ADVANCED

Image Processing and Stochastic Modeling

TP5: Image Restoration

Prof. SVIATOSLAV VOLOSHYNOVSKIY,
OLGA TARAN <olga.taran@unige.ch>,
MAURITS DIEPHUIS.

Stochastic Information Processing Group

May 16, 2017

UNIVERSITÉ DE GENÈVE
FACULTÉ DES SCIENCES
Département d'informatique

SIP stochastic information processing

Submission

Please archive your report and codes in "Name_Surname.zip" (replace "Name" and "Surname" with your real name), and upload to "`Assignments/TP5: Image Restoration`" on `https://chamilo.unige.ch` before Wednesday, May 31 2017, 23:59 PM. Note, **the assessment is mainly based on your report, which should include your answers to all questions and the experimental results**.

Image restoration concerns itself with restoring a degraded image using a statistical model of the degradation process. The two major components are thus the distortion model and the method required to invert the distortion process.

Model

The following model will be used to describe the degradation process:

$$g(x,y) = h(x,y) * f(x,y) + \eta(x,y) \qquad \text{Spacial domain} \tag{1}$$

$$G(u,v) = H(u,v)F(u,v) + N(u,v) \qquad \text{Frequency domain} \tag{2}$$

where $g(x,y)$ is the distorted image pixel or value, $f(x,y)$ the original, and $h(x,y)$ is the so called *distortion* function. Finally $\eta(x,y)$ models the Additive White Gaussian Noise.

Noise

If we for now assume that we only have to deal with noise, i.e.

$$g(x,y) = f(x,y) + \eta(x,y) \qquad \text{Spacial domain}$$

$$G(u,v) = F(u,v) + N(u,v) \qquad \text{Frequency domain}$$

Two important factors to consider are:

1. The spacial characteristics of the noise
2. Whether or not the noise is correlated to the image

For example, noise that which has a constant spectrum in the Fourier domain, meaning that all frequencies are present and all have equal energy, is known as *white noise*.

If we assume the noise to be uncorrelated to the image, it can be modelled with a stochastic variable whose behaviour is fully determined by its probability density function.

Degradation function

The degradation function $H(u,v)$ can broadly be ascertained in three ways:

- Observation
- Experimentation
- Mathematical Modelling

This process is also known as *blind deconvolution* as the true degradation function is almost never known.

1 Simple Inversion

If we assume we have attained $H(u,v)$ in some way, we can ascertain the restored image estimate $\hat{F}(u,v)$ from the distorted image $G(u,v)$ in the following way:

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)}$$

Substituting for $G(u,v)$ with Equation 1:

$$\hat{F}(u,v) = F(u,v) + \frac{N(u,v)}{H(u,v)}$$

Shows there are yet two more major issues that need to be resolved:

- Even if we know $H(u,v)$, the orginal image may still not be recovered because Fourier transform of the noise might be unknown.
- Even if we know $H(u,v)$ and $N(u,v)$, the recovery might fail the moment $H(u,v)$ has very small or zero values. The fraction $H(u,v)/N(u,v)$ will then be large, so large in fact that it dominates the estimate $\hat{F}(u,v)$

1.1 Exercise

In this exercise you will perform a simple restoration using a known $H(u,v)$ function without noise.

- Read in the image $f(x,y)$, `cameraman.tif`. Convert it to doubles in the 0..1 range.
- Define $h(x,y)$ as a so called *disk* blur with a radius of 4. Use Matlab `fspecial`.
- Determine $H(u,v)$ by taking the Fourier transform of $h(x,y)$. Note that you should explicitly tell Matlab `fft2` to give an output that has the same dimension as your image $f(x,y)$.
- Attain $F(u,v)$, the Fourier transform of $f(x,y)$.
- Multiply $H(u,v)$ and $F(u,v)$ to attain $G(u,v)$. Then show the blurred image $g(x,y)$ in the real domain by taking the real part of the inverse Fourier transform of $G(u,v)$.

- Attain the estimated reconstructed image $\hat{F}(u,v)$ as follows:

$$\hat{F}(u,v) = G(u,v)/H(u,v)$$

- Attain $\hat{f}(x,y)$ via the real component of the inverse Fourier transform of $\hat{F}(u,v)$ and show it. Comment on the quality of the reconstruction.

