Exercise 1

Figure 1: Toy stripes

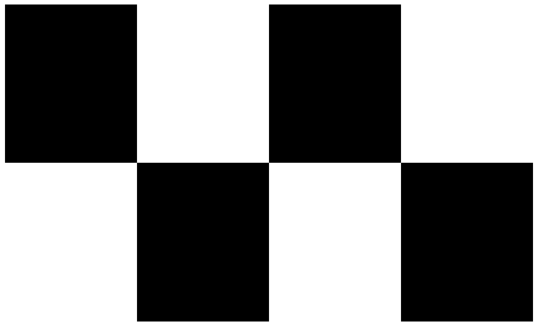

Figure 2: Toy checkerboard

Exercise 2

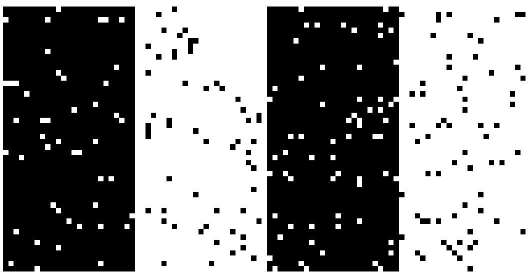

Figure 3: Noisy toy stripes

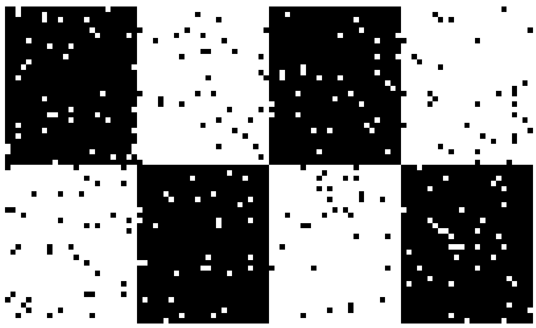

Figure 4: Noisy toy checkerboard

Figure 5: Image with Gaussian noise and $\sigma = 5$ (left), $\sigma = 25$ (middle) and $\sigma = 50$ (right).

Figure 6: Noisy toy stripes median filtered.

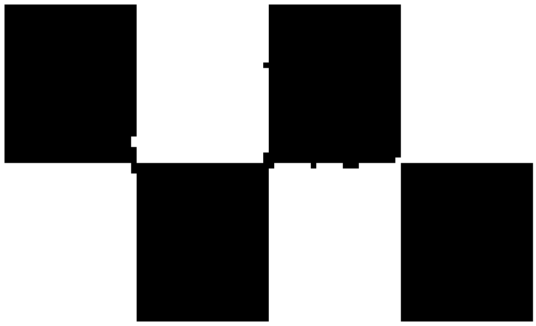

Figure 7: Noisy toy checkerboard median filtered.

Figure 8: Image with Gaussian noise and $\sigma = 5$ (left), $\sigma = 25$ (middle) and $\sigma = 50$ (right) median filtered.

Observation

For the artificial images the noise was completely deleted except for some small artifacts. In the natural image the denoising also gave a good result, although the image was significantly blurred. In all images edges are preserved. The PSNR values (see Table 1) all increased significantly for the denoised versions, especially for the artificial images.

Varying noise for median filter

We have included noisy images and the corresponding filter results for sigma values 5 and 50. The smaller sigma, the smoother the filtered image becomes. Natural images usually contain smooth areas in which the median filters deletes any high frequencies. If there is a lot of noise in an image the result looks less smooth and more like there is still noise left, which makes sense since also the median in each window can shift significantly if the noise is strong, so there is often a big difference for pixels even with overlapping windows.

	Noisy Original	Median	Gauss MRF	Median + Gauss MRF	Student MRF
Stripes	13.3348	29.7881	18.9248	22.5530	13.4212
Checker	13.4040	26.6421	17.9571	20.5386	13.4872
Image	20.4622	24.1984	25.0029	23.2165	23.5804

Table 1: PSNR for different denoising approaches.

Exercise 3

Parameter Tuning

We picked σ as 10 and η as 0.1. For bigger sigma the images tend to not get denoised at all. For smaller sigma the computed result is very far off from the ground truth image. For smaller values of eta the gradient ascent did not always converge. For bigger values of eta the gradient ascent converged quicker, but the achieved PSNR value (see Table 1) of the result was lower.

Median filter vs Gauss MRF

For the salt and pepper noise the median filter works a lot better, which makes sense since this kind of noise only affects a few single pixes, which leaves the median in the filtering window mostly intact. The Gauss MRF only adds blur to the images resulting in no great effect. On the natural image the median filter adds a lot of blur, deleting all the noise but also many image features. The Gauss MRF works better on natural images, leaving more image features intact. These observations are verified if one consideres the PSNR values, which also indicate that for the salt and pepper noise the Median filter is better and for the gaussian noise the Gauss MRF is better.

