Experimental report

Article 1: *EPLL: An Image Denoising Method Using a Gaussian Mixture Model Learned on a Large Set of Patches*, Hurault, Samuel and Ehret, Thibaud and Arias, Pablo

Experiments on EPLL

EPLL is an improvement of the NL-Bayes algorithm. However, there is 2 main differences.
First of all, EPLL uses a Gaussian Mixture to model all the similar patches. The second difference is that this GMM is trained on an extern database. This means that EPLL is a global algorithm compared to NL-Bayes. Solving a Maximization A posteriori Problem leads to find the optimal denoised patch.

To conduct those experiments, we will first try the algorithm on 3 different images with the same parameters: noise standard deviation at 30, step is 4, maximum rank is 50 and we apply the algorithm on 2 scales.

We will then experiment a bit with the parameters before comparing with other denoising algorithms.

Experiments on images

First, we will try with a completely artificial image that I created. Indeed, as the EPLL model is trained from a database, we will try to push the model to its limits with an image composed of unknown patches.

Finally, we notice that the results are quite good (see Figure 2). However, the difference image (see Figure 3) shows us patches of low frequency. These patches are probably due to the stride which is 4.

Figure 1: Original image

Figure 2: Denoised image (PSNR: 42.83dB)

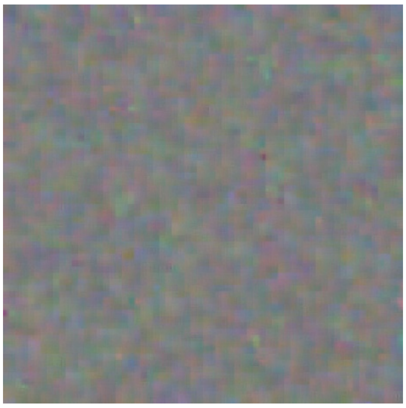

Figure 3: Difference

The second experiment is on an image with an object but weakly textured. We try to put forward an image more contrasted than before but still with a weak texture.

Once again, we notice that the denoising is good (see Figure 6). On the other hand, the edges do not render well, especially on the separation between the desk and the wall and on the writing on the book. The difference image shows that the algorithm has once again difficulties on the low frequency noise. Thus, the wall is completely homogenized, which makes the image very clean, without denoising.

