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 7

Room: H-C 7326

Wednesday, 07.11.2016, 14:15-15:45

Submission deadline: Monday, 05.11.2016, 16:00 in the mailbox

Programming: email to jonas.geiping@uni-siegen.de

Theory

Exercise 1 (4 Points: Convex Relaxation Continued)**.** In some cases one desires to find binary solutions, i.e. images u with $u(x) \in \{0,1\}$ for all x. We will get to know several such example once we start talking about image segmentation.

To constrain the solution u to be binary, consider the energy

$$\delta_{\{0,1\}}(u) = \begin{cases} 0 & \text{if } u \in \{0,1\}, \\ \infty & \text{otherwise.} \end{cases}$$

Show that $\delta_{\{0,1\}}$ is not convex and compute the biconjugate $\delta^{**}_{\{0,1\}}$, i.e. the largest convex underapproximation of $\delta_{\{0,1\}}$.

Hint: To determine the first conjugate, distinguish the cases of the input argument p being negative or non-negative. For the second conjugate, distinguish the cases of the input argument u being negative, being greater than 1, and being an element of $[0,1]$.

Exercise 2 (4 Points: Poisson Editing Results in a Linear System)**.** Consider the Poisson editing approach discussed in lecture, i.e.,

$$\min_u \|\nabla u - \nabla g\|_2^2 \qquad \text{subject to } u(x) = f(x) \; \forall x \notin M,$$

for some inpainting domain M inside image f and a new image g that is going to be edited into f.

Show that the discrete version of the above problem reduces to solving a linear system.

Hint: Define a diagonal matrix D_M such that $(D_M u)_i = 0$ if $i \notin M$, and $(D_M u)_i = u_i$ otherwise. Can you incorporate the constraints by writing u as the sum of two parts which are multiplied with D_M and $I - D_M$ respectively?

Programming

Exercise 3 (4 Points)**.** The goal of this exercise is to implement Poisson image editing based on your solution of exercise 2.

1. Find two fun images to fuse; if you don't know where to start, *www.pixabay.com* is a data base with many free-to-use images. Be aware that not all every image can be fused by Poisson editing, try some and see what works well.

2. Write a routine with which the user can interactively select a region in one image (e.g. using *roipoly*) and allow the user to select a location for this object in the second image.

3. Determine the inpainting domain M from the previous input, solve the linear system as computed in exercise 2, and display the final fused image.

Challenge: Be creative and create a fun fusion result! The results will be shown in the lecture. The funniest and most creative submission will receive a chocolatey award!