

7. Image Processing - Image Restoration

Institute for Biomedical Imaging, Hamburg University of Technology

- 🎓 Lecture: Prof. Dr.-Ing. Tobias Knopp
- 🎓 Exercise: Konrad Scheffler, M.Sc.

Table of Contents

7. Image Processing - Image Restoration

- 7.1 Introduction
 - 7.1.1 Model for Degradation / Restoration
 - 7.1.2 Reconstruction
- 7.2 Noise Models
 - 7.2.1 Probability Density Functions
 - 7.2.2 Noise in Frequency Space
 - 7.2.3 Determining Noise Statistics
- 7.3 Noise Reduction
 - 7.3.1 Simple Filtering
 - 7.3.2 Adaptive Filtering
 - 7.3.3 Periodic Noise Filtering
- 7.4 Obtaining the Degradation Function
 - 7.4.1 Estimation by Experimentation
 - 7.4.2 Estimation by Observation
 - 7.4.3 Estimation by Modelling
- 7.5 Deconvolution
 - 7.5.1 Naive Deconvolution
 - 7.5.2 Wiener Deconvolution
- 7.6 Wrapup

7.1 Introduction

In the last two lectures we discussed all aspects of filtering in the spatial and the Fourier domain. We next look more general at image degradations and how they can be removed/reduced.

7.1.1 Model for Degradation / Restoration

Until now we applied image enhancement without knowledge about the imaging process. A general model of the imaging process can be formulated as:

$$g(\mathbf{r}) = h(\mathbf{r}) * f(\mathbf{r}) + \eta(\mathbf{r})$$

where

- $g(\mathbf{r})$ is the output image. It is the final result at the end of the imaging process.
- $f(\mathbf{r})$ is the original image that is unknown. It is what we aim to reconstruct.
- $h(\mathbf{r})$ is the kernel of the imaging process. Sometimes also called degradation function.
- $\eta(\mathbf{r})$ is the noise function. This is a random function which means that it will be different when imaging the same object twice.

Equivalently, in frequency space, the imaging process can be written as

$$\hat{g}(\mathbf{k}) = \hat{h}(\mathbf{k}) \hat{f}(\mathbf{k}) + \hat{\eta}(\mathbf{k})$$

7.1.2 Reconstruction

Reconstruction or restoration is the process of calculating $f(\mathbf{r})$. There are different scenarios

1. In the best case both h and a statistical model of η is known. In that case we have a linear *inverse problem* that consists of undoing the convolution. This is called *deconvolution*.
2. h is known but η is not. Then one can still perform a deconvolution but needs to guess the noise model.
3. Last case is that nothing is known. This is then a non-linear inverse problem that is much harder to solve. It is called a *blind deconvolution* and usually requires some sort of prior knowledge.

7.2 Noise Models

Any measurement, like capturing light with a digital camera, is affected by noise. Noise can be modeled by a random variable in each pixel that follows a certain statistic. The statistic is described by the probability density function (PDF), which reports the probability that a certain value is measured.

Image noise can be either be correlated or uncorrelated. Uncorrelated means that the measurement in one pixel is independent of all other pixels. We here focus on uncorrelated noise, which is the most relevant in image processing.

7.2.1 Probability Density Functions

We next discuss some important PDFs $p(z)$ occurring in practice. Two important characteristics of a random variable Z are its expected value $E(Z)$ that is defined as

$$E(Z) = \int_{-\infty}^{\infty} z p(z) dz$$

and its variance

$$\text{Var}(Z) = \int_{-\infty}^{\infty} (z - E(Z))^2 p(z) dz.$$

Instead of the variance one often instead considers the standard deviation

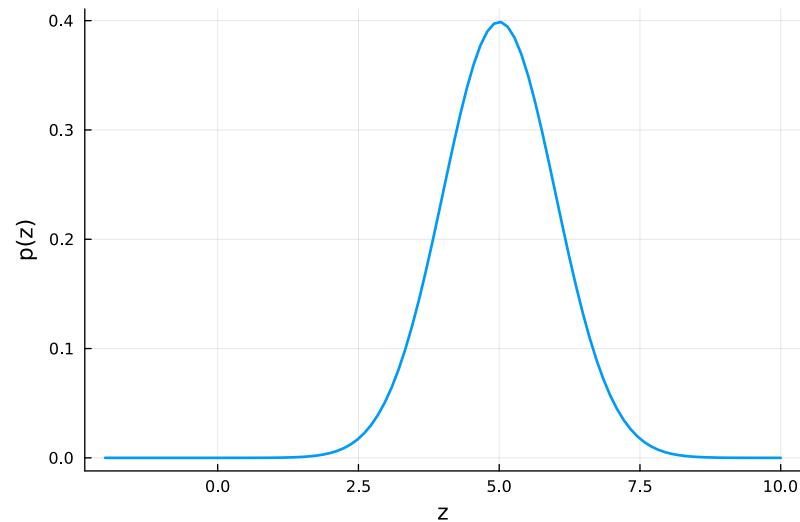
$$\text{Std}(Z) = \sqrt{\text{Var}(Z)} = \sigma.$$

Gaussian Noise

The PDF of Gaussian noise is given by

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(z-a)^2}{2\sigma^2}\right).$$

Its expected value is $E(Z) = a$ the standard deviation is $\text{Std}(Z) = \sigma$.