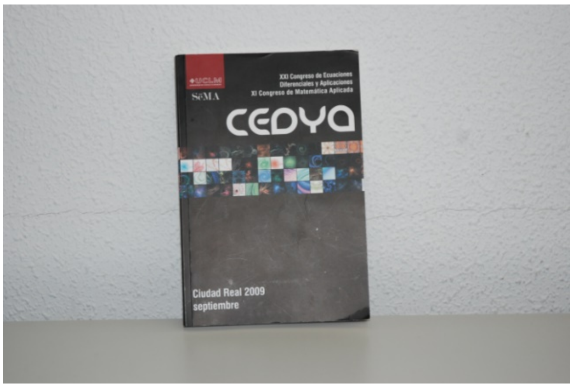

Figure 5: Original image

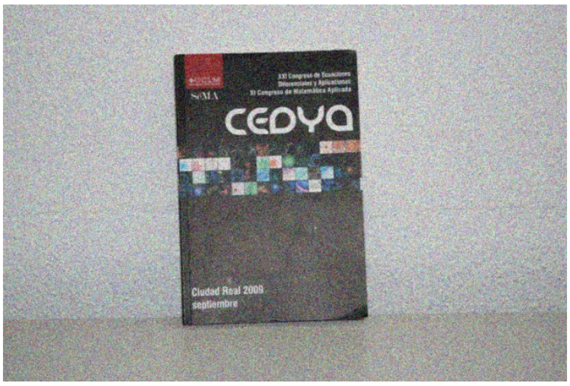

Figure 7: Noisy image

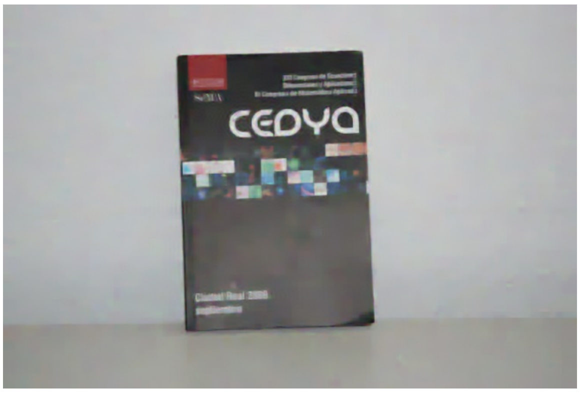

Figure 6: Denoised image (PSNR: 33.25dB)

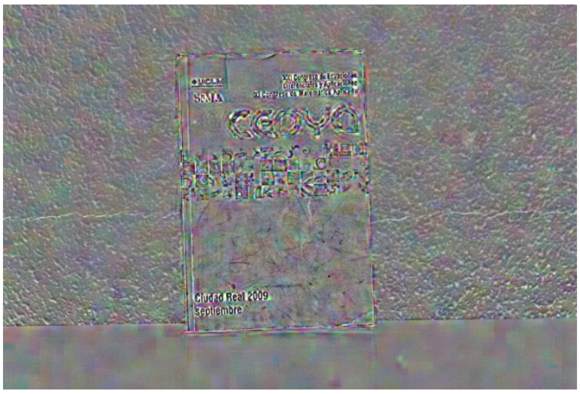

Figure 4: Difference

Finally, the last image is completely textured and natural.

The algorithm produces a fairly clean result overall the clouds are well rendered, as is the crosswalk. On the other hand, if we zoom in the image, we realize that the textures are very poorly rendered. There is a lot of blur, which attenuates the already weak contrasts, especially on the balconies (see Figure 9). The high contrast areas are not spared either, like the street lamps which seem to be added in post-production.

Figure 11: Original image

Figure 8: Noisy image

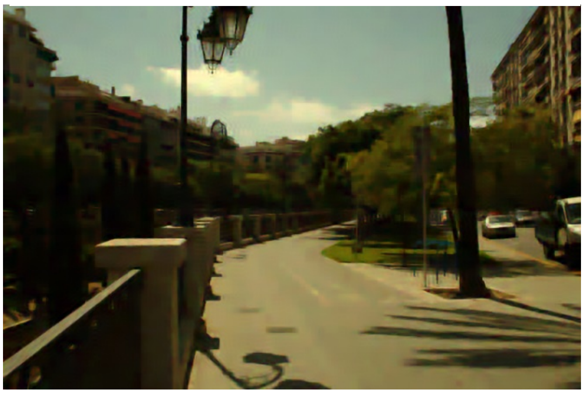

Figure 9: Denoised image at multiscale (PSNR: 28.61dB)

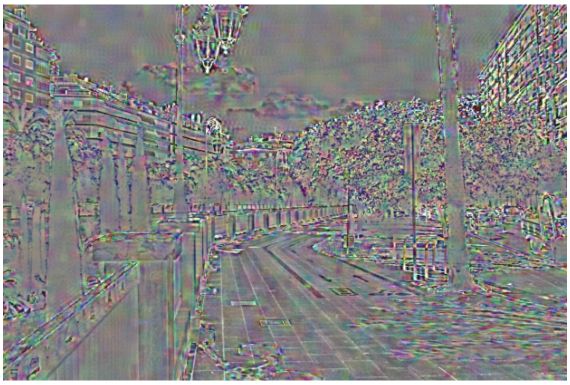

Figure 10: Difference

To conclude, we can notice that the overall image reconstruction is good. This method succeeded in denoising different types of images and gives high performance. Moreover, when playing with the above images, I noticed the multiscale was often doing worse than the single scale. This means that the algorithm has difficulties retrieving lower frequency noise. Furthermore, this method performs better in the case of homogeneous images. When the image become more structured, the denoising performance decreases. This method is not able to differentiate between the noise and small frequencies found in the vegetation area. In addition, the EPLL is not able to reconstruct the edges and the small details.

