Exercise 2).

Moreover, you have to compute the noise to signal power ratio. In this, we know the variance of the noise $(var_n is given as an input parameter)$, and the variance of the signal can be approximated with the variance of the degraded image (var(y(:))).

$$NSPR = \frac{\sigma_N^2}{\sigma_X^2}$$

Exercise 4 – continued

After this, implement the formula for the **R** restoration matrix:

$$R(\omega_1, \omega_2) = \frac{H^*(\omega_1, \omega_2)}{|H(\omega_1, \omega_2)|^2 + \frac{P_{NN}(\omega_1, \omega_2)}{P_{XX}(\omega_1, \omega_2)}} = R(\omega_1, \omega_2) = \frac{H^*(\omega_1, \omega_2)}{|H(\omega_1, \omega_2)|^2 + NSPR}$$
assuming white noise

in which H^* means complex conjugate (use the conj() function in MATLAB).

Finally, apply the restoration system to the input in the frequency domain. Multiply **Y** with **R** elementwise and transform back the result to the spatial domain.

$$\tilde{x} = \left| \text{IDFT} \left\{ R \cdot \text{DFT} \left\{ y \right\} \right\} \right|$$

Execute script4.m and check your results!



Degraded image (filter + noise)



Restored image (Wiener, NSPR)



THE END