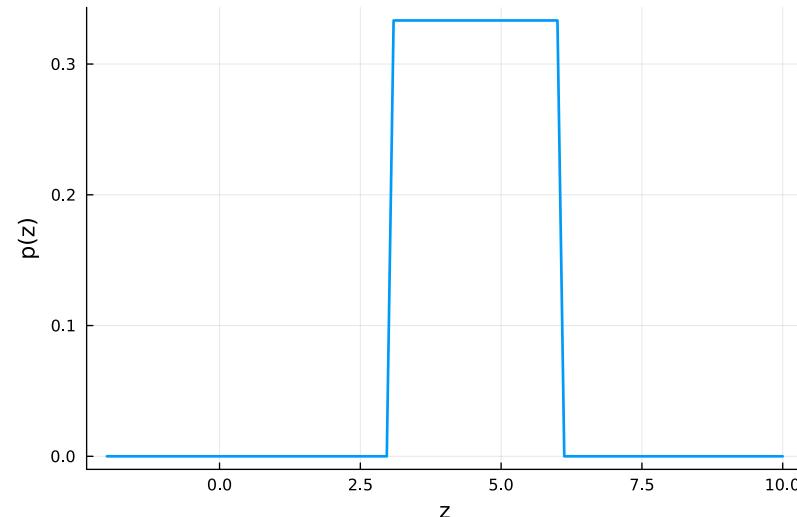


Uniform Noise

The PDF of uniform noise is given by

$$p(z) = \begin{cases} \frac{1}{b-a} & \text{if } a \leq z \leq b \\ 0 & \text{otherwise} \end{cases}$$

with the expected value $E(Z) = \frac{a+b}{2}$ and the standard deviation $\text{Std}(Z) = \sqrt{\frac{(b-a)^2}{12}}$.

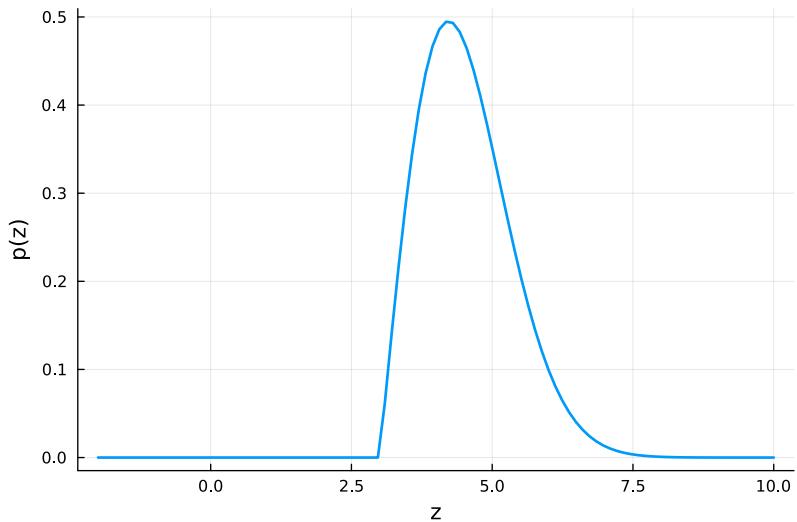


Rayleigh Noise

The PDF of Rayleigh noise is given by

$$p(z) = \begin{cases} \frac{2}{b}(z-a)\exp\left(-\frac{(z-a)^2}{b}\right) & z \geq a \\ 0 & z < a \end{cases}$$

with the expected value $E(Z) = a + \sqrt{\pi b/4}$ and the standard deviation $\text{Std}(Z) = \sqrt{\frac{b(4-\pi)}{4}}$.



It features two important characteristics that distinguish it from the Gaussian PDF:

1. It has a value a below which no noise can occur.
2. It is not symmetric and can model a bias in one direction.

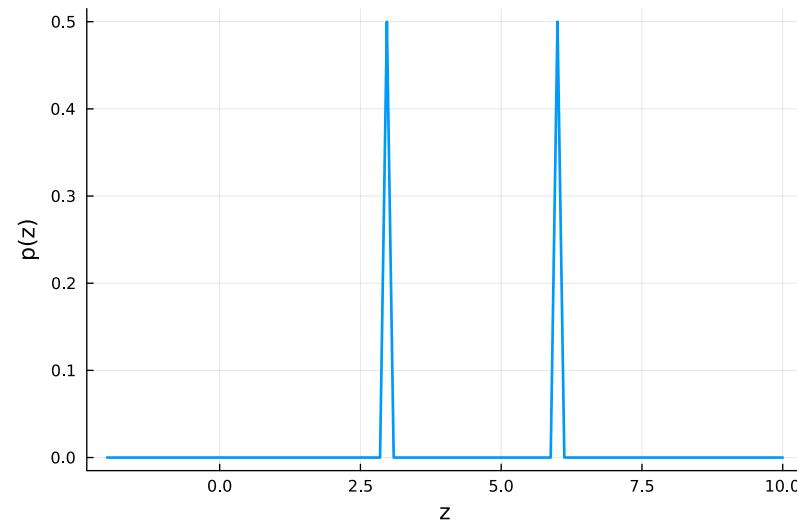
An example of such noise is a detector that counts events. It cannot go below zero, since the number of events is always positive. Furthermore the noise is higher for a low number of events, while it is lower for a high number of events.

Salt and Pepper Noise

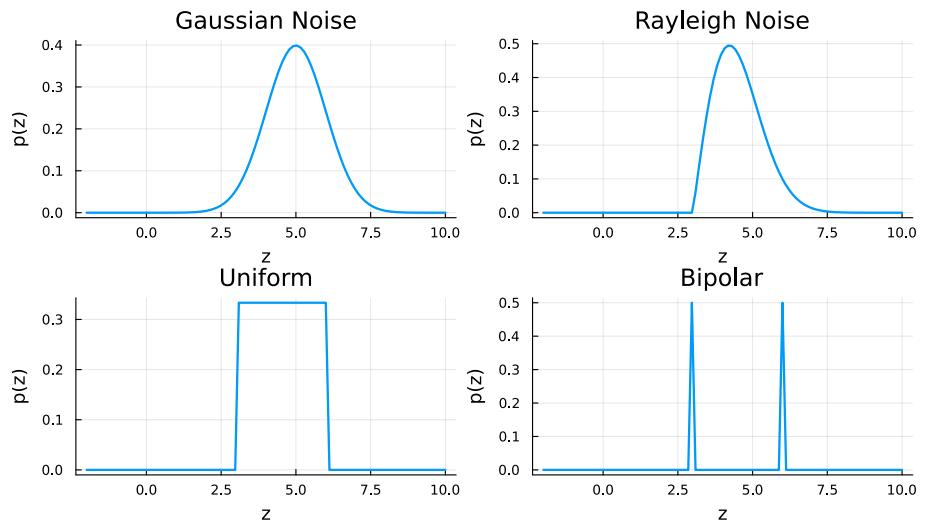
Salt and pepper noise, also named bipolar noise is not an additive noise since it includes saturation effects. In particular it may happen within the electronics during digitization where a pixel is either mapped to the highest or the lowest value (0 or $2^k - 1$ for a k bit digitizer). This also explains why the noise is not additive since you can change the value of the underlying image but the noise pixel will still be mapped to 0 or $2^k - 1$.