Understanding parameters

In this part, we will rather focus on the influence of hyperparameters on the reconstruction on the natural image above. In all, we will keep a constant noise of 30, and we will always take the single-scale denoised image.

Influence of step:

We begin by noting that increasing the step decreases the PSNR. Indeed, there are less debruised patches and thus their aggregation is less good.

On the other hand, we notice that the results are visually better for a small step. Indeed, with a high step, we notice checkerboard patterns, very visible to the human eye.

Finally, despite a higher computation time, a low step is better.

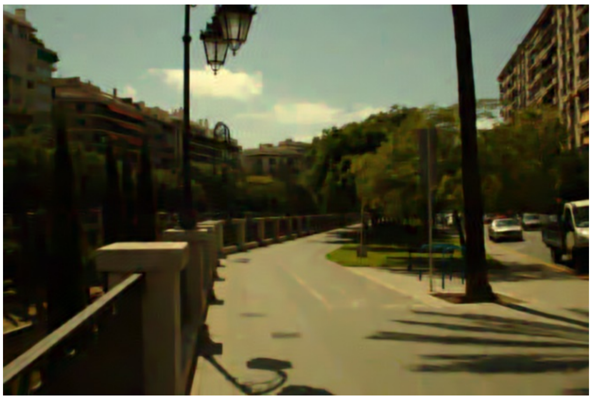

Figure 13: Denoised image with step=3 (PSNR: 29.57dB)

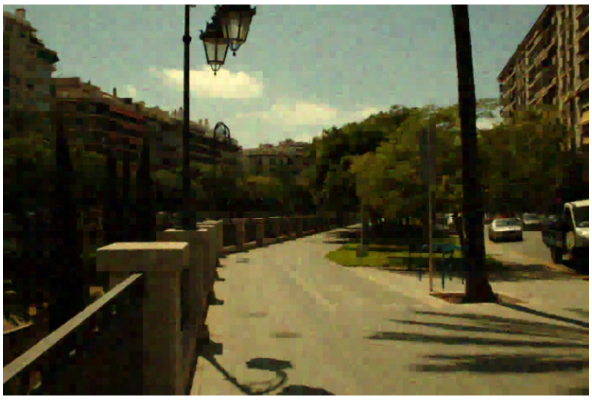

Figure 12: Denoised image with step=8 (PSNR: 27.95dB)

Influence of maximum rank:

Another parameter to compare is the maximum rank of the covariance matrices.

Obviously, the reconstructed images have better quality when the rank is higher because our gaussian mixture model takes more information into account.

Finally, the gain in dB is quite low (about 0.3dB) but the visual differences are quite noticeable. Thus, with a high rank, the images are much deeper. This comes from the fact that the model fetches information further away, which is contained in the high frequencies and therefore the details are more complete. We can notice the differences in the foliage once again (see Figure 15, Figure 14, Figure 16)

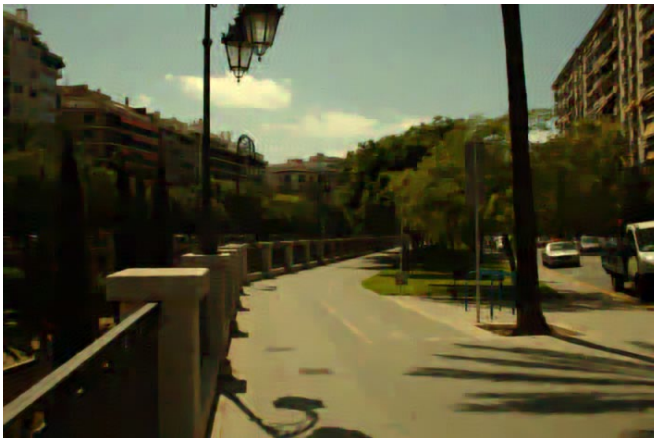

Figure 15: Denoised image with rank = 50% (PSNR: 29.51dB)

