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 4

Room: H-C 6336

Friday, 17.11.2017, 14:15-15:45

Submission deadline: Tuesday, 14.11.2017, 14:15 in Room H-C 6336

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

Theory

Exercise 1 (2 Points)**.** Let

$$W = \begin{pmatrix} A & B \\ B^T & B^T A^{-1} B \end{pmatrix}$$

for $W \in \mathbb{R}^{N\times N}$, $A \in \mathbb{R}^{r\times r}$, and $B \in \mathbb{R}^{r\times N-r}$ symmetric matrices. Prove that $\text{rank}(W) \leq r$ by showing the identity

$$W = \bar{U}\ \Sigma\ \bar{U}^T, \qquad \text{for} \quad \bar{U} = \begin{pmatrix} U \\ B^T U \Sigma^{-1} \end{pmatrix} \in \mathbb{R}^{N\times r}$$

where $A = U\Sigma U^T$ is the eigendecomposition of A, and deduce from this identity, that the rank is r or less.

It can be shown that this W with rank$(W) \leq r$ has a (reduced) eigendecomposition $W = VDV^T$ for some $V \in \mathbb{R}^{N\times r}$ and $D \in \mathbb{R}^{r\times r}$ being diagonal [1], but you can skip this step.

Exercise 2 (4 points)**.** Let $W \in \mathbb{R}^{N\times N}$ be a symmetric rank r matrix with eigendecomposition $W = VDV^T$, $D_{ii} \leq 1\ \forall i \in \{1, \dots, n\}$. Show that the solution to

$$\hat{u} = \arg\min_u \frac{1}{2}\|u - f\|^2 + \frac{\alpha}{2}\langle u, (I - W)u\rangle$$

is $\hat{u} = VXV^T f$, where $X \in \mathbb{R}^{r\times r}$ is a diagonal matrix with entries $X_{ii} = \frac{1}{1+\alpha-\alpha D_{ii}}$.
Start with the optimality condition and insert what you know about W.
How would you implement this formula (pseudocode) without using a full matrix-vector product in dimensions $n \times n$.

[1] See Fowlkes et al. Spectral Grouping Using the Nyström method, (2004)

Understanding Code

Exercise 3 (4 points)**.** Download the functions "IntegralNLM.m", "IntegralImage.m", and "ImShift.m" from supplementary material for sheet 4. Answer the following questions:

- What does "ImShift.m" do?
- What does "IntegralImage.m" do?
- State a mathematical formula for the (i, j)-th entry of the variable *ssd* in IntegralNLM.m.
- Based on *ssd*, state a formula for *PatchDist*.

Programming

Exercise 4 (6 points)**.** Load the image *cityscape* provided in the supplementary material for sheet 4 and create a noisy version by adding Gaussian noise (`help randn`). Use the downloaded functions from the previous exercise to generate a similarity matrix W of the noisy image.

- Perform nonlocal means denoising with the similarity matrix W
- Compute a diagonal matrix D with

  $$d_i = \sum_j W_{i,j}$$

  on the diagonal. Perform nonlocal regularization with

  $$R(u) = \frac{\alpha}{2}\langle u, Lu \rangle$$

  for $L = D - W$ and for $L = I - D^{-1/2} W D^{-1/2}$. Tune the values of α.
- Check the *PSNR* [2], (`psnr(image,reference_image)` in Matlab) of your output versus the reference image (in our case the original image). Compare this value to the PSNR value for your result from the previous exercise (Huber-TV denoising).
  Which denoising algorithm yields the best results?

[2]PSNR, the peak signal to noise ratio measures the distance of two images in a logarithmic scale, if the images are very similar, then their PSNR will be high.