Keeping that in mind we can still write the PDF in the form

$$p(z) = \begin{cases} P_a & \text{for } z = a \\ P_b & \text{for } z = b \\ 0 & \text{otherwise.} \end{cases}$$

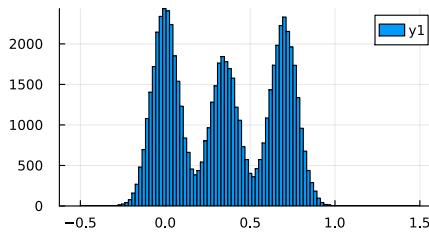
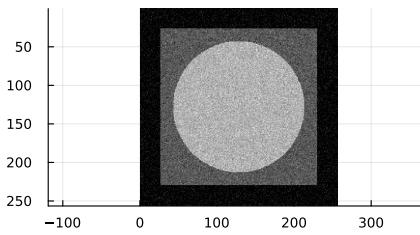


The following illustrates all four PDFs in a single figure for comparison:

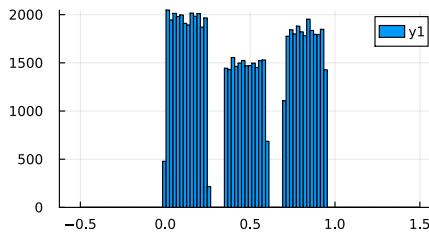
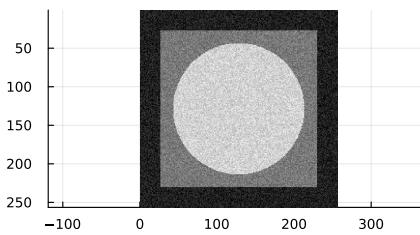


Let us next look at some images effected by these noise models and look at the histograms:

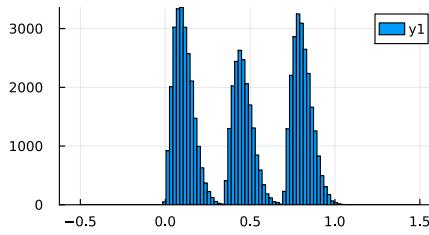
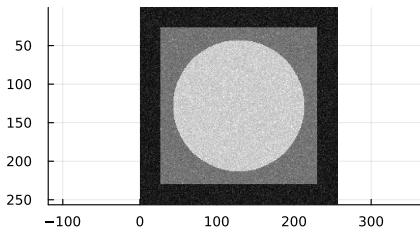
Gaussian



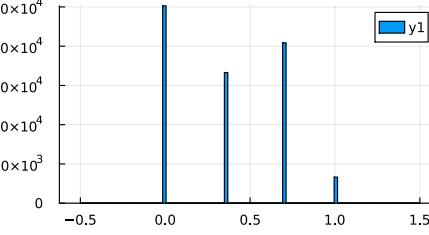
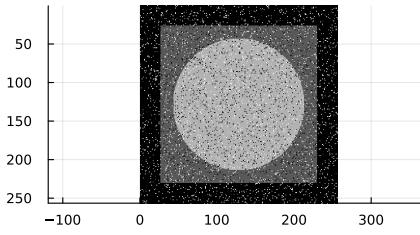
Uniform



Rayleigh



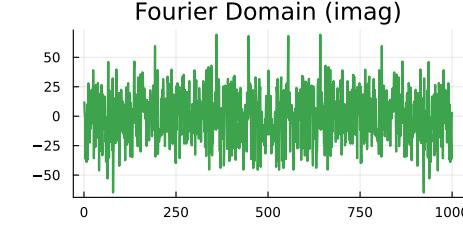
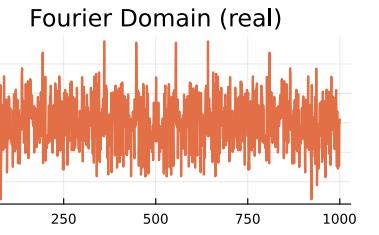
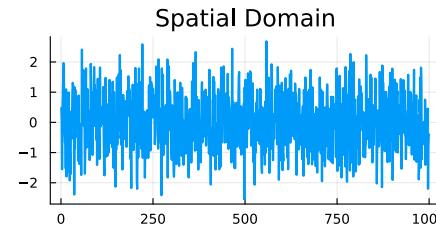
Salt and Pepper



7.2.2 Noise in Frequency Space

One interesting question is how the noise translates into the Fourier space. In case of Gaussian uncorrelated noise, the noise carries over along the entire spectral line. In other words: Noise contains all frequencies with the same likelihood. We call this white noise motivated by the fact, that light usually is of this form.

