Exercise Sheet 06

Deadline: 06.06.2018, 1pm
Regulations:

- You need $\geq$ **50% of all points** in the weekly exercises and successfully work on a $\geq$ **2 weeks fulltime project** and submit a **written report** in order to pass this class.
- Hand in your solutions in **groups of two or three people**.
- Make sure your code works with python 3.6 or later.
- For exercises that include coding, hand in a **Juypter Notebook** for each exercise containing all your answers, results, plots and code. Separate the different sub-exercises clearly by using headings and describing text. Label each file with the number of the exercise. Before submission, make sure your Jupyter Notebook compiles without any errors when you do Run All. Additionally hand in a PDF version of the Notebook.
- For exercises that can be solved without coding, hand in a **PDF file** for each exercise containing all your answers and results. Separate the different sub-exercises clearly by using headings and describing text. Label each file with the number of the exercise.
- Submit all your results in a **single** .zip archive. The title of the submitted ZIP-file as well as the subject line of your email **must** start with **EX06** followed by the full names of **all group members**. **People not mentioned in the subject will not get credits.**
- Submit all your results to mlcvss18@gmail.com.
- Submissions that do not adhere to the deadline and regulations, or that are chaotic, will not be graded.

Exercise 1. 5 P.

Consider a 3x3 image represented as a grid graph with the usual four connected neighborhood. Assume that all edges have a weight of 1 and assume non-periodic boundary conditions. How does the Graph Laplacian matrix look like?

Exercise 2. 15 P.

In this exercise we will use a one dimensional Gaussian Markov Random Field represented by a simple chain of latent variables z_i and corresponding observations x_i. Each z_i is connected to its left and right neighbors. Further assume that each observation x_i is Gaussian distributed with mean z_i and variance σ^2.

This model can be represented with the following energy function:

$$E(z) = \underbrace{\frac{1}{2\sigma^2}(\mathbf{x}-\mathbf{z})^T(\mathbf{x}-\mathbf{z})}_{\text{"unaries"}} + \underbrace{\frac{1}{2}\mathbf{z}^T\mathbf{Q}\mathbf{z}}_{\text{"second order"}} \tag{1}$$

where the matrix $\mathbf{Q}$ encodes the structure of the graph. When $\mathbf{Q} = \mathbf{Q}(\mathbf{x}, \gamma)$ is a function of the observations the second order terms are no longer a prior and we instead directly model the posterior distribution. Such models are then called Conditional Gaussian Random Fields.

Use the 1D function from the file `fn.npy` available on the website and complete the following tasks.

a) Plot the precision and covariance matrices when all *non-zero* off-diagonal values of Q have a constant value. Which constant value do you need to use? How do the diagonal elements look like?

b) Plot the precision and covariance matrices when all *non-zero* off-diagonal values of Q are given as $-\exp(-\gamma(x_i - x_j)^2)$ with $\gamma > 0$. How do the diagonal elements have to be computed this time?

c) Compute and plot the MAP solution for different values for σ and γ. Explain the effects of both parameters on the solution with examples.

★ Bonus Exercise 3. 5 P.

In most approaches that have been discussed in this class the energy function was the sum of two terms: a data fidelity term which measure the likelihood of the input image given the output and a prior term which encodes prior assumptions about the output. In tasks such as image denoising, where the output is a natural image, the prior should capture some knowledge about the space of natural images. This space is only a tiny fraction of the space of $N \times M$ matrices. One robust property of this space is the distribution of filter output when applying derivative-like filters to a natural image [1]. In this exercise, we will compute this distribution for the image shown in figure 1[1] and compare it to the prior assumption made by GMRFs.

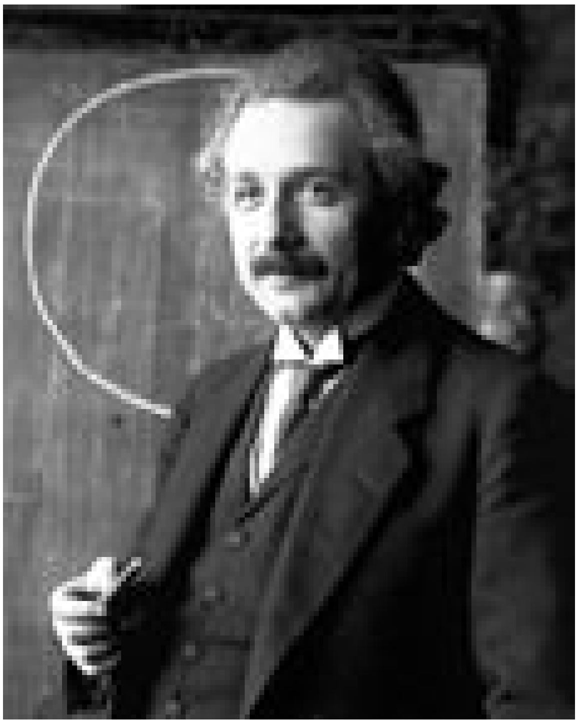

Figure 1: Natural Image

a) Load the image from Einstein.jpg into a numpy array.

b) Compute the derivative of the image by convolving it with an appropriate filter (e.g. Sobel derivative).

c) Vectorize the output and plot a histogram of it with log-scale on the y-axis.

d) Plot the quadratic smoothness prior used in GMRFs $\exp\left(-.5 \cdot (z_i - z_j)^2\right)$ and the Cauchy loss $\exp\left(-.5 \cdot \log(1 + .5 \cdot (z_i - z_j)^2)\right)$. Comment on the results. Which prior would you choose for the task of denoising images and why?

[1] figure taken from http://www.bhm.ch/de/news_04a.cfm?bid=4&jahr=2006,Gemeinfrei,https://commons.wikimedia.org/w/index.php?curid=12851706

References

[1] Yair Weiss and William T. Freeman. What makes a good model of natural images? In *CVPR*. IEEE Computer Society, 2007.