1.2 Exercise

In this exercise you will perform a simple restoration using a known $H(u,v)$ function with noise.

- Read in the image $f(x,y)$, `cameraman.tif`. Convert it to double in the 0..1 range.
- Define $h(x,y)$ as a so called *disk* blur with a radius of 4. Use Matlab `fspecial`.
- Determine $H(u,v)$, the Fourier transform of $h(x,y)$.
- Multiply $H(u,v)$ and $F(u,v)$ to attain $G(u,v)$. Use the real part of the inverse Fourier transform of $G(u,v)$ to attain $g(x,y)$.
- Add Additive White Gaussian Noise such that the PSNR value is 40dB between the original image and the noisy blurred image. See TP 1 to express the PSNR value as function of $sigma_z$, the variance of the noise. This is the new $g(x,y)$ which is both blurred and noisy.
- Show the blurred noisy image. Is 40dB a lot of noise?
- Take the fourier transform of $g(x,y)$ to attain $G(u,v)$.
- Again, attain the estimated reconstructed image $\hat{F}(u,v)$ as follows:

$$\hat{F}(u,v) = G(u,v)/H(u,v)$$

- Attain $\hat{f}(x,y)$ and show it on screen. Draw conclusions.

1.3 Exercise

To help overcome the rather bad result of the previous reconstruction we will deploy two simple strategies.

- Visualize $H(u,v)$ by taking its absolute value and using Matlab `surf`.

This shows $H(u,v)$ has quite a bit of small values. We can greatly enhance our results by removing results corresponding to these small values.

- Zero out the frequency components in the inverse filter result for which the frequency response is below a threshold. In other words, from $G(u,v)/H(u,v)$ set to zero, those elements, still in the Fourier domain, where the corresponding absolute value of $|H(u,v)|$ is below 0.1

- Show the reconstruction result in the real domain. Draw conclusions.

If all goes well, the result with the removed components is dramatically better than the previous inversion. Distortions that are still present are caused by the fact that some frequencies have now not been restored. The 'ringing' effect that is visible is called the *Gibbs phenomenon*. A slightly better but still very simple result can be attained via so called regularization using a different form of pseudo-inverse filter.

2 Regularization

Let $\mathbf{H} = H(u, v)$, then we define for some known $\mathbf{H}$:

$$\mathbf{H}_\lambda = \frac{\mathbf{H}^*}{|\mathbf{H}|^2 + \lambda^2} \tag{3}$$

where $\mathbf{H}^*$ is the complex conjugate of $\mathbf{H}$. Then:

$$\begin{aligned} \mathbf{H}_\lambda &\approx \frac{1}{\mathbf{H}} && \text{if } |\lambda| << |\mathbf{H}| \\ \mathbf{H}_\lambda &\approx 0 && \text{if } |\lambda| >> |\mathbf{H}| \end{aligned} \tag{4}$$

This is exactly like the previous pseudo inverse filter with the zeroed coefficients, but with much more smooth transitions.

2.1 Exercise

- Implement the pseudo inverse filter with regularization
- Repeat the previous exercise with the new regularized filter. Experiment with different values for λ, starting with $\lambda = 10^{-2}$.

Note

As stated in Equations (3, 4), the regularized filter $\mathbf{H}_\lambda$ behaves as the inverse of $\mathbf{H}$ when $|\lambda| << |\mathbf{H}|$. This means that in the final reconstruction step $G(u, v)/H(u, v)$ you need to multiply when using $\mathbf{H}_\lambda$.

3 The Wiener Filter

Minimum square error, or wiener, filtering is a restoration method that takes both the degradation function and the noise into account. It is founded on the basis that the images and the noise can be modelled as random process.