Let us verify this using a simple code example:



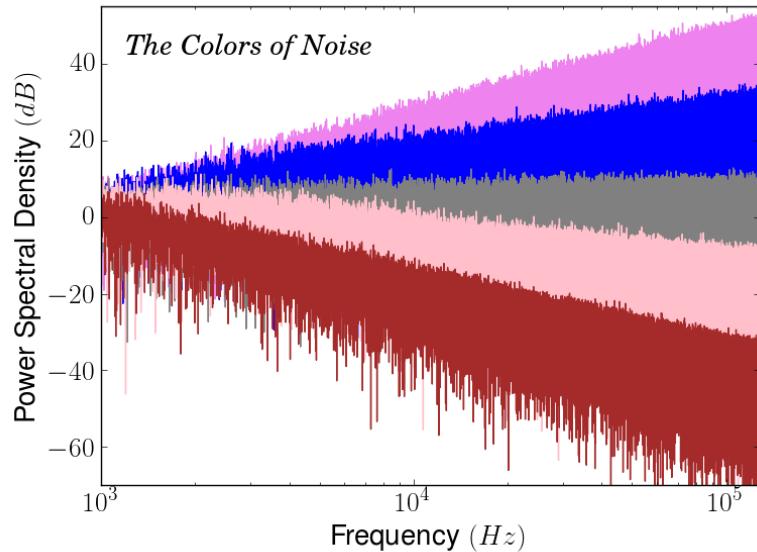
The lower right plot shows the quantity $|\hat{f}(\mathbf{u})|$, which we call the *power spectrum* of \hat{f} . It is usually considered to quantify the amount of noise in frequency space.

Colored Noise

Noise is not always white but can have different color. Different color here means that the power spectrum varies depending on frequency as is shown in the following image.

Note

In practice it depends a lot on the application what type of noise one gets. The more you know about the noise distribution, the better you can handle it. In most cases a Gaussian distribution arises. It is thus quite common to assume Gaussian noise even if one has no precise knowledge about the concrete noise distribution.



Such noise appears in electronical applications because of the nature of certain circuits. [Thermal noise](#), that occurs in all resistors is white (Gaussian) noise.

7.2.3 Determining Noise Statistics

There are different ways to determine the noise statistics if it is not known from a physical model.

With Access to Background Images

Let us have a look again at the imaging equation:

$$g(\mathbf{r}) = h(\mathbf{r}) * f(\mathbf{r}) + \eta(\mathbf{r})$$

Quite often one has the possibility to capture an *empty* or *background* image

$$g_{\text{empty}}(\mathbf{r}) = h(\mathbf{r}) * 0(\mathbf{r}) + \eta(\mathbf{r}) = \eta(\mathbf{r})$$

where $0(\mathbf{r})$ is the function that maps all spatial positions to zero. Having this one can perform a statistical analysis if(!) the mean and variance is constant over the entire image. If that is not the case one can capture a series of images $g_{\text{empty}}^l(\mathbf{r})$ and perform the statistical analysis along the samples l for each pixel \mathbf{r} independently.

Without Access to Background Images

Quite often one just has the image g and no additional background image. In that case one can still estimate the noise statistics:

- In the spatial domain one can seek for a region $R_{\mathbf{r}}$, which appears to be (almost) constant. Then, within this constant area one can derive the noise variance. The mean of the background can be determined only if the selected area contains background only.
- In case of a smooth but not constant foreground in the region $R_{\mathbf{r}}$ it is possible to derive a linear or polynomial model of the signal using [regression methods](#). Let this model be y . Then the noise variance can be estimated from $g(R_{\mathbf{r}}) - y(R_{\mathbf{r}})$.
- Finally, it is often easy to separate noise from the image entirely in Fourier space. To this end, one performs the statistical analysis in an outer region of the Fourier space. This is based on the fact that natural images usually decay rapidly in Fourier space while the noise level is constant.

Statistical Analysis

The [mean](#) can be calculated from sampled values within the region $R_{\mathbf{r}}$ by

$$\mathbb{E}(f(R_{\mathbf{r}})) = \frac{1}{|R_{\mathbf{r}}|} \sum_{\mathbf{r}' \in R_{\mathbf{r}}} f(\mathbf{r}').$$

Here, the region $R_{\mathbf{r}}$ can also be the entire image.

The [variance](#) can be calculated by

$$\text{Var}(f(R_{\mathbf{r}})) = \frac{1}{|R_{\mathbf{r}}|} \sum_{\mathbf{r}' \in R_{\mathbf{r}}} (f(\mathbf{r}') - \mathbb{E}(f(R_{\mathbf{r}})))^2.$$

7.3 Noise Reduction

We first consider the case that the imaging system is not affected by the degradation function h , i.e. we assume that $h = \delta$ and in turn the imaging equations in spatial and Fourier domain can be expressed as

$$g(\mathbf{r}) = f(\mathbf{r}) + \eta(\mathbf{r})$$

$$\hat{g}(\mathbf{k}) = \hat{f}(\mathbf{k}) + \hat{\eta}(\mathbf{k})$$

The first idea could be to subtract $\eta(\mathbf{r})$ from $g(\mathbf{r})$ but this of course does not work since $\eta(\mathbf{r})$ is a random variable and in turn we can only know the mean and variance from it but not the concrete instance that affected $g(\mathbf{r})$.

7.3.1 Simple Filtering

In the case that one has a series of measurements

$$g_l(\mathbf{r}) = f(\mathbf{r}) + \eta_l(\mathbf{r}) \quad l = 1, \dots, L$$

one can simply average over the elements of the series. This is often the case when \mathbf{g} is a temporal function and one can *block average* over time:

$$\bar{g}(\mathbf{r}) = \frac{1}{L} \sum_{l=1}^L g_l(\mathbf{r})$$

Of course this requires that the image \mathbf{f} is static for $l = 1, \dots, L$.

Note

What we described in a discrete setting here is the same as the integration time or in photography the exposure time. The longer one exposures, the more light reaches the detector and in turn the noise is reduced. If the image moves, however, one gets motion artifacts.

In case that only a single image is available one cannot average over a series and instead needs to average in space. This can be done using the filtering techniques we discussed in the last two lectures. Recall that applying a Gaussian filter is a good way when the noise has a Gaussian noise distribution (or uniform) while salt and pepper noise is better handled with a median filter.