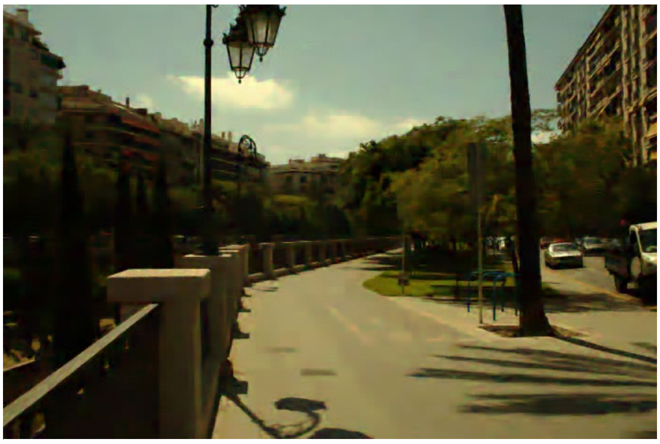

Figure 14: Denoised image with rank = 75% (PSNR: 29.67dB)

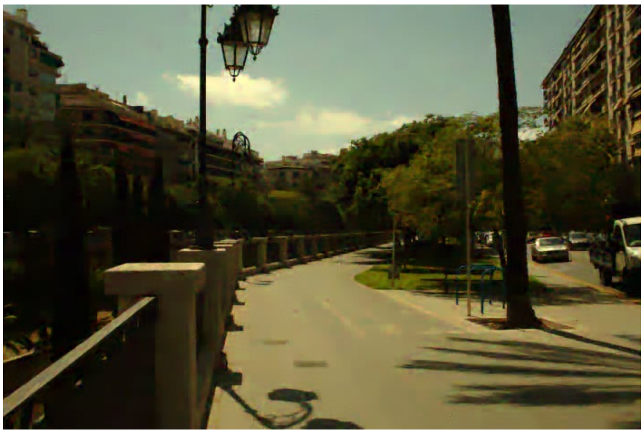

Figure 16: Denoised image with rank = 100% (PNSR: 29.77dB)

Influence of the scale:

We noticed earlier that the scale did not make much difference for a noise of 30. Let's check this for other noise values.

Intuitively, we expect the model to produce similar results whatever the scale for low noises and that the scale really intervenes only if the noise is high.

For a low noise (see Figure 20, Figure 19), we realize that the single-scale denoiser is more efficient, with a higher PSNR value (+ 1.4dB). Indeed, we can interpret this by the fact that the multi-scale will complicate the task, and will therefore not be accurate enough. Moreover, we notice that the multiscale has Gibbs oscillations, especially next to the street lamps. These artifacts are much more visible than on the DCT.

On the contrary, for high noise values (see Figure 18, Figure 17), the multiscale image is better (see Figure 17). Although the performance in PSNR is similar, the single-scale algorithm keeps the noise in low frequency, which makes many multi-coloured patches appear on the image (see Figure 18). This is because using different scales increases the redundancies for a pixel. The artifacts created by the multi-scale will then be negligible compared to the noise.

This approves what we saw during the lesson.

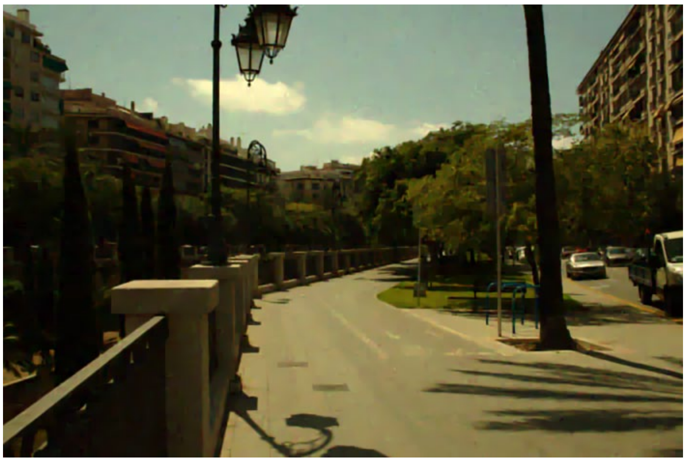

Figure 20: Denoised image single-scale σ=20 (PSNR: 31.17dB)