Initializing gradient ascent with the result of the median filter

According to the PSNR values (see Table 1) this approach works better for the artificial videos and worse for the real image. The visual impression for us was mostly an additional slight blur in all cases. In terms of performance the gradient ascent converged slower if we initialized it with the result of the median filter in all cases.

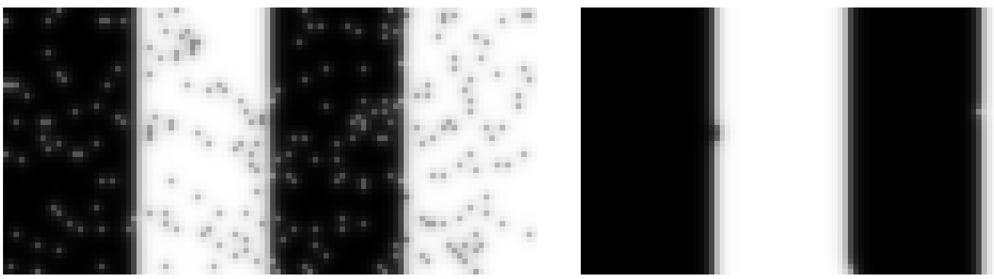

Figure 9: Noisy stripes filtered with MRF Gaussian (left) and filtered with MRF Gaussian initialized with Median Filter result.

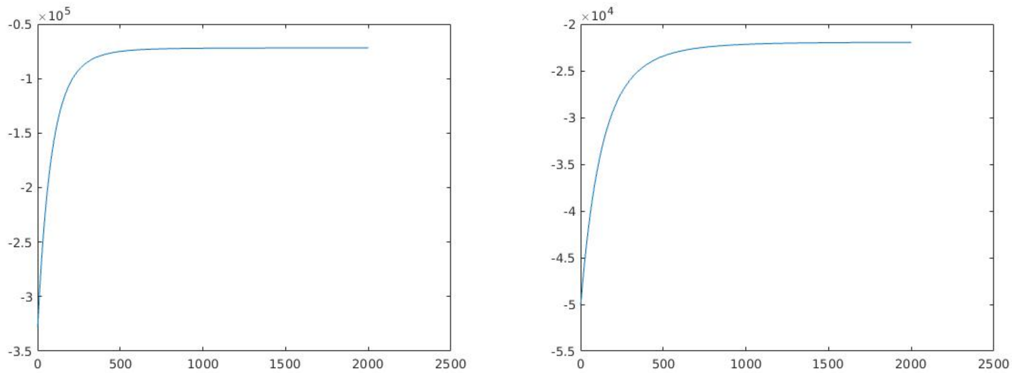

Figure 10: Posterior curve for noisy stripes filtered with MRF Gaussian (left) and filtered with MRF Gaussian initialized with Median Filter result.

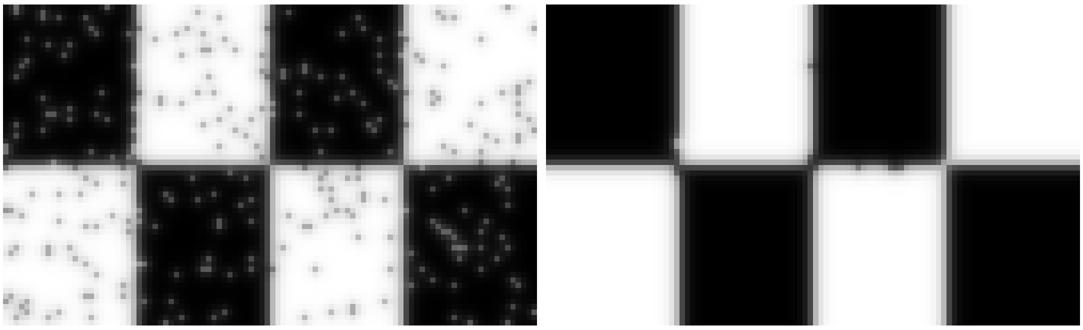

Figure 11: Noisy checkerboard filtered with MRF Gaussian (left) and filtered with MRF Gaussian initialized with Median Filter result.

