Image Restoration

by Martino Ferrari

1 Simple Inversion

Exercise 1.1

In this exercise we will first blur out the image *Cameraman* using a circular blur filter of radius 4 and then we will restore it using the pseudo inverse of the filter.

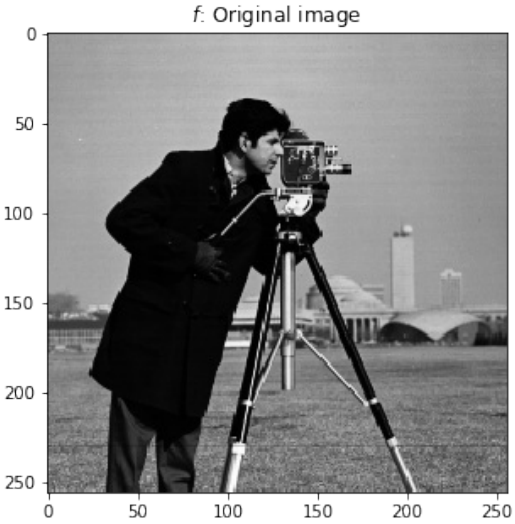

Figure 1. *Cameraman* original image

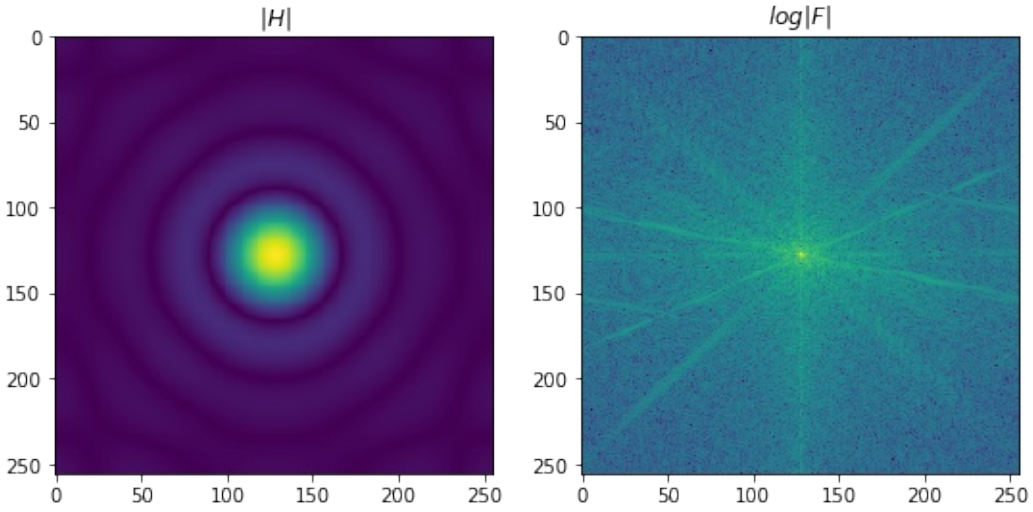

Figure 2. Filter and Image in frequency domain

The blurred image is created:

$$G = H \cdot F$$

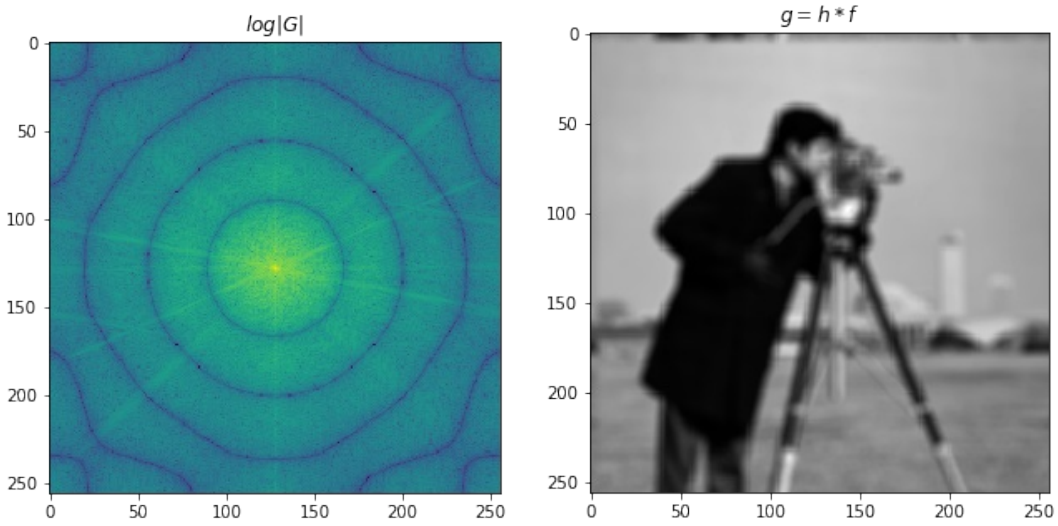

Figure 3. Resulting image in both spatial and frequency domain

By inverting H is possible to reconstruct the original image:

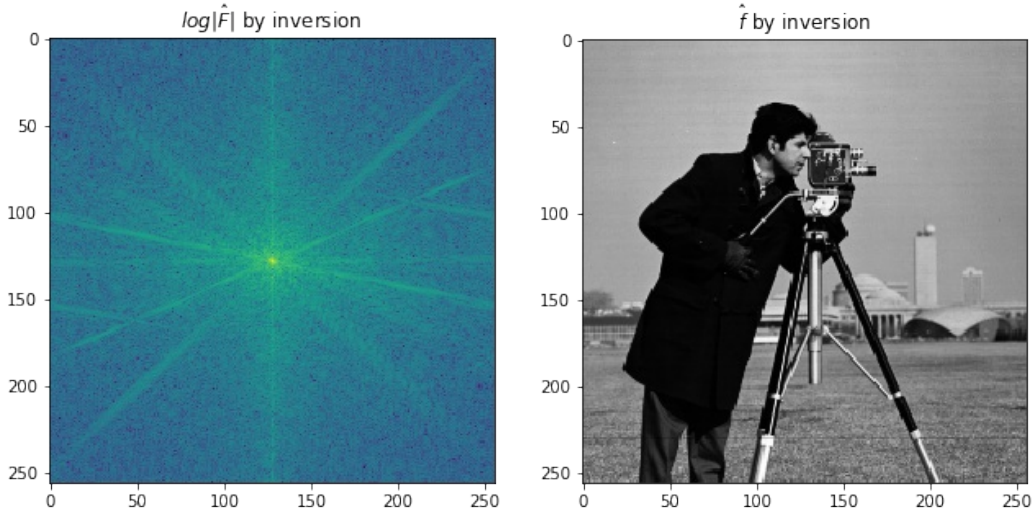

Figure 4. Reconstructed image in both spatial and frequency domain

```
mse: 5.39e-33
```

Using the inversion in this simple case where the f image is only modified by the filter h, $g = h * f$, is very effective and the reconstruction is perfect.

Exercise 1.2

In this exercise however we will add some small Gaussian noise z to g, the resulting PSNR is 40dB, meaning that the noise is really small compared to the original image ($\sigma^2_{noise} = 0.01$) and it is not visible (at least for me).

```
PSNR: 40.01dB
```

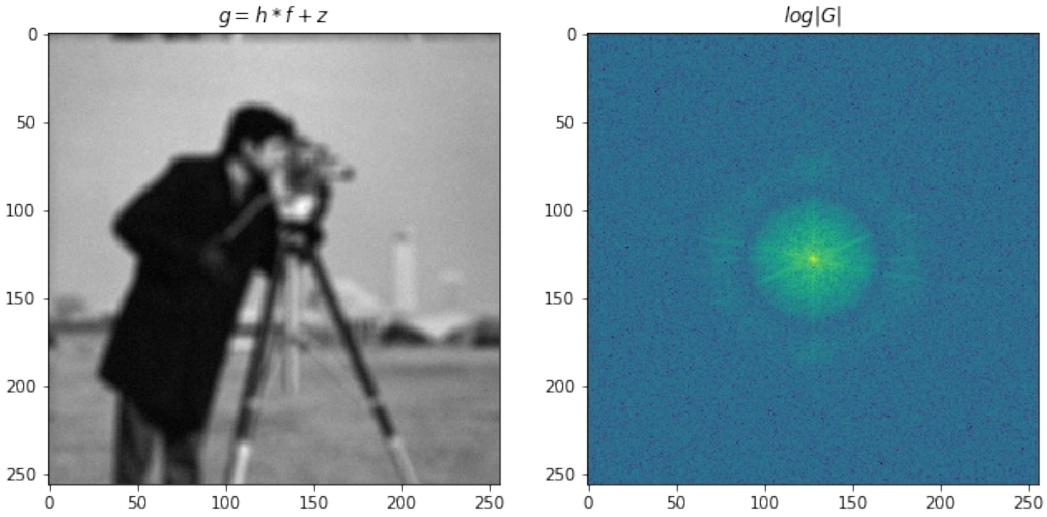

Figure 5. Blurred and noisy image

Reconstructing the image by inversion:

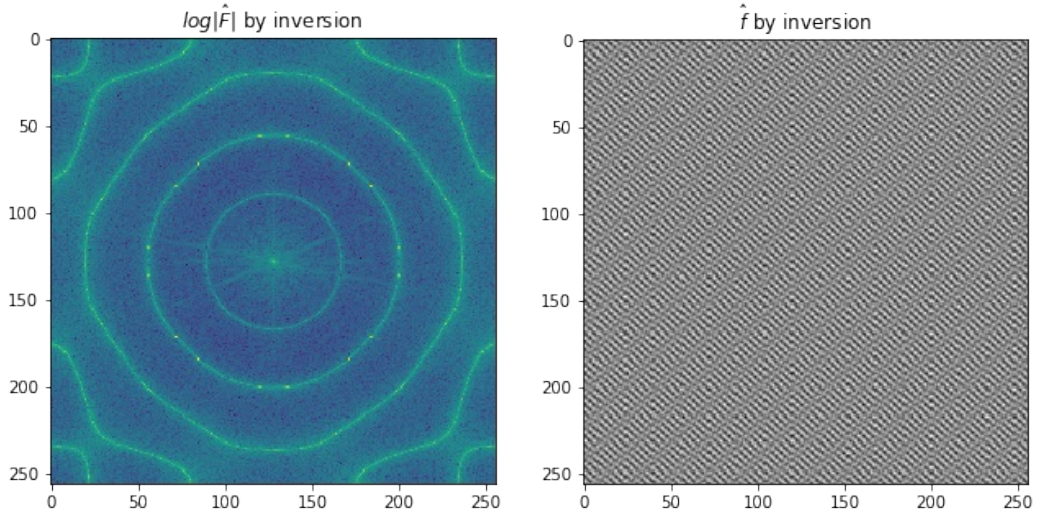

Figure 6. Reconstruction in frequency domain

The result of the reconstruction is totally wrong, even with such small noise. This is due the fact the H has many singular point (where its value is very close or equal to 0), as the noise is equally distributed in the frequency domain when reconstructing the image the noise present in the singular

point of the filter is amplified of many order of magnitudes. This amplification is very well visible in the plot of $log(|\hat{F}|)$.

Exercise 1.3

To avoid the noise explosion a simple solution is to threshold the filter H to bigger values:

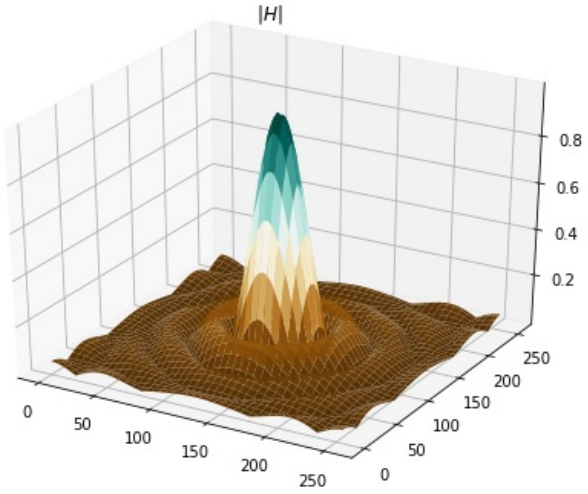

Figure 7. $|H|$ in 3D

To begin I start to tresholding all values of H smaller than $\tau = 0.1$:

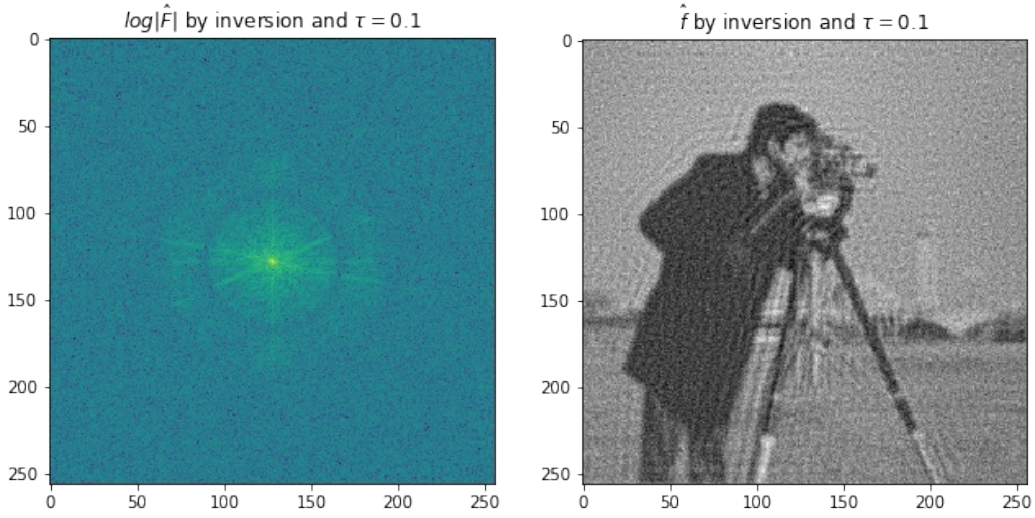

Figure 8. Reconstruction using treshold

`mse: 1.45e-02`

As it possible to see the reconstruction is already much better than the noise pattern of before, however I tried different configuration of the threshold τ to see what was the best one:

```
tau optimal: 0.27 - 8.9e-03
```

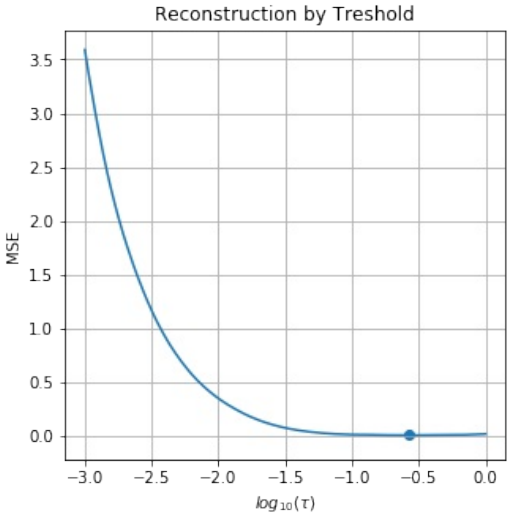

Figure 9. *mse* depending on τ

The optimal threshold seems to be $\tau = 0.27$ with an *mse* of only $8.9\,e^{-3}$ this the result reconstruction:

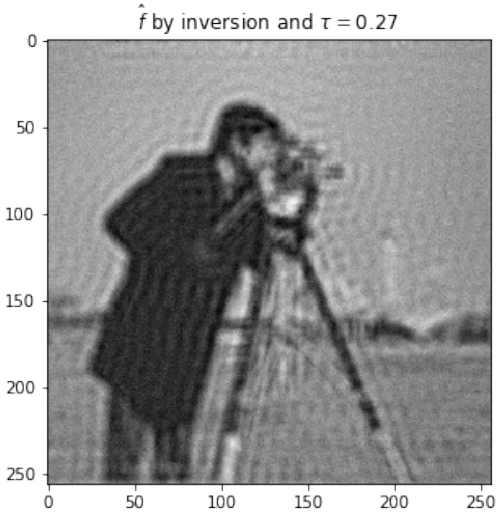

Figure 10. "Best" reconstruction

```
mse: 8.87e-03
```

Even if the *mse* is smaller I still prefer the reconstruction with $\tau = 0.1$.

2 Regularization

Another simple way to avoid the noise amplification is to regularize the filter H by adding some

small value λ that avoid (too) small values:

$$H_\lambda = \frac{H^*}{|H|^2 + \lambda^2}$$

Note that H_λ is already inverted respect to H.

