AIC

Cours TC5

Result of TP6

Students:
Xiaoxiao CHEN
Dong FEI

Supervisor:
Prof. Yohanne TENDERO

03 nov. 2016

1 Exercise: Wiener Filter

This exercise aims to compare the deconvolution simply by division and the deconvolution with Wiener Filter.

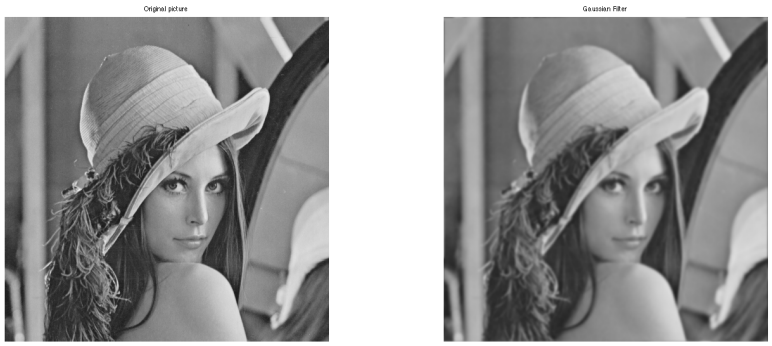
To better compare the results, i) a noise-free gaussian filter is implemented first to the picture and the simple division deconvolution is conducted, ii) a white noise is added to the gaussian filter, the simple division is conducted; iii) the Wiener Filter is implemented.

1.1 Deconvolution without noise

A gaussian filter is conducted to the picture 'Lena' as

$$J := I * G_\sigma$$

,where $\sigma = 3.5$. Here we use the gaussian filter function completed in tp4.



(a) Origin picture (b) Gaussian filter $\sigma = 3.5$

Figure 1: Convolution without noise

In this case, deconvolution is conducted with simple division by the DFT of the gaussian filter:

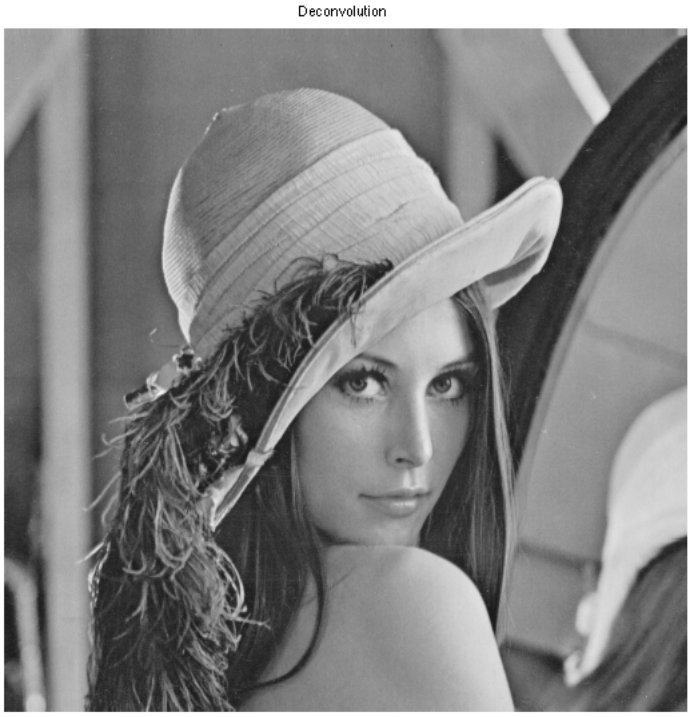


Figure 2: deconvolution without noise

As is shown in this deconvolution result, simple division works well because there is no noise:

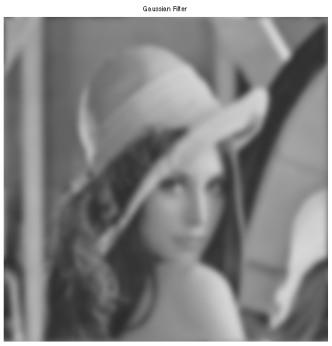
$$I(f) = \hat{H}(f) * J(f)$$

,where $\hat{H}(f) = \frac{1}{G_\sigma}$.

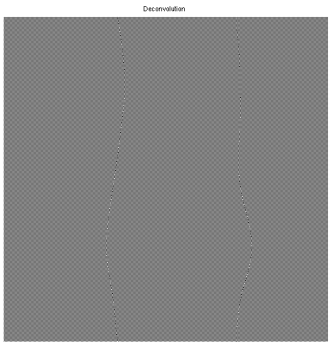
Different σ is modified to tell the difference:

(a) Gaussian filter $\sigma = 1$

(b) Deconvolution $\sigma = 1$



(c) Gaussian filter $\sigma = 10$



(d) Deconvolution $\sigma = 10$

Figure 3: Deconvolution without noise, where $\sigma = 1$, $\sigma = 10$

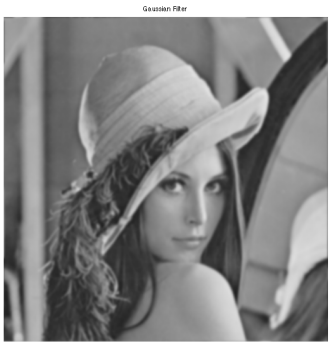
It is found that when σ is too large, for example 10, the deconvolution doesn't make sense.