Figure 19: Denoised image multi-scale σ=20 (PSNR: 29.70dB)

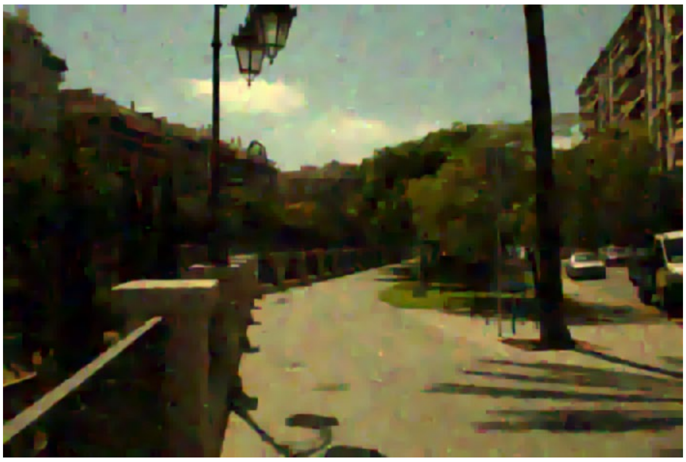

Figure 18: Denoised image single-scale σ=70 (PSNR: 25.95dB)

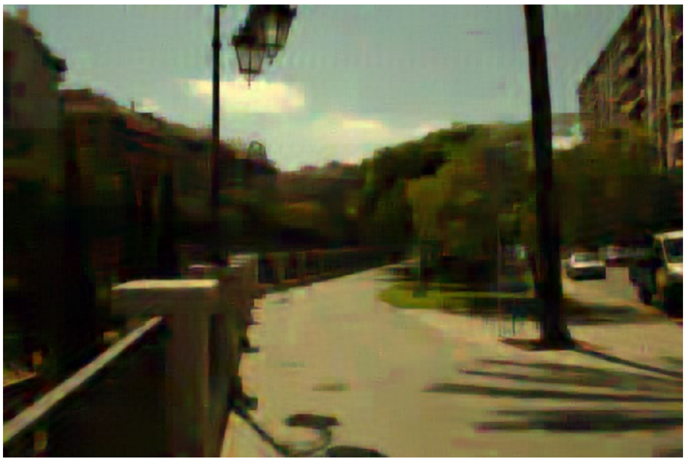

Figure 17: Denoised image multi-scale σ=70 (PSNR: 25.64dB)

OPP transformation:

Finally, the last change that can be studied is the move of the image into the PPO domain.

We do not notice any difference, even on the difference image (see Figure 22, Figure 21). This feeling is approved by the equal PSNR.

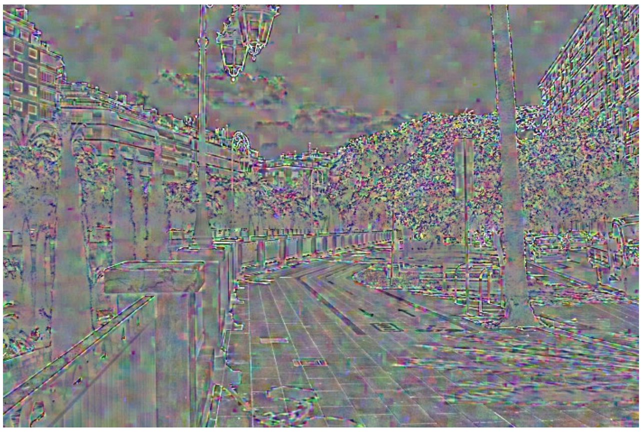

Figure 22: Denoised image without OPP transformation (PSNR: 29.53dB)

