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 10

Room: H-C 6336

Friday, 12.01.2018, 14:15-15:45

Submission deadline: Thursday, 11.01.2018, 16:15

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

Theory: Linear Operations

Exercise 1 (Bonus exercise - 4 Points)**.** Consider the following optimization problem

$$\min_{X \in \mathbb{R}^{n \times m}} G(\mathrm{vec}(AX))$$

for a matrix $A \in \mathbb{R}^{p \times n}$ and a convex differentiable function $G : \mathbb{R}^{pm} \to \mathbb{R}$.
There are two ways to implement gradient descent for this:

$$X^{k+1} = X^k - \tau A^T \mathrm{mat}(\nabla G(\mathrm{vec}(AX^k)))$$

and

$$x^{k+1} = x^k - \tau (I \otimes A)^T \nabla G((I \otimes A) x^k)$$

First show that

$$\langle y, \mathrm{vec}(X) \rangle_{\mathbb{R}^{nm}} = \langle \mathrm{mat}(y), X \rangle_{\mathbb{R}^{n \times m}}$$

for all vectors $x \in \mathbb{R}^{nm}$ and matrices $Y \in \mathbb{R}^{n \times m}$ and then extend this to

$$\langle y, \mathrm{vec}(AX) \rangle_{\mathbb{R}^{pm}} = \langle A^T \mathrm{mat}(y), X \rangle_{\mathbb{R}^{n \times m}}.$$

Now show that both gradient descents are actually always equal with $\mathrm{vec}(X^k) = x^k$ and both minimize the given optimization problem.

Programming: Tomographic Inversion

Exercise 2 (Bonus Exercise - 12 Points: Radon inversion with TV regularization)**.** Download the data "radonInversion.zip" from the webpage and load the contained .mat file in MATLAB. The file contains sinogram data $\mathcal{R}(u)$ in the variable `measurementMatrix` as well as the measurement angles used to simulate the data.

- Visualize the sinogram data using imagesc.

- Create a sparse matrix R that computes the Radon transform $\mathcal{R}$ of a 16×16 pixel image using the angles stored in `measurementAngles`. What is the size of R and how many nonzeros does R have? Relate this number to the size of the input image and argue why storing R in matrix form is not feasible for reasonably sized images.
  Hint: You just need to know the Radon transformation of the canonical basis elements, which MATLAB's `radon` can do for you.

- Write function handles for applying R and R^* to an image for given angles. You will need to apply that the unfiltered backprojection is the adjoint operator. See MATLAB's 'iradon' and its parameters `'linear','none'`. Due to the way MATLAB's `iradon` is implemented you need to crop the (2:end-1,2:end-1) center part in order to actually get the adjoint operator.

- Implement the variational tomographic reconstruction of the provided data with a regularization.

$$E(u) = \frac{1}{2}||Ru - f||^2 + \alpha \text{Reg}(u)$$

  and find out what is hidden in the sinogram.
  The smoothed TV regularization usually works best for these tasks. *Hint: You need to be more patient than usual because applying `radon` is quite slow. Also: the output image has size 512 by 512 pixels.*

This exercise is a Christmas-bonus-exercise and gives points which do not count to the total number of points of which 50% have to be achieved.

Merry Christmas and a Happy New Year!