Variational Methods for Computer Vision
Lecture: M. Möller
Exercises: J. Geiping
Winter Semester 16/17

Visual Scene Analysis
Institute for Computer Science
University of Siegen

Weekly Exercises 3

Room: H-C 7326
Wednesday, 09.11.2016, 14:15-15:45
Submission deadline: Monday, 07.11.2016, 16:00 in the lecture
Programming: email to jonas.geiping@uni-siegen.de

Repetition

Let us recall a few basics from "Mathematik für Visual Computing".

A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is called *continuous at* $x_0 \in \mathbb{R}^n$ if for all $\epsilon > 0$ there exists a $\delta > 0$ such that all $x \in \mathbb{R}^n$ with $\|x - x_0\| \leq \delta$ meet $\|f(x) - f(x_0)\| \leq \epsilon$.

A function is $f : \mathbb{R}^n \to \mathbb{R}^m$ is called *continuous* if it is continuous at every $x_0 \in \mathbb{R}^n$.

The composition of continuous functions is a continuous function.

Let $E : \mathbb{R}^n \to \mathbb{R}$ be a function that is *partially differentiable*, i.e. a function for which $\frac{d}{du_i}E$ exists for every $i \in \{1, ..., n\}$. The *gradient of* E at u is given by

$$\nabla E(u) = (\frac{d}{du_1}E(u), \frac{d}{du_2}E(u), ..., \frac{d}{du_n}E(u))^T$$

and hence is a function $\nabla E : \mathbb{R}^n \to \mathbb{R}^n$. We call E continuously differentiable if ∇E exists and is continuous.

Now let $G : \mathbb{R}^m \to \mathbb{R}^n$ be another function. The composition of E and G is then $(E \circ G)(v) = E(G(v))$, mapping $\mathbb{R}^m \to \mathbb{R}$. The *gradient of* $E \circ G$ is, according to the *chain rule*,

$$\nabla E(G(v)) = DG(v)^T \, \nabla E(G(v))$$

where $DG(v)$ is the *Jacobian* of G at v, i.e. a matrix containing all partial derivatives of G in all dimensions:

$$DG_{ij}(v) = \frac{\partial d}{\partial dv_j}G_i(v).$$

As an example, if G is a linear operator, $G(v) = Av$, then $DG(v) = A$.

Theory

Exercise 1 (4 points). Let us denote the Huber-loss by

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

where

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

Show that

$$\begin{aligned} (H_\epsilon \circ D) : \mathbb{R}^n &\to \mathbb{R} \\ u &\mapsto H_\epsilon(Du) \end{aligned}$$

is continuously differentiable for D being a finite difference matrix $D \in \mathbb{R}^{2n \times n}$.

Programming

Exercise 2 (4 points). Use your implementation of the gradient descent algorithm with backtracking of the previous exercise sheet to implement the image denoising algorithm

$$\hat{u} = \text{argmin}_u \frac{1}{2}\|u - f\|_2^2 + \alpha H_\epsilon(Du)$$

where H_ϵ denotes the Huber-loss and 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 $u_{i,j,k} - u_{i,j-1,k}$.
Test your implementation with the egg image from the first exercise by reading the image and adding sufficient Gaussian noise with the 'imnoise' function. Compare your results to 'Salt&Pepper' noise of similar visual intensity.

Exercise 3 (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 $k \neq l$.