My first experiment is again with $\lambda = 0.1$:

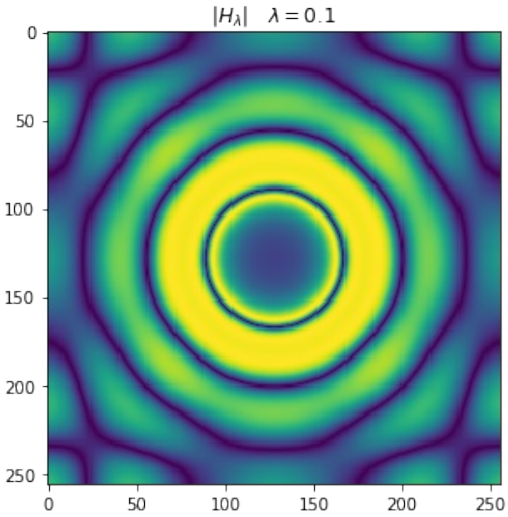

Figure 11. Reconstruction filter

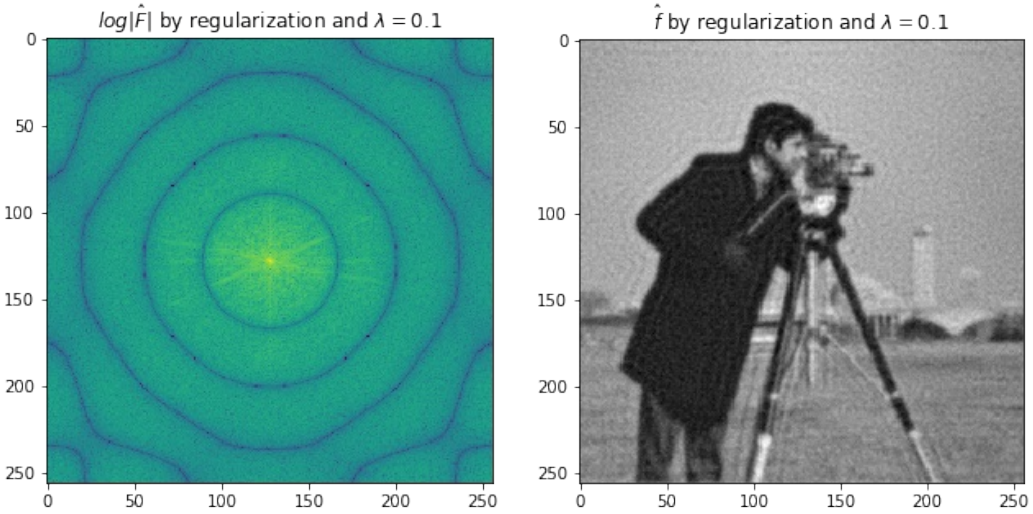

Figure 12. Reconstruction by regularization

As before I looked which λ minimize the error:

```
Lambda optimal: 0.10 - 3.9e-03
```

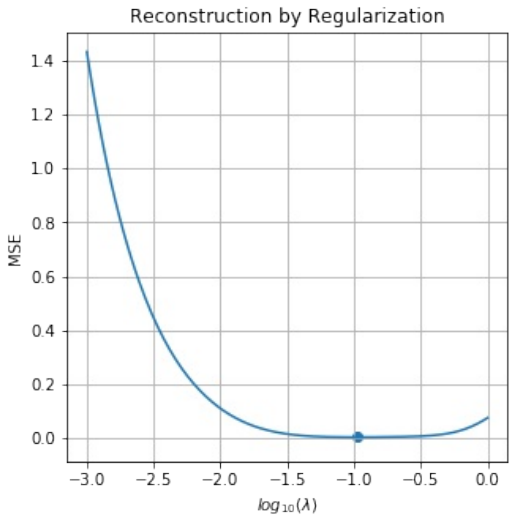

Figure 13. *mse* depending on λ

In this case appear that $\lambda = 0.1$ is already the best solution with an *mse* of only $3.9\,e^{-3}$, while the best reconstruction by treshold was capable to optain $8.91\,e^{-3}$, more or less the double. As well the visual reconstruction is much more pleasant.

3 The Wiener Filter

The Wiener filter use some prior information, so is a statistical approach, on the signal to reconstruct it in such way to minimize the *mse*. In particular it use the hypothetical power spectrum of noise (S_z) and signal (S_f) to do such reconstruction:

$$\hat{F} = \frac{H^*}{|H|^2 + S_z / S_f} G$$

I recall that $|H|^2 = H \cdot H^*$ and the block $\frac{|H|^2}{H}$ as been replaced by only H^*.