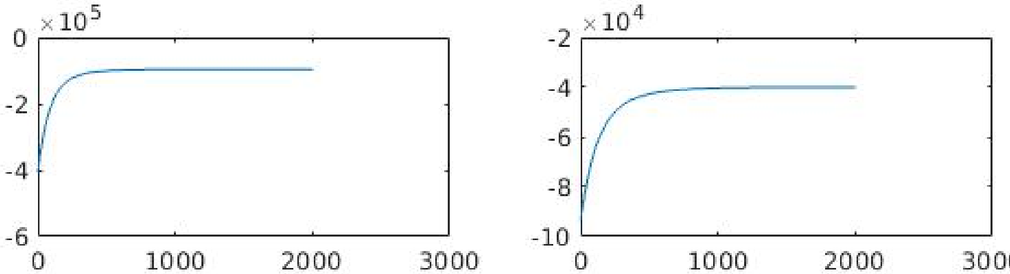

Figure 12: Posterior curve for noisy checkerboard filtered with MRF Gaussian (left) and filtered with MRF Gaussian initialized with Median Filter result.

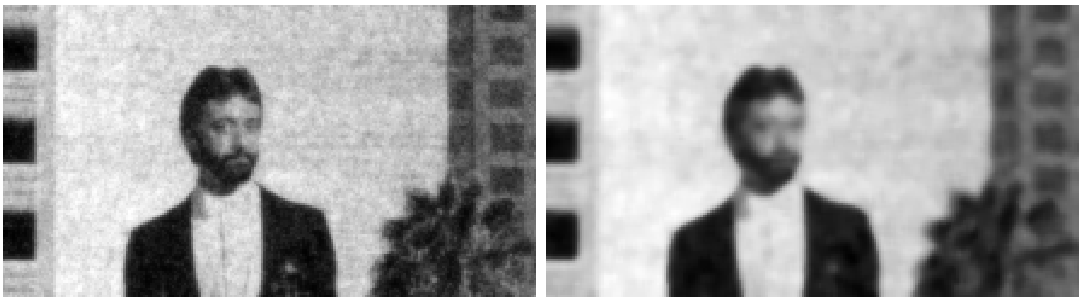

Figure 13: Noisy image filtered with MRF Gaussian (left) and filtered with MRF Gaussian initialized with Median Filter result.

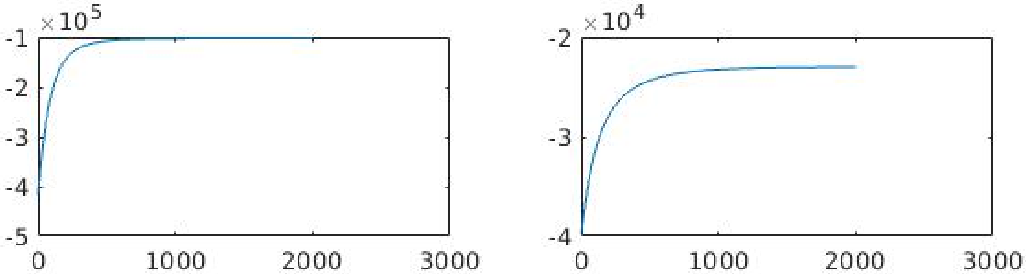

Figure 14: Posterior Curve for noisy image filtered with MRF Gaussian (left) and filtered with MRF Gaussian initialized with Median Filter result.

Exercise 4

Convergence

The overall convergence is slower than when using the Gaussian prior in exercise 3. It takes the full 2000 iterations to approximately converge.

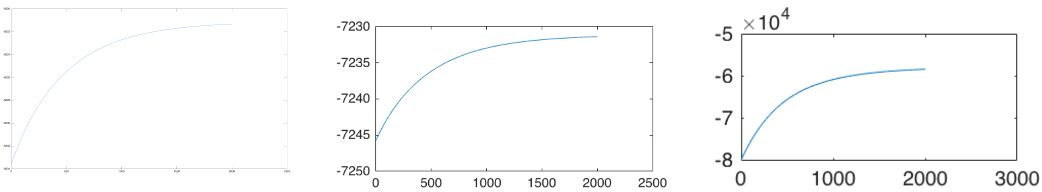

Figure 15: Increasing log-posterior curve for stripes (left), checkerboard (middle) and image (right) using MRF Student Filter.

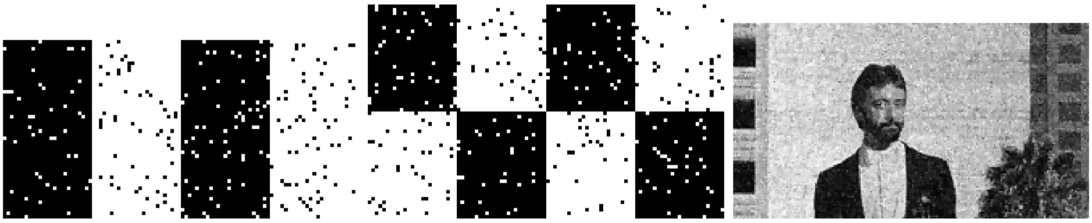

Figure 16: Denoising results for stripes (left), checkerboard (middle) and image (right) using MRF Student Filter.