Note

The kernel width of the averaging filter needs to be adapted to the noise level σ in the image.

7.3.2 Adaptive Filtering

One drawback of the regular spatial filtering is that the filter kernel is spatially independent. This leads to the situation that regions with strong image intensity are treated the same way as regions with low image intensity although the signal-to-noise ratio is higher in the first case.

To address this issue *adaptive filters* can be used. These filters have the ability to calculate local noise statistics and then change the amount of averaging based on that. These filters are obviously not *shift-invariant* anymore and we loose the connection to the Fourier space. In addition adaptive filters are often *non-linear*.

Adaptive Local Noise Reduction Filter

There are many adaptive filters, here we present a local noise reduction filter that is based on the box averaging filter. Of course it can be generalized to other noise reduction filters.

Two ideas to make the filter adaptive are

1. Change the filter kernel size based on the local noise.
2. Use a convex combination of filtered and unfiltered image based on the local noise.

We chose the second strategy here. Our filter is a *sliding window* filter that operates on a certain region $R_{\mathbf{r}}$ at a time. What we now do is to relate the standard deviation of the local patch σ_L to the standard deviation of the entire image σ_G . The algorithm needs the later as an input parameter, which can be estimated with the methods discussed earlier.

The algorithm then distinguishes the following cases:

- if $\sigma_G \approx \sigma_L$ global and local noise are similar and we can average without problems since the underlying image is within a smooth region.
- if $\sigma_G \ll \sigma_L$ the underlying image \mathbf{f} has large variations e.g. due to an edge. In those cases we want do use the local value $g(\mathbf{r})$ without averaging.

In this way we achieve that edges are preserved, while surface areas are averaged. The updated value is then calculated by

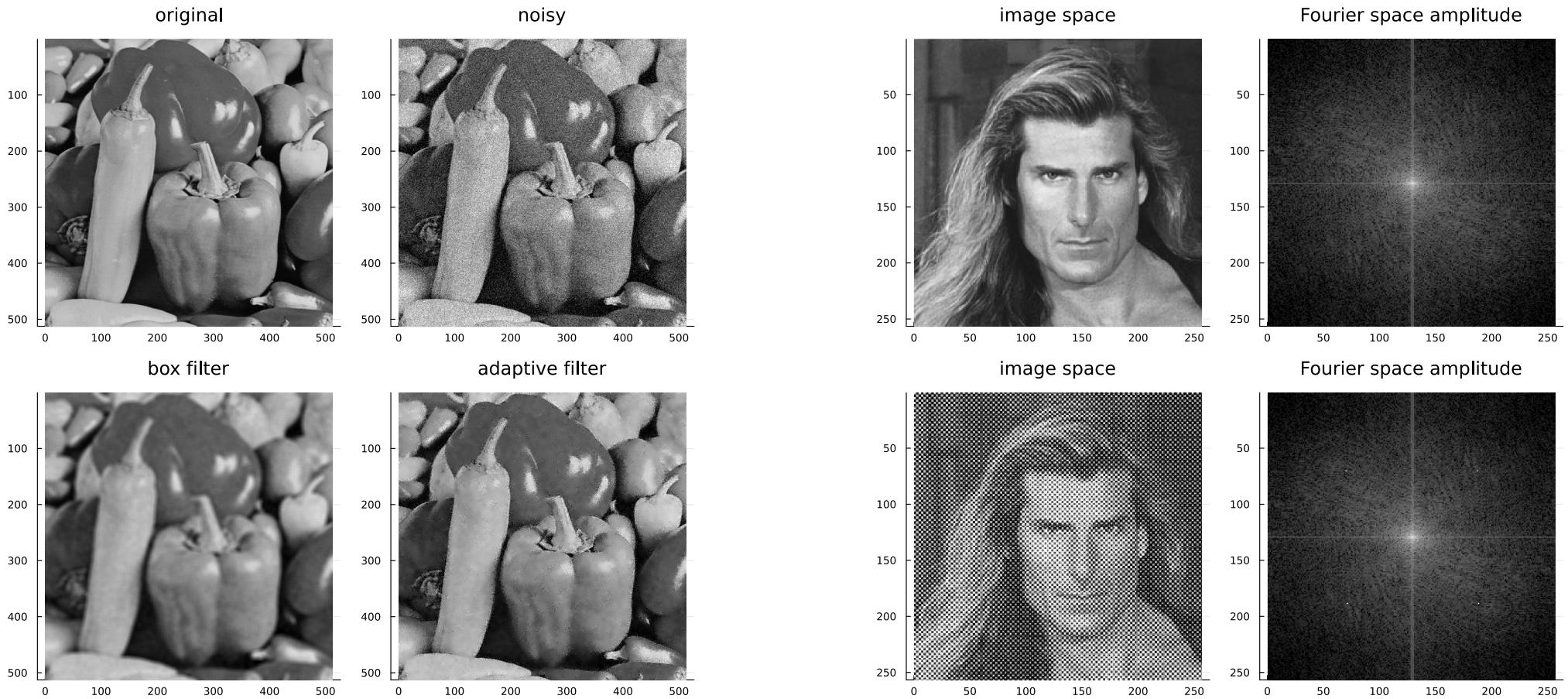
$$g_{\text{filtered}}(\mathbf{r}) = g(\mathbf{r}) - \min \left(\frac{\sigma_G^2}{\sigma_L^2}, 1 \right) (g(\mathbf{r}) - E(g(R_{\mathbf{r}})))$$

One can see that we added a special handling for the case $\sigma_L \ll \sigma_G$ since otherwise, the filtered value would return non-valid results. This operation makes the filter non-linear.

The following shows the adaptive filter in action:

Note

Adaptive filters share some ideas with filtering methods that employ neural networks.



One can clearly see that the image is less blurred although the noise is reduced the same amount as for the box filter.

