Variational Methods for Computer Vision
Lecture: M. Möller
Exercises: J. Geiping
Winter Semester 17/18

Visual Scene Analysis
Institute for Computer Science
University of Siegen

Weekly Exercises 3

Room: H-C 6336

Friday, 10.11.2017, 14:15-15:45

Submission deadline: Tuesday, 07.11.2017, 14:15 in Room H-C 6336

Programmming: Email your solution to `jonas.geiping@uni-siegen.de`

Theory

Exercise 1 (6 points)**.** In this exercise we would like to determine the shortest path $\phi : [0,1] \to \mathbb{R}^2$ from a point $a \in \mathbb{R}^2$ to a point $b \in \mathbb{R}^2$, i.e. , $\phi(0) = a$, $\phi(1) = b$. Without restriction of generality you may assume that $a_1 \leq b_1$, and you may assume without a proof that it never makes sense to "go backwards" on the x-axis. Mathematically, the latter means that we may reduce our problem to finding the *graph* of a 1D function. In other words, we may parametrize the desired shortest path ϕ as

$$\phi(x) = (xb_1 + (1-x)a_1, f(x)) \tag{1}$$

and look for the unknown 1D function $f : \mathbb{R} \to \mathbb{R}$.

- The length of a path $\psi : [0,1] \to \mathbb{R}^2$ is given by

$$l(\psi) = \int_0^1 |\psi'(x)| \, dx = \int_0^1 \sqrt{\psi_1'(x)^2 + \psi_2'(x)^2} \, dx.$$

  Compute the length of the path ϕ from (1) in terms of a, b and f.

- Consider the shortest path problem, i.e. the problem of minimizing $l(\phi)$. As a and b are fixed, we only need to consider the unknown function f:

$$\hat{f} = \arg\min_f \; l(\phi).$$

  Determine an optimality condition using the Euler-Lagrange equations!

- Conclude that the derivative of f must be constant.

You have successfully proven that the shortest path between two points is a line!

Exercise 2 (4 points)**.** Think of the discretization of the problem in exercise 1. Assume you discretize f at $n+2$ equidistant points $0 = x_0, x_1, ..., x_n, x_{n+1} = 1$. You know that $f_0 = f(x_0) = a_2$ and $f_{n+1} = f(x_{n+1}) = b_2$, so you only have n variables. Which discrete energy do you want to minimize to implement exercise 1? What is the gradient of your energy in the discrete case?

Programming

Exercise 3 (4 points)**.** Use your optimization framework from the previous exercise sheet to implement the following image denoising algorithm:
Given a noisy image f and parameters α, ϵ, a denoised image $\hat{u}$ is given as the solution of the optimization problem:

$$\min_u \frac{1}{2}\|u - f\|_2^2 + \alpha H_\epsilon(Du)$$

This implies that we are looking for an image that is similar to the input image, but applying its derivatives (D) to the function H_ϵ yields a small result.
Here H_ϵ denotes the Huber-loss

$$H_\epsilon(z) = \sum_{i=1}^{2n} h_\epsilon(z_i)$$

where

$$h_\epsilon(z_i) = \begin{cases} \frac{1}{2}u^2 & \text{if } |u_i| \leq \epsilon \\ \epsilon(|u_i| - \frac{1}{2}\epsilon) & \text{else} \end{cases}$$

. You can set the parameter ϵ to 0.05. D denotes the finite difference gradient operator, i.e. a stacked version of all $u_{i,j,k} - u_{i-1,j,k}$ and all $u_{i,j,k} - u_{i,j-1,k}$ as seen in the lecture.
Test your implementation with the `peppers` image from the first exercise. Read the image and add sufficient Gaussian noise with the 'imnoise' function, then apply your algorithm. Do the same for 'Salt&Pepper' noise of similar visual intensity and compare the results. Find an appropriate value of α for your chosen noise level and each experiment.

Exercise 4 (2 points)**.** Extend your previous implementation to a double-opponent Huber denoising by replacing D from the previous exercise with $\tilde{D} = [D; D_2]$ where D_2 stacks all $(u_{i,j,k} + u_{i,j,l}) - (u_{i-1,j,k} + u_{i-1,j,l})$ and $(u_{i,j,k} + u_{i,j,l}) - (u_{i,j-1,k} + u_{i,j-1,l})$ for all $k \neq l$.

Exercise 5 (Bonus)**.** You can (re-)gain points by fixing the implementation of your energy framework from the last exercise.