Variational Methods for Computer Vision: Exercise Sheet 4

Exercise: November 20, 2019

Part I: Theory

The following exercises should be **solved at home**. You do not have to hand in your solutions, however, writing it down will help you present your answer during the tutorials.

1. **Euler-Lagrange for one-dimensional functions with second spatial derivative.**
   Let $u \in C^2(\Omega; \mathbb{R})$ be a real-valued function and $\Omega \subset \mathbb{R}$. And let
   $$E(u) = \int_\Omega \mathcal{L}(u(x), u'(x), u''(x)) \, \mathrm{d}x$$
   be a real-valued Gâteaux differentiable functional which depends on:
   $$u(x), \qquad u'(x) = \frac{\mathrm{d}}{\mathrm{d}x} u(x), \qquad \text{and} \qquad u''(x) = \frac{\mathrm{d}^2}{\mathrm{d}x^2} u(x).$$
   Calculate the Gâteaux derivative of $E(u)$:
   $$\left. \frac{\mathrm{d}E(u)}{\mathrm{d}u} \right|_h$$
   for any differentiable direction h.

2. **Multidimensional integration by parts.**
   Prove the following identities for $\Omega \subset \mathbb{R}^n$, $f \in C^1(\mathbb{R}^n; \mathbb{R})$, $g \in C^1(\mathbb{R}^n; \mathbb{R}^n)$:
   (a) $\operatorname{div}(fg) = \langle \nabla f, g \rangle + f \operatorname{div} g$,
   (b) $\int_\Omega \langle \nabla f, g \rangle \, \mathrm{d}x = \int_{\partial\Omega} f \langle g, n \rangle \, \mathrm{d}s - \int_\Omega f \operatorname{div} g \, \mathrm{d}x$,
   where n is the unit vector normal to the boundary $\partial\Omega$. Hint for 2b: divergence theorem on exercise sheet 1.

3. **Euler-Lagrange for multidimensional functions.** Let $u \in C^1(\Omega; \mathbb{R})$ be a real-valued function and $\Omega \subset \mathbb{R}^3$. And let
   $$E(u) = \int_\Omega \mathcal{L}(u(x), \nabla u(x)) \, \mathrm{d}x$$
   be a real-valued Gâteaux differentiable functional. Calculate the Gâteaux derivative of $E(u)$. Hint: Use multidimensional integration by parts as derived in question 2b.

4. **Total variation.** Let $u \in C^1(\Omega; \mathbb{R})$ be a real-valued function with $\Omega \subset \mathbb{R}^2$. Derive the Euler-Lagrange equations of the following energies:
   (a) The *total variation* of the function u
   $$E_1(u) = \int_\Omega |\nabla u(x)| \, \mathrm{d}x = \int_\Omega \sqrt{\nabla u(x)^T \nabla u(x)} \, \mathrm{d}x.$$
   (b) The anisotropic total variation
   $$E_2(u) = \int_\Omega \sqrt{\nabla u(x)^T D(x) \nabla u(x)} \, \mathrm{d}x,$$
   where $D(x) \in \mathbb{R}^{2\times 2}$ is a real-valued matrix.

Part II: Practical Exercises

This exercise is to be solved **during the tutorial**.

1. **Variational image denoising.** In the lecture you encountered the continuous counterpart of the denoising energy from the previous exercise sheet

$$\frac{1}{2}\int_\Omega (u-f)^2 \,\mathrm{d}x + \frac{\lambda}{2}\int_\Omega |\nabla u(x)|^2 \,\mathrm{d}x,$$

where $\Omega \subset \mathbb{R}^2$ represents the image domain, $u : \Omega \to \mathbb{R}$ denotes the optimization variable and $f : \Omega \to \mathbb{R}$ stands for the input image. The corresponding Euler-Lagrange equation is given as

$$(u-f) - \lambda \Delta u = 0. \tag{1}$$

This Euler-Lagrange equation is linear in u. Unlike in the previous week, where we implemented Gauss Seidel to solve a similar linear equation system derived in a discrete setting, this week we solve a discretized version of (1) using Matlab's linear solver. We make use of Matlab's sparse matrices, since the size of this equation system is typically too large to construct with dense matrices. In the finite-dimensional and discrete setting, we consider the images as vectors $u \in \mathbb{R}^N$, $f \in \mathbb{R}^N$, i.e. we stack the two-dimensional images column wise.

   (a) Discretize the continuous gradient operator as $\nabla = (\partial_x^+ \;\; \partial_y^+)^\mathsf{T} \in \mathbb{R}^{2N\times N}$ using forward differences with Neumann boundary conditions, representing it as a sparse matrix (`help spdiags`, `help speye`).

   (b) Solve the linear system arising from (1) using the backslash command (`help \`). Note that the divergence is the negative adjoint of the gradient, thus $\Delta = \operatorname{div}\nabla = -\nabla^\mathsf{T}\nabla$.

   (c) Investigate how fast Matlab's sparse solver is by experimenting with different input image sizes (`help imresize`). In order to see what the backslash operator does internally, you can enable detailed output with `spparms('spumoni', 2);`.

2. **Diffusion with implicit time-discretization.** Equipped with sparse image gradient, revisit the implementation of diffusion from tutorial 2 and add the implicit time-discretization scheme discussed in the lecture (Chapter 2). Investigate larger time step sizes.