7.3.3 Periodic Noise Filtering

We next focus on non-white, spatially dependent noise, which has a local support in the Fourier space. Let us have a look at the following distorted image:

One can clearly see that there is an overlaying periodic pattern that degrades the image quality. Such a pattern is very common if one, for instance, digitizes some printed raster image.

In Fourier space one can clearly identify the peaks containing the noise component.

Band-Stop Filter

In order to remove such periodic noise we can apply a so called *band-stop* or *notch filter*. Basically one can use a high pass filter and move the center to a different frequency. For instance a Gaussian band-stop filter can be written as

$$\hat{h}_{\text{BS}}^{\sigma, u_0, v_0}(u, v) = 1 - \exp\left(-\frac{(u - u_0)^2 + (v - v_0)^2}{2\sigma^2}\right)$$

Periodic noise always occurs not only at frequency (u_0, v_0) but also at $(-u_0, -v_0)$ since the image f is real and therefore we have the symmetry

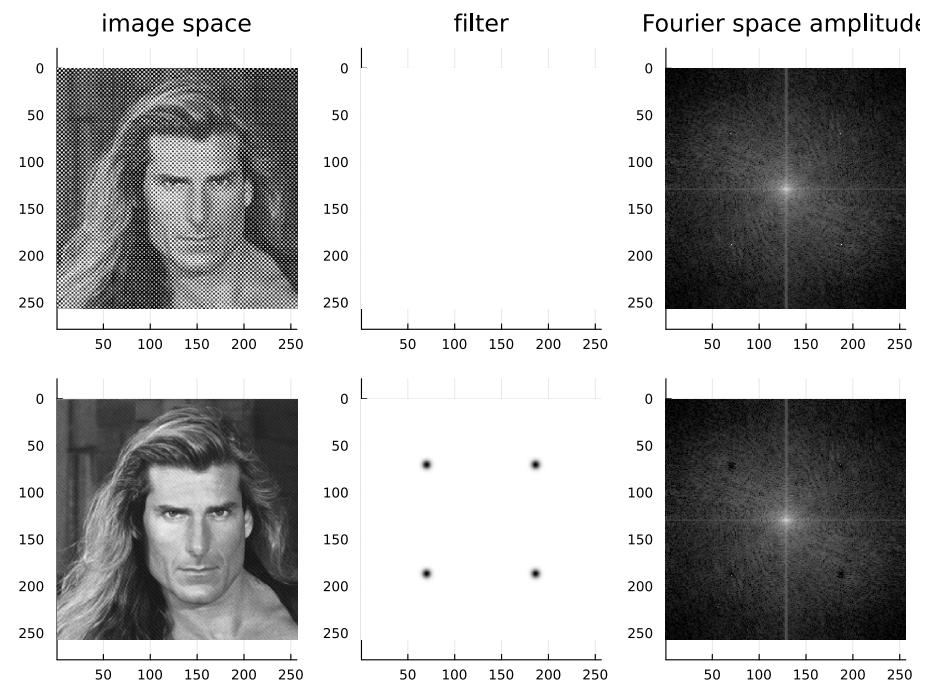
$$\hat{f}(u_0, v_0) = \overline{\hat{f}(-u_0, -v_0)}.$$

But one usually also see the signal at $(u_0, -v_0)$ and $(-u_0, v_0)$. Consequently, the applied band-stop should filter out the signal in all four quadrants:

$$\begin{aligned} \hat{h}_{\text{BS, symmetric}}^{\sigma, u_0, v_0}(u, v) &= 1 - \exp\left(-\frac{(u - u_0)^2 + (v - v_0)^2}{2\sigma^2}\right) - \exp\left(-\frac{(u + u_0)^2 + (v + v_0)^2}{2\sigma^2}\right) \\ &\quad - \exp\left(-\frac{(u - u_0)^2 + (v + v_0)^2}{2\sigma^2}\right) - \exp\left(-\frac{(u + u_0)^2 + (v - v_0)^2}{2\sigma^2}\right) \end{aligned}$$

Our example before was distorted by two periodic frequencies and in turn we need to combine four band-stop filters.

The following shows the developed band-stop filter in action:



Note

Periodic degradations often are the result of interference, which sometimes cannot be avoided. The Moire effect is an example of an interference resulting in periodic patterns that can be removed by band-stop filtering.

Note

We have not discussed how to determine the frequencies of the periodic noise. In case that they are clearly apart from the center of the Fourier space once usually can determine them by thresholding of the power spectrum.

7.4 Obtaining the Degradation Function

We finally switch to the general setting

$$g(\mathbf{r}) = h(\mathbf{r}) * f(\mathbf{r}) + \eta(\mathbf{r}).$$

First we discuss how to determine the kernel h . Depending on the application there are different approaches here:

7.4.1 Estimation by Experimentation

In case that we have access to the underlying experiment we can perform the imaging with an $f(\mathbf{r})$ that we know and then determine \mathbf{h} by experimentation. If we, for instance insert a dirac delta into the imaging equation we obtain

$$g(\mathbf{r}) = \mathbf{h}(\mathbf{r}) * \delta(\mathbf{r}) + \eta(\mathbf{r}) = \mathbf{h}(\mathbf{r}) + \eta(\mathbf{r})$$

Since a real delta distribution cannot be realized in practice one should divide this equation by the area of the applied sample. The noise component $\eta(\mathbf{r})$ can often be ignored since the delta sample has a high intensity and is much larger than η . Furthermore, since we have access to the experiment itself we can also measure a series of images and apply averaging. Finally, if the kernel is still noisy we can also smooth the kernel in spatial domain.

7.4.2 Estimation by Observation

The second method is similar but has no access to the experiment itself. Instead only a distorted image \mathbf{g} is available. In that case one can look within the image for local structures like an edge, where the true function $f(\mathbf{r})$ can be estimated by semantic knowledge about the data. Then lets focus on the local region $R_{\mathbf{r}}$ and lets assume that we derive a local *unblurred* image $f_{\text{unblurred}}$ just by the knowledge that there needs to be an edge or a dot. Then we can estimate \mathbf{h} in Fourier space by

$$\hat{\mathbf{h}}(\mathbf{k}) = \frac{\hat{g}(\mathbf{k})}{\hat{f}_{\text{unblurred}}(\mathbf{k})}$$

where all quantities are restricted to the local region $R_{\mathbf{r}}$.