1.2 Deconvolution with noise (σ fixed at 3.5)

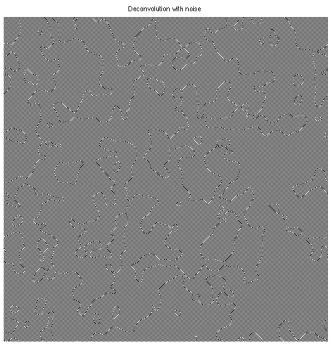
A white gaussian noise is added to the previous treatment:

$$Y := G_\sigma * I + B$$

, where B is the white gaussian noise.



(a) Gaussian filter with a white noise



(b) Deconvolution by division

Figure 4: Deconvolution with noise by division

In this case, the simple division is tested and returns a bad result. This is mainly because, when conducted with the division, the noise is multiplied with a factor $\frac{1}{|\hat{G}_\sigma(f)|}$. As for a gaussian filter with $\sigma = 3.5$, there exists several frequencies for which $|\hat{G}_\sigma(f)|$ is small. So the noise is to some extent enlarged.

1.3 Wiener Filter

As problem posed in the previous section, Wiener filter is implemented for convolution with noises.

$$\hat{H}(f) = \frac{1}{|\hat{G}_\sigma(f)|}[\frac{\hat{G}_\sigma(f)^2}{\hat{G}_\sigma(f)^2 + \frac{S_B(f)}{S_X(f)}}]$$

, where $S_B(f)$ is a constant that is the variance of the the noise. Here we use a white gaussian noise with $\sigma = 0.01$. Thus, $S_B = 10^{-4}$. In the same time $S_X(f)$ is the module of the DFT of the origin picture. To conclude, the Wiener filter $\hat{H}(f)$ is calculable.

Figure 5: deconvolution with noise using Wiener filter

Here, $\frac{1}{\hat{G}(f)}$ is the inverse of the original system, and $SNR = \frac{S_X(f)}{S_B(f)}$ is the signal-to-noise ratio. When there is zero noise (i.e. infinite signal-to-noise), the term inside the square brackets equals 1, which means that the Wiener filter is simply the inverse of the system, as we conducted previously. However, as the noise at certain frequencies increases, the signal-to-noise ratio drops, so the term inside the square brackets also drops. This means that the Wiener filter attenuates frequencies dependent on their signal-to-noise ratio.

2 Exercise: Exchange angle

In this exercise, two natural pictures are separated (in the frequency field) into modules and angles and combined one with another.

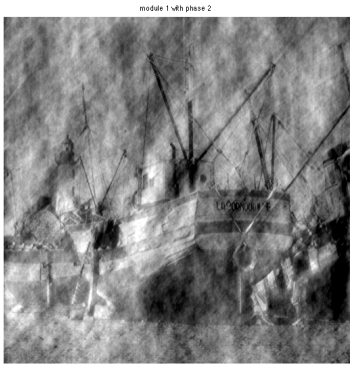
To begin with, we choose two picture as follows:

(a) Origin picture 1

(b) Origin picture 2

Figure 6: Origin picture to conduct the exercise, picture 1: a young lady and picture 2: a boat

After the exchange, we get two pictures as follows:



(a) Exchanged picture 1

(b) Exchanged picture 2

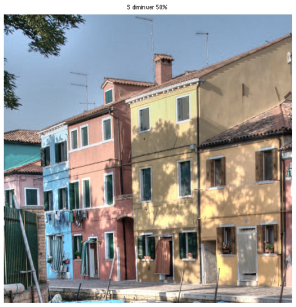
Figure 7: After the exchange, the picture(a) on the left has the module of the lady and the angle of the boat when the picture(b) on the right has the module of the boat and the angle of the lady

Without doubt, it is the angle who matters compared with the module. So angles contain more information.

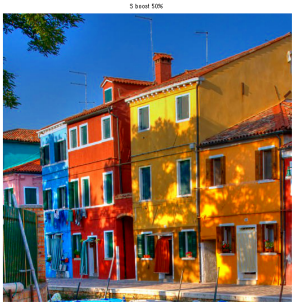
3 Exercise: Change the saturation in HSV colormap

The HSV color space (Hue, Saturation, Value) is often used by people who are selecting colors (e.g., of paints or inks) from a color wheel or palette, because it corresponds better to how people experience color than the RGB color space does. We use the functions *rgb2hsv* and *hsv2rgb* convert images between the RGB and HSV color spaces.