Exercise 3.1

By using the real power spectrum of the signal and of the noise the result is the following:

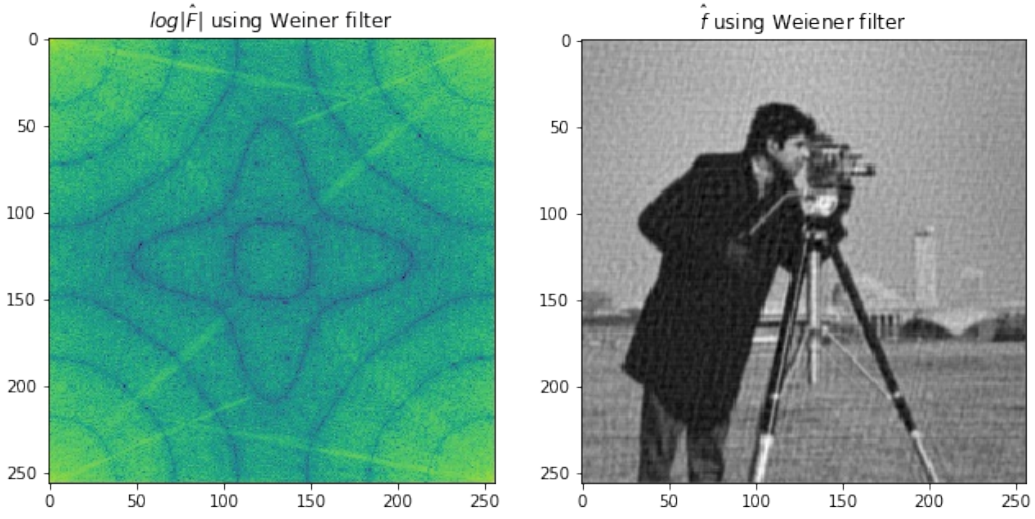

Figure 14. Reconstruction using the Weiner filter

`mse: 2.7e-03`

The resulting reconstruction is slightly better than using the regularization. However we used the original image to obtain such reconstruction, and in real application we don't have access to this information, but it's possible to use the power spectrum of another natural image (as most of real images have similar spectrum) to do such reconstruction:

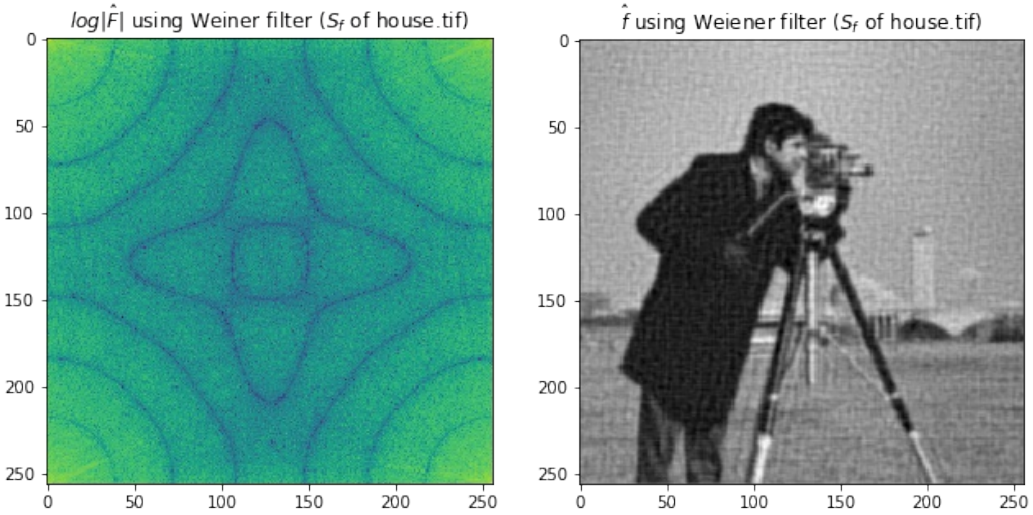

Figure 15. Reconstruction using the Weiner filter but different S_f

`mse: 3.5e-03`

Even in this case the reconstruction is slightly better then using a simple regularization. The lost of performance against using the original S_f are around $\sim$30%.