Note

We only roughly sketched this method. To end up at a stable method one needs to postprocess the kernel and apply further prior knowledge. For instance isotropy of the kernel can be assumed when only an edge in a single direction is available.

7.4.3 Estimation by Modelling

The third method does not try to derive the kernel in a data-driven fashion but instead uses a model. For a large class of applications (i.e. photography), the kernel will have the form of a Gaussian kernel and we can use the model

$$\hat{h}_{\text{Gaussian}}^{\sigma}(u, v) = \exp\left(-\frac{u^2 + v^2}{2\sigma^2}\right)$$

for the kernel. Now the only unknown parameter is the kernel width σ . This can then be estimated by performing several reconstructions (see section about deconvolution) and inspecting the resulting image quality.

7.5 Deconvolution

Now we assume that we know the kernel \mathbf{h} and aim to solve

$$g(\mathbf{r}) = \mathbf{h}(\mathbf{r}) * f(\mathbf{r}) + \eta(\mathbf{r})$$

or equivalently

$$\hat{g}(\mathbf{k}) = \hat{\mathbf{h}}(\mathbf{k}) \hat{f}(\mathbf{k}) + \hat{\eta}(\mathbf{k})$$

What we can directly see is that the solution is much easier in Fourier space since a scalar multiplication can be easier undone than a convolution.

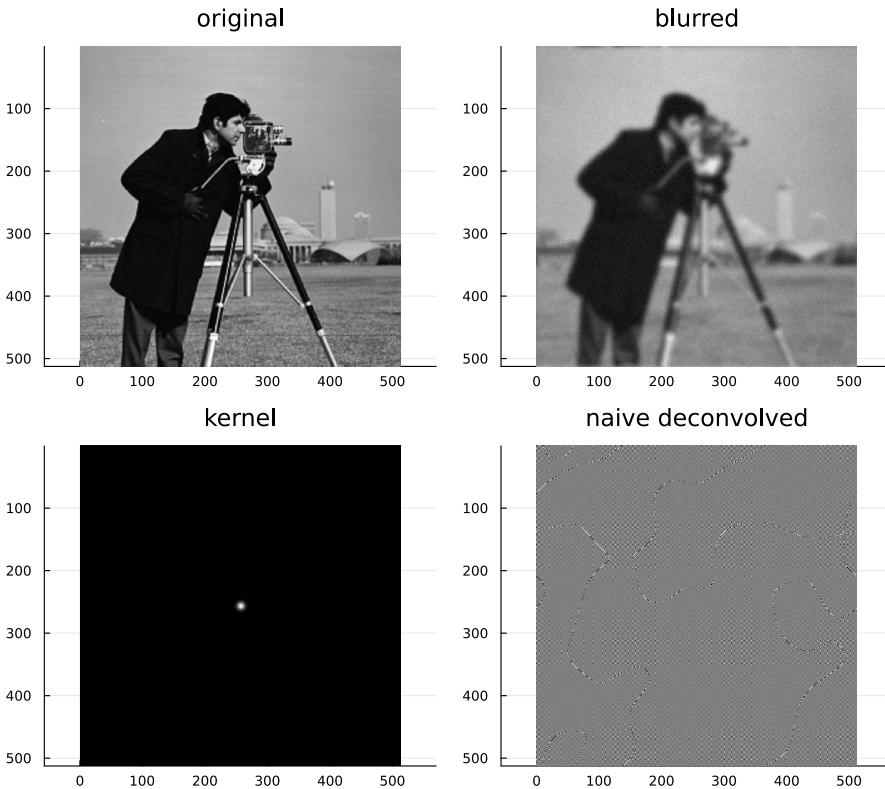
7.5.1 Naive Deconvolution

If we ignore the noise we can perform the deconvolution by

$$\hat{f}_{\text{deconv}}(\mathbf{k}) = \frac{\hat{g}(\mathbf{k})}{\hat{\mathbf{h}}(\mathbf{k})}$$

This has of course the issue that we might divide by zero but let us for the moment assume that this does not happen since $\hat{\mathbf{h}}(\mathbf{k})$ is non-zero.

The following example shows what happens when using this approach:



Observation

- The deconvolved image looks completely wrong. Why?

The reason is that we need to take the noise into account. If we do this, the naive convolution looks like this:

$$\hat{f}_{\text{deconv}}(\mathbf{k}) = \frac{\hat{g}(\mathbf{k})}{\hat{h}(\mathbf{k})} = \frac{\hat{h}(\mathbf{k})\hat{f}(\mathbf{k})}{\hat{h}(\mathbf{k})} + \frac{\hat{\eta}(\mathbf{k})}{\hat{h}(\mathbf{k})} = \hat{f}(\mathbf{k}) + \frac{\hat{\eta}(\mathbf{k})}{\hat{h}(\mathbf{k})}$$

Thus, the original image $\hat{f}(\mathbf{k})$ is overlaid by the noise component $\frac{\hat{\eta}(\mathbf{k})}{\hat{h}(\mathbf{k})}$. Now the bad thing is that we divide the noise by our kernel function $\hat{h}(\mathbf{k})$, which has very small values in outer Fourier space regions. Thus the noise is *amplified* by the deconvolution. This amplification can be so strong that the original image is not visible anymore.

7.5.2 Wiener Deconvolution

To solve the issue with the noise amplification we need to apply a technique called **regularization**. Regularization means that we still try to solve the underlying inverse problem but do so without amplifying the noise. Let's first write the naive deconvolution formula as

$$\hat{f}_{\text{deconv}}(\mathbf{k}) = \hat{g}(\mathbf{k})\hat{w}(\mathbf{k})$$

with $\hat{w}(\mathbf{k}) = \frac{1}{\hat{h}(\mathbf{k})}$. The naive deconvolution is thus just a filtering operation with an inverse transfer function.