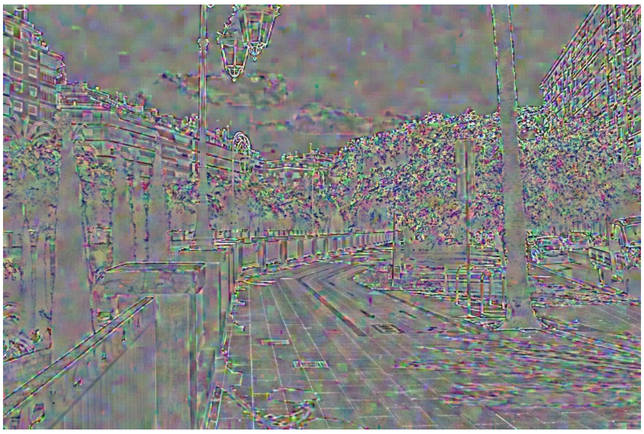

Figure 21: Denoised image with OPP transformation (PSNR: 29.52dB)

Comparison with other algorithms

In the end, we can try to compare the different algorithms we have used in class, to a natural image. In all cases, I would take the best results on all hyperparameters, despite the computation time, for a noise of 30. The results are on Figure 26, Figure 25, Figure 23 and Figure 24.

First of all, from a performance point of view, the BM3D algorithm has the best PSNR, and EPLL has the worst. We also notice that each of these algorithms has different weak points. For example, EPLL has difficulties on the edges while Multi-Scale DCT gives an impression of blur. It is then difficult to choose which is the best algorithm to implement on a camera, because the use cases are very varied. Finally, we are looking for the most flexible algorithm possible for use in a smartphone. On the contrary, we will prefer a specific algorithm for industrial uses.

Moreover, we understand quite quickly that the complexity of an algorithm is not a guarantee of performance. Multi-Scale DCT is the algorithm with the most intuitive assumptions, and yet its results are very competitive. On the other hand, we realize that the performance measurement with PSNR is not very reliable. Indeed, it is a metric that puts low and high frequency noise at the same level. However, humans are first struck by the low frequencies before noticing the high frequencies when they deeply analyze an image.

It is obviously quite difficult to select a single best algorithm on an image. Personally, I prefer the BM3D algorithm, because the egdes are well defined which reinforces the contrasts of the image.

Figure 26: Multi-Scale DCT Denoising (PNSR: 30.73dB)

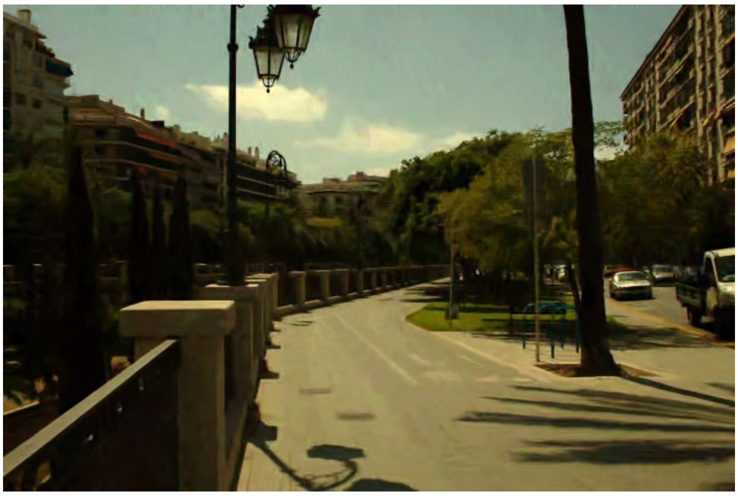

Figure 25: BM3D (PNSR: 31.29dB)

Figure 23:NL-Bayes (PSNR: 30.98dB) (Note: the first step was better)