Performance

Although there is a slight improvement in PSNR (see Table 1) when denoising the artifical images, this difference is not visible when looking at the images. This can be explained by the fact that the Student-t prior is suited for natural images but not for artificial ones. For the real image, the PSNR improves significantly. The difference is also visible when comparing the images. Overall, the result is less noisy and especially not as blurry as the result when using the Gaussian prior. Edges and also face features are maintained better.

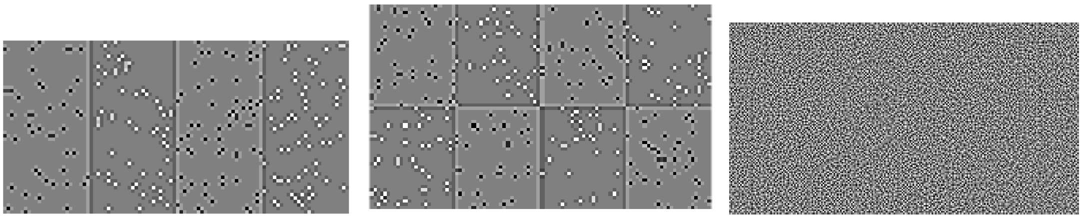

Figure 17: Gradients of the log-prior for stripes (left), checkerboard (middle) and image (right) using MRF Student Filter.

Gradients of the student prior

- artificial images: The gradient of the student prior has a high absolute value in the noisy pixels. The direction of the gradient is opposite for noisy salt-pixels in a black area and noisy pepper-pixels in a white area of the image. That is because the pixel values have to change in opposite directions (increase, decrease) to achieve a better value in the student prior that wants to make pixels similar to their neighbors. Therefore, also the gradient at the edges of the stripes or the checkerboard is stronger but not as strong as in the noisy pixels since roughly half of the neighbors have the same color.
- real image: The gradient in the real image looks in most regions more or less random what also makes sense because the Gaussian noise was random. Only in

edge regions, the gradient is significantly less strong. That matches the result that the edges are less blurry when using this filtering technique.

Exercise 5

Reasonable assumption?

This depends on the noise type that is present in the image. Noise can be added for example as salt-and-pepper noise caused by transmission errors or defects in the camera sensor. In this case, only some pixels are affected randomly and therefore the noise is independent. In the digitization of old movies, film grain noise occurs in lines, so the noise is not independent. Overall, in many applications the assumption is reasonable.

What happens?

- Gaussian prior: When using the Gaussian prior, the whole image is blurred. This reduces the noise in the area of the stripe but blurrs also all edges in the remaining image.
- Student prior: Noisy pixels whose values do not differ so much from the true value are denoised properly. More intense noise remains. But other regions of the images (not affected by the noisy stripe) are barely changed.

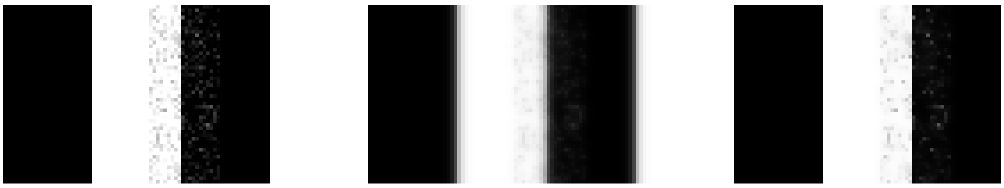

Figure 18: Toy stripes with noisy stripe (left). Denoising results with MRF Gaussian (middle) and MRF Student (right).

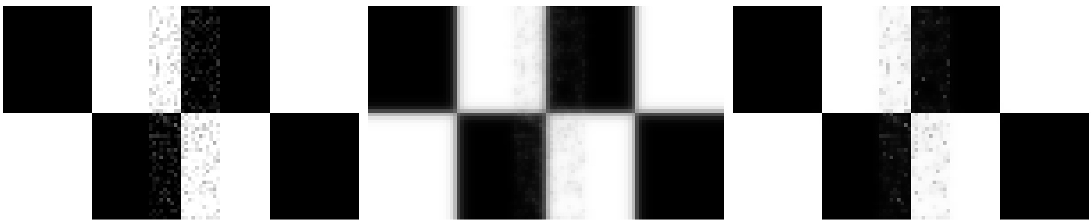

Figure 19: Toy checkerboard with noisy stripe (left). Denoising results with MRF Gaussian (middle) and MRF Student (right).

Figure 20: Image with noisy stripe (left). Denoising results with MRF Gaussian (middle) and MRF Student (right).

Exercise 6

There are several ways to denoise a color image. One idea is to denoise the different channels of the image separately using the same techniques that we used for grey value images. One could also use another color space, like HSV, where the color can be more easily split into color and brightness information (color noise is more disturbing than brightness noise for the human).