What we now want to do is to prevent the filter kernel to *blow up*. This happens if

- the term $\hat{h}(\mathbf{k})$ gets very small and
- the term $\hat{g}(\mathbf{k})$ is dominated by noise

Based on these two properties one can derive the Wiener filter

$$\hat{w}_{\text{wiener}}(\mathbf{k}) = \frac{1}{\hat{h}(\mathbf{k})} \left(\frac{|\hat{h}(\mathbf{k})|^2}{|\hat{h}(\mathbf{k})|^2 + \frac{|\hat{\eta}(\mathbf{k})|}{|\hat{f}(\mathbf{k})|}} \right) = \frac{1}{\hat{h}(\mathbf{k})} \left(\frac{|\hat{h}(\mathbf{k})|^2}{|\hat{h}(\mathbf{k})|^2 + \frac{1}{\text{SNR}(\mathbf{k})}} \right)$$

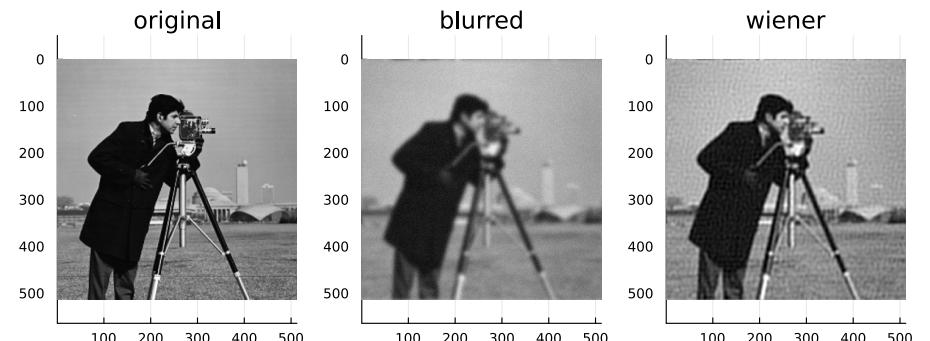
with the signal-to-noise ratio $\text{SNR}(\mathbf{k}) = \frac{|\hat{f}(\mathbf{k})|}{|\hat{\eta}(\mathbf{k})|}$.

Interpretation

The interpretation is as follows:

- If the SNR is high, the term in brackets is close to 1 and the filter acts as a regular deconvolution kernel.
- If the SNR is small, the term $\frac{1}{\text{SNR}(\mathbf{k})}$ becomes dominant and in turn the entire bracket approaches 0 if \hat{h} gets small. This means that the noise amplification is prevented.

Let us apply Wiener deconvolution to our example before:



One can see that the deconvolved image now indeed looks similar to the original one. It is not perfect but the spatial resolution is clearly improved while only the noise is only slightly increased.

Unknown Parameters

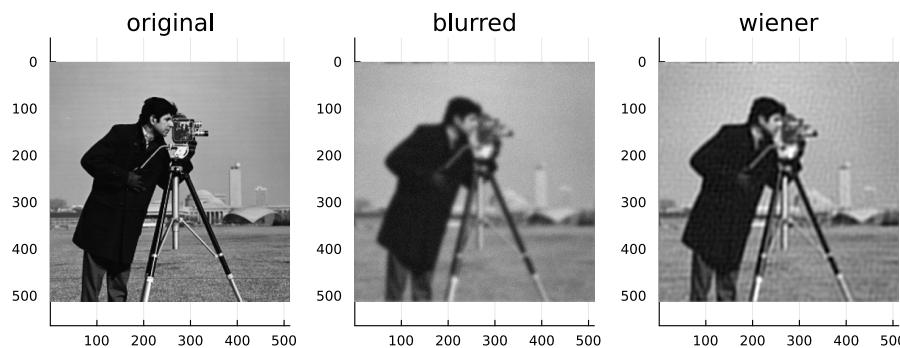
The Wiener filter requires the noise image $|\hat{\eta}(\mathbf{k})|$ and the power spectrum of the undegraded image $|\hat{f}(\mathbf{k})|$ as the input. The noise image can usually be replaced by a constant K in case that the noise is white. K can be estimated from noise statistics (discussed before) or chosen to be a user defined parameter. $|\hat{f}(\mathbf{k})|$ is usually approximated by $|\hat{g}(\mathbf{k})|$. Combining this we can write the Wiener deconvolution filter as

$$\hat{w}_{\text{wiener}}^{\text{impl 1}}(\mathbf{k}) = \frac{1}{\hat{h}(\mathbf{k})} \left(\frac{|\hat{h}(\mathbf{k})|^2}{|\hat{h}(\mathbf{k})|^2 + \frac{K}{|\hat{g}(\mathbf{k})|}} \right).$$

Instead one can also choose a constant term for the SNR ending up with

$$\hat{w}_{\text{wiener}}^{\text{impl 2}}(\mathbf{k}) = \frac{1}{\hat{h}(\mathbf{k})} \left(\frac{|\hat{h}(\mathbf{k})|^2}{|\hat{h}(\mathbf{k})|^2 + \lambda} \right).$$

The following slider allows to adjust K and look at the resulting deconvolved image:



7.6 Wrapup

In this lecture we have learned that images are degraded by a kernel \mathbf{h} and noise $\boldsymbol{\eta}$. The more we know about both terms, the better we can reconstruct the original image. Since both terms are commonly unknown, they need to be estimated in some way. Prior to image reconstruction one thus first needs to *analyse* the image in detail and based on that chose an adequate strategy for image improvement.

Note

Wiener filtering is related to [Tikhonov regularization](#). Tikhonov regularization is equivalent to Wiener filtering if the imaging operator is linear and shift invariant.