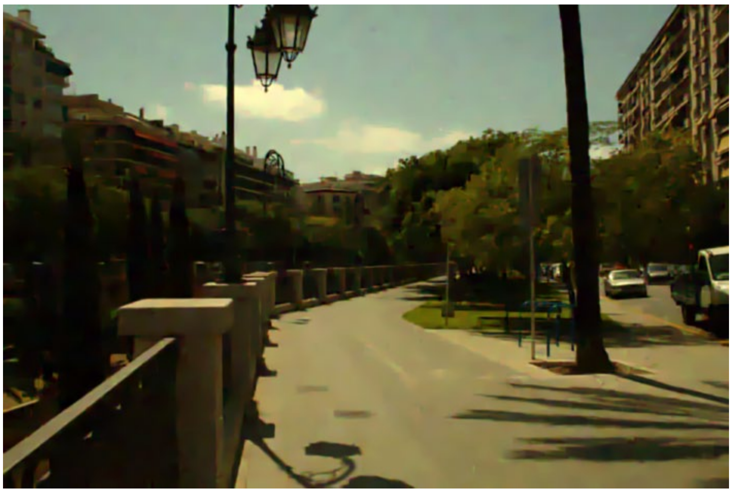

Figure 24:EPLL (PSNR: 29.81dB)

Zoran-Weiss Gaussians Mixture Model

The proposed document describes the Gaussian Mixture components along with their probabilities. They introduce only 200 gaussians, trained from a database. The gaussians have very low probability of happening overall. The highest probability is 0.03950 while the lowest is 0.00003.

These gaussians all decompose very different elements. Some of them represent edges while others represent textures. Some gaussians have very strong pattern (see Figure 27, which represents the eigen vectors of the gaussian) and thus tend to be very sparse. On the other hand, some gaussians represent textures (see Figure 28).