Then I try to use the wiener filter implemented in the *skimage* python package, this implementation doesn't ask for any prior S_f or S_n so I suppose is some how coded or extracted from G directly:

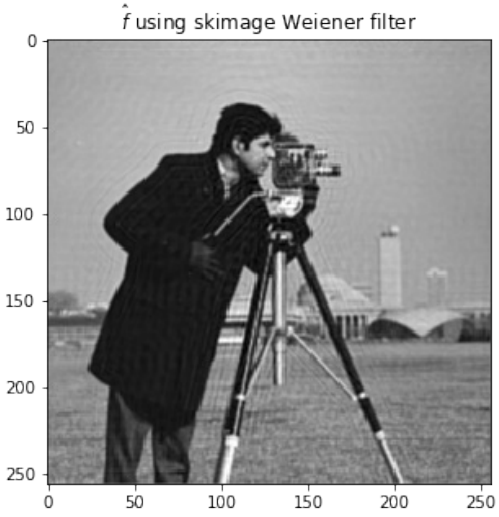

Figure 16. Reconstruction by using *skimage* Wiener filter

```
mse: 1.1e-03
```

The result of such reconstruction is even better then the previous ones (∼3 times better than using the original S_f).

After a quick look at their code to understand what is difference, their implementation is very close to our:

```
reg, _ = uft.laplacian(image.ndim, image.shape, is_real=is_real)
wiener_filter = np.conj(H) / (np.abs(H) ** 2 + balance * np.abs(reg) ** 2)
return uft.uifft2(wiener_filter * uft.ufft2(image))
```

Where `reg` replace our $S_z \,/\, S_f$ and is computed using a laplacian filter (H_L, second derivate of the image), and the "balance" (λ) constant is a constant to improve the balance between the data adaquacy and the restoration.

The results is:

$$\hat{F} = \frac{H^*}{|H|^2 + \lambda |H_L \cdot G|^2} G$$

Exercise 3.2

Another way to compute the S_f without having access to the original image is to compute it from the noisy image itself:

$$r(x, y) = \sigma_f^2 \rho^{-\sqrt{x^2+y^2}} + \mu_f$$

where σ_f^2 is the variance of the image and μ_f its mean. While ρ is the local correlation:

```
var : 4.92e-02
mu  : 4.66e-01
rho : 9.82e-01
```

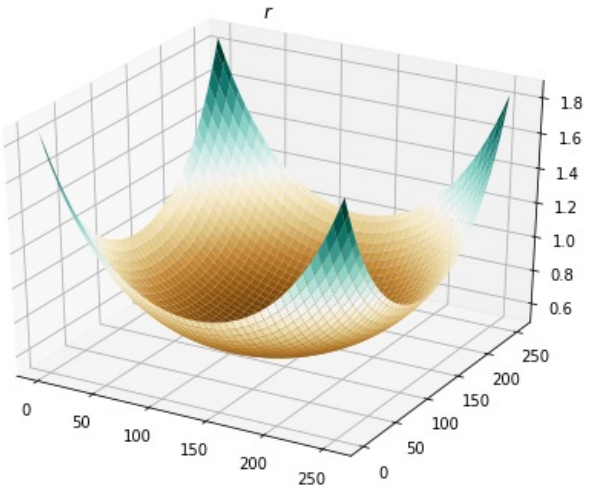

Figure 17. Computed r

The resulting reconstrution has *mse* (and visual result) very similar to the reconstruction with S_f from a different image:

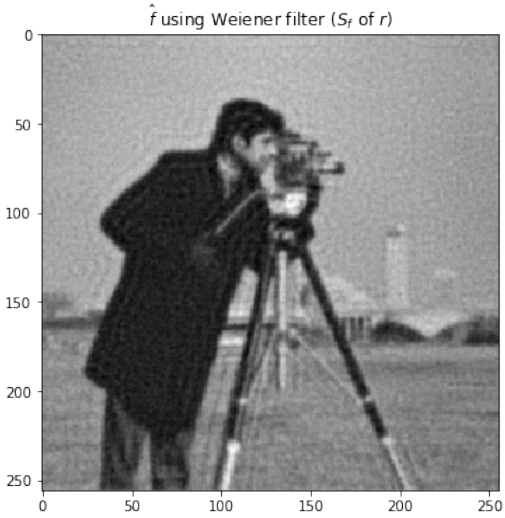

Figure 18. Reconstruction using r instead of S_z / S_f