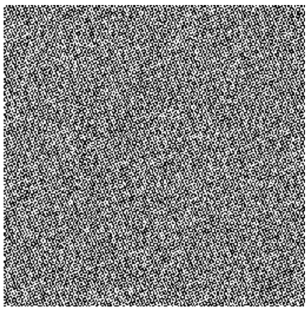
The core objective is to find an estimate $\hat{f}$ of image f such that the mean square error is minimized:

$$e^2 = E\{(f - \hat{f})^2\}$$

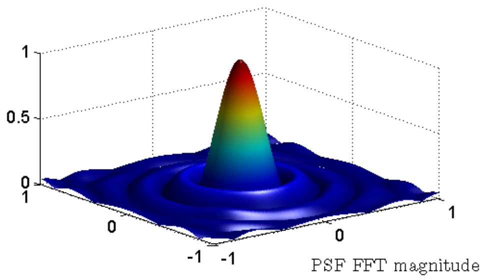
(a) Original

(b) Blurred

(c) Blurred and Gaussian noise



(d) Reconstruction using a simple inversion



(e) $|H(u,v)|$

Furthermore:

- It assumes that the noise and the image are uncorrelated
- Either the noise of the image is zero mean
- The gray levels in the estimate are a linear function of the gray levels in the distorted image

It is defined as:

$$\hat{F}(u,v) = \Big[\frac{1}{H(u,v)} \cdot \frac{|H(u,v)|^2}{|H(u,v)|^2 + S_\eta(u,v)/S_f(u,v)}\Big] G(u,v)$$

where

- $H(u,v)$ is the degradation function.
- $H(u,v)^*$ is the complex conjugate of $H(u,v)$.
- $|H(u,v)|^2 = H(u,v)H(u,v)$.
- $S_\eta(u,v) = |N(u,v)|^2$ is the power spectrum of the noise
- $S_f(u,v) = |F(u,v)|^2$ is the power spectrum of the original image

A simplification is possible when the image is corrupted by white noise and consequently the spectrum $|N(u,v)|^2$ is a constant. Further more, the power spectrum of the original image is seldom known, leading to the following approximation:

$$\hat{F}(u,v) = \Big[\frac{1}{H(u,v)} \cdot \frac{|H(u,v)|^2}{|H(u,v)|^2 + K}\Big] G(u,v)$$

where K is some constant.

3.1 Exercise

In this first exercise we will circumvent the Wiener filter requirement of needing $S_f(u,v)$, the power spectrum of the original image by leveraging two things. First, most images have roughly the same power spectrum, so we can use one from a different image. Secondly, the Wiener filter is insensitive for minor fluctuations of the power spectrum of the original image.

- Read in the image $f(x,y)$, `cameraman.tif`.
- Distorted with blurring and noise as in the previous exercise
- Restore it using Wiener filtering and using the power spectrum from the original undistorted image.
- Read in a second image, `House.tif` available on `Dokeos`.

- Restore the distorted image $f(x, y)$ a second time using Wiener filtering, but now using the power spectrum from the second image.
- Verify your results with Matlab `deconvwnr`. Report the results and give the PSNR values for both reconstructed images.

3.2 Exercise

In this exercise we will again use Wiener filtering to denoise and restore a distorted noisy image. We will deploy a model for the unknown power spectrum $S_f(u, v)$ of the un distorted image $\mathbf{f}$.

A simple model for the power spectrum $S_f(u, v)$ is formulated as follows:

$$r_x(x, y) = \sigma_{\mathbf{f}}^2 \rho^{-\sqrt{x^2+y^2}} + \mu_{\mathbf{f}} \tag{5}$$

where $\sigma_{\mathbf{f}}^2$ and $\mu_{\mathbf{f}}$ are the global variance and mean of an image $\mathbf{f}$ and ρ is the correlation coefficient between all pixels, one pixel apart.

- Implement the model using the global mean and variance from the `cameraman.tif` image.
- Again blur an image with 'disk' blurring with a radius of 4 and add Gaussian noise such that the PSNR value between the distorted and original image is 40dB.
- Deploy an Wiener filter to restore the noisy distorted image. Compare the results visually and with the PSNR against the previously attained results.

Hints

- To determine $\boldsymbol{\rho}$ efficiently, use Matlab `circshift` to shift a copy of the original image 1 pixel up, and a copy to shift all values one pixel to the left. Vectorize the shifted copies and two copies of the original image, giving 2 vectors each with dimension $2 \times NM$ where N and M are the image dimensions. Compare them using Matlab `corrcoef`.
- You will notice that Matlab `corrcoef` returns a 4×4 matrix. Take care selecting a single ρ value from that matrix.
- To efficiently evaluate Equation 5, first define a grid with all possible spacial coordinates values x and y using Matlab `ndgrid`. Note that for a 256×256 image, these coordinates should run from -127 to 128 for both axes.