(a) Original picture



(b) Reduce the saturation



(c) Raise the saturation

Figure 8: Change the saturation:(a) is the original picture; (b) has the saturation reduced by 50%; (c) has the saturation raised by 50%

4 Exercise: Change the contrast

Normally in a picture in matlab, the value of a pixel $\in [0, 1]$. When it is not the case, the picture is too dark or too bright for human eyes and tend to loose information.

As a result, the contrast could be changed from the range $[min, max] \rightarrow [0, 1]$:

$$u' := \frac{u - min}{max - min}$$

(a) Origin picture (b) Changed contrast

Figure 9: After the change of the contrast, the picture(b) has the part becoming more bright (like the window) as well as the part remaining dark (like the roof).

The picture on the right has a histogram ranging more widely. And the picture after the change is more readable for human eyes.

To complete, when the previous picture has pixels 0 and 1, how can we improve the contrast? In this case, we seek for a better distribution.

For example: $\phi(x) = x^\gamma$, where $\gamma > 0$. With a correct γ, the distribution won't concentrate on a certain range any more.

5 Exercise: Median filter

5.1 Description of the algorithm

This exercise aims to implement a Median Filter and realize the function of removing noises for an image.

The median filter is a nonlinear digital filtering technique, often used to reduce noises. One reason makes Median filtering widely used in image processing is that, under certain conditions, it preserves edges while removing noise.

The main idea of the median filter is to run through the image pixel by pixel, replacing each pixel with the median of neighboring pixels. The pattern of neighbors is called the "window", which slides, pixel by pixel, over the entire image.for $u(i, j)$,we have $median = u(k, l) : k \in i - \omega, ..., i + \omega, l \in j - \omega, ..., j + \omega$.

5.2 Analysis the results

(a) Lena with noises

(b) Denoise with median filter

Figure 10: Effects of median filter

From the results, we notice that the median filter can effectively reduce the noises. Even though there are blurs on every original details, we can still successfully tell the general contours of every objects which is a huge improvements compared to the image with noises.

6 Exercise: Written exercise

Please refer to scanned version below.

Exercise écrit Tp6. Binôme Xiaoxiao Chen et Dong Fei

(1) $\hat{f}(\xi) = \int_{\mathbb{R}} f(x) e^{-ix\xi} dx$

$= \int_{\mathbb{R}} \frac{\mathbb{1}_{[0,k]}(x)}{k} e^{-ix\xi} dx$

$= \frac{1}{k} \int_0^k e^{-ix\xi} dx$

$= -\frac{1}{i\xi k} [e^{-ix\xi}]_0^k$

$= \frac{1}{i\xi k} (1 - e^{-ik\xi})$

$= \frac{1}{i\xi k} e^{-\frac{1}{2}ik\xi} (e^{\frac{1}{2}ik\xi} - e^{-\frac{1}{2}ik\xi})$

$= \frac{1}{i\xi k} e^{-\frac{1}{2}ik\xi} \cdot 2i \sin(\frac{1}{2}k\xi)$

$= e^{-\frac{i}{2}k\xi} \cdot \mathrm{sinc}(\frac{k\xi}{2\pi})$ (in the note, it's writen $\mathrm{sinC}(x) = \frac{\sin(\pi x)}{\pi x}$

(2) when a convolution is inversible. the transform of $\hat{f}(\xi)$ should not be zero $\Rightarrow$ $0 < k \leq \pi$. when $k > \pi$, the convolution is null.

~~$\frac{k\xi}{2\pi} \neq n\pi \quad n \in \mathbb{N}$~~

$\Rightarrow$ so the convolution is inversible when $0 < k \leq \pi$.

1. $obs(x) = \int_0^{\Delta t} u(x - vt)\, dt$

$= \frac{1}{v} \int_0^{\Delta t} u(x - vt)\, d(vt)$

$= \frac{1}{v} \int_0^{v\Delta t} u(x - vt) \times \mathbb{1}\, d(vt)$

we have f_k ~~take~~ give $k = v\Delta t$ $\quad f_{v\Delta t}(x) = \frac{\mathbb{1}_{[0, v\Delta t]}(x)}{v\Delta t}$

$\Rightarrow obs(x) = \frac{1}{v} \int_0^{\Delta t} u(x - vt)\, v\Delta t\, f_{v\Delta t}(vt)\, d(vt)$

$= \Delta t \cdot u * f_{v\Delta t}(x)$.

(2) when $\Delta t \uparrow$, the image is more blurring. (plus flou que l'image u).

3. i) - calculate the TFD of the picture u.

ii) - ~~calculate the TFD~~ multiply the TFD(u) with TFD(f_k)

iii) - use ifft2() the received the picture with blur.