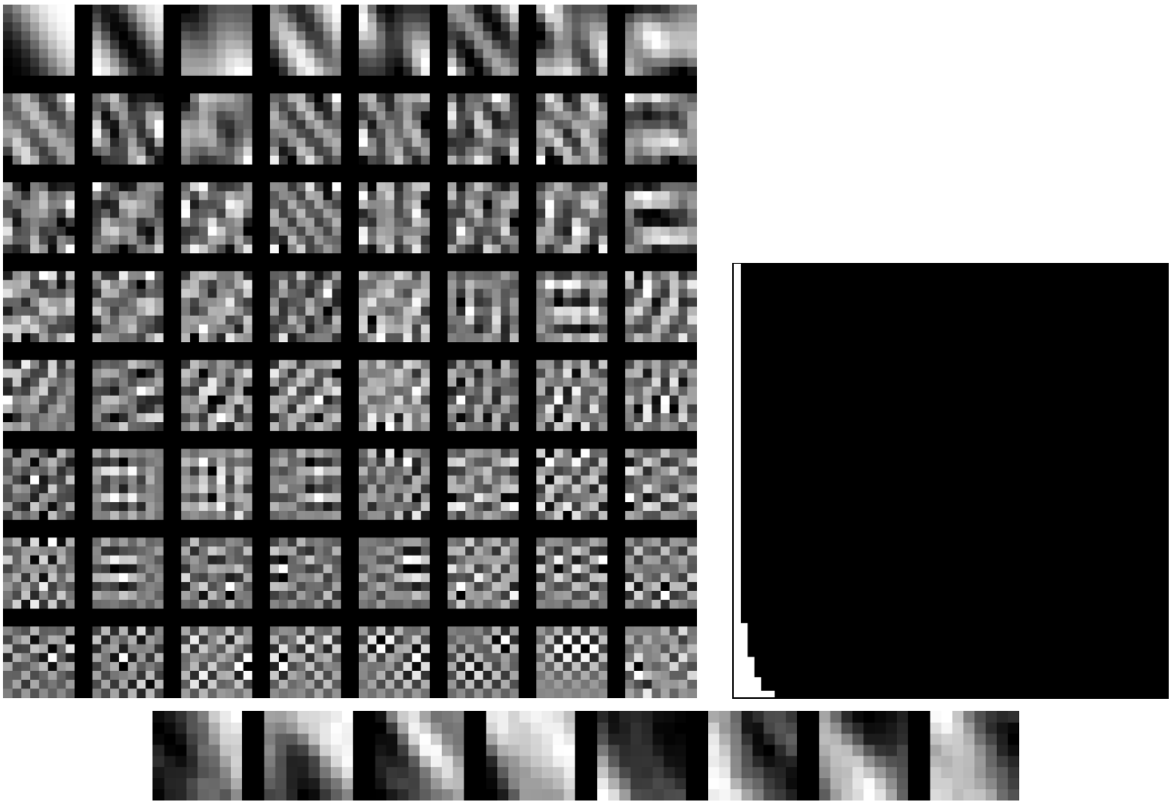

Figure 27: Eigen vectors, eigen values and simulations of the 8th gaussian.

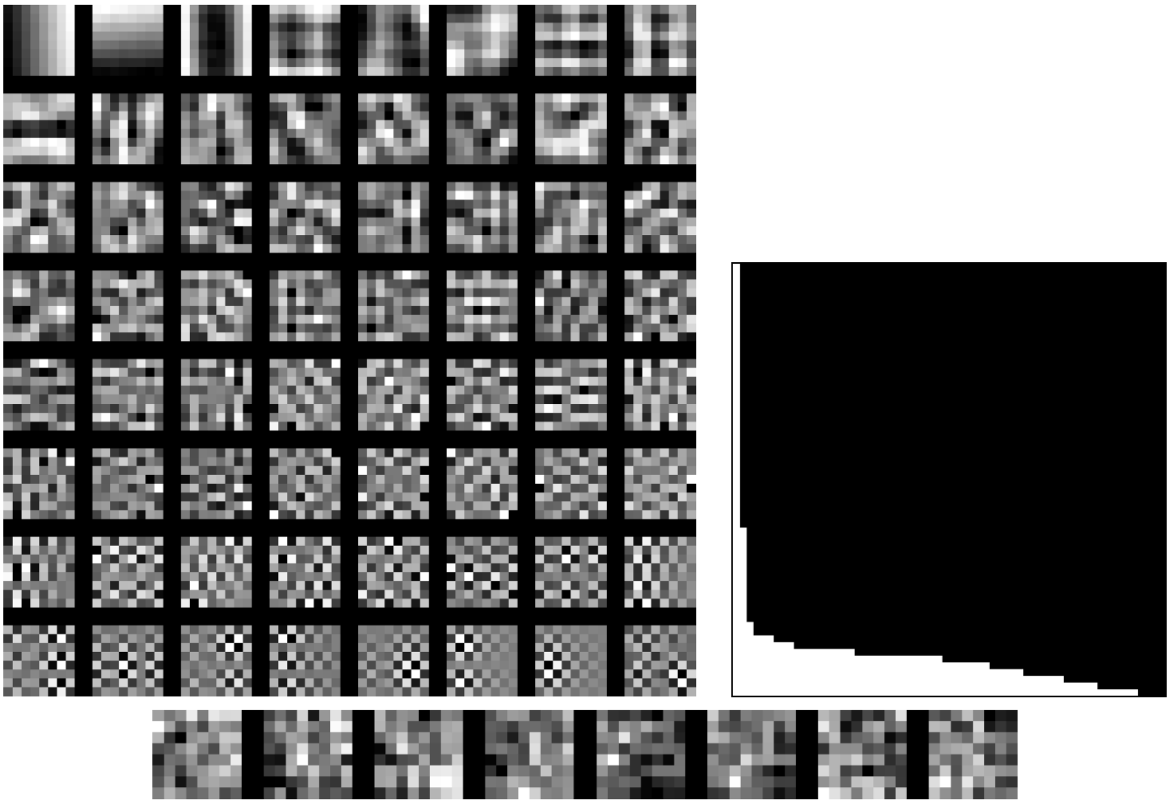

Figure 28: Eigen vectors, eigen values and simulations of the 141st gaussian.