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 5

Room: H-C 6336

Friday, 24.11.2017, 14:15-15:45

Submission deadline: Tuesday, 21.11.2017, 14:15 in Room H-C 6336

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

Theory

Exercise 1 (2 Points)**.** Let $g \in \mathbb{R}^n$ be a discrete signal. Its *discrete Fourier transform* is given by

$$\mathcal{F}(g)(k) = G_k = \sum_{l=0}^{n-1} g_l \exp(-2\pi ikl/n).$$

We can evaluate this transformation for any $k \in \mathbb{Z}$, but it is n-periodic, $G_{k+N} = G_k$. We will first compute a few properties of this transformation:

- Verify that the Fourier transform $\mathcal{F}(g)$ is a linear transformation and show the n- periodicity of the Fourier transform.
- What would the coefficients of a matrix $F \in \mathbb{C}^{n \times n}$ be, so that Fg equals the n distinct Fourier coefficients of f?

Exercise 2 (4 Points)**.** Let $f, g \in \mathbb{R}^n$ be a discrete signal.

$$F_k = \sum_{l=0}^{n-1} f_l \exp(-2\pi ikl/n), \quad G_k = \sum_{m=0}^{n-1} g_m \exp(-2\pi ikm/n), \quad k \in \mathbb{Z}$$

denote the coefficients of the discrete Fourier transforms of f and g respectively.

Consider the discrete circular convolution with coefficients

$$c_k := \sum_{l=0}^{n-1} f_l \; g_{(k-l)_n},$$

where $g_{((k-l))_n}$ denotes the n-periodic extension of g (so $(k-l) \bmod n$ if you think back to your discrete mathematics lectures.)

Prove that the discrete circular convolution theorem

$$C_k = F_k \cdot G_k$$

holds, where C_k is the Fourier transform of the circular convolution of f and g.

Exercise 3 (2 Points)**.** For a linear transformation $T : \mathbb{R}^n \to \mathbb{R}^n$ it can be cumbersome to compute the matrix representation of T just to compute the adjoint matrix. If we define the adjoint operation T^* of a linear transformation more generally by the formula

$$\langle T(x), y\rangle = \langle x, T^*(y)\rangle \quad \forall x, y \in \mathbb{R}^n \tag{1}$$

we can compute it analytically from T.

Compute the adjoint operation to the circular convolution from the previous exercise.

Hint: You should end up with a formula for a transformation T^*, so that

$$\langle T_g(x), y\rangle = \sum_{k=0}^{n-1}\sum_{l=0}^{n-1} x_l g_{(k-l)_n} y_k = \sum_{k=0}^{n-1} [T_g^*(y)]_k \; x_k = \langle x, T_g^*(y)\rangle \quad \forall x, y \in \mathbb{R}^n$$

if $T_g(x)$ is the circular convolution of signals x and g.

Programming

Exercise 4 (4 Points)**.** Using the discrete convolution theorem, we can generate a blurry image by

$$\hat{g} = \texttt{fft}^{-1}(\texttt{fft}(f) \cdot \hat{k})$$

with k being a suitable blur kernel, for example `fspecial('gaussian',5,2)` and f an image, e.g. *peppers*. $\hat{k}$ is given by $\max(\texttt{fft}(k), 1\text{e}{-6})$. `fft` denotes a fast implementation of the discrete Fourier transform and $\texttt{fft}^{-1}$ its corresponding inverse transform, denoted for two dimensions by `fft2` and `ifft2` in Matlab. Remember to use the padding parameters of the `fft`.

Verify that

$$\hat{u} = \texttt{fft}^{-1}(\texttt{fft}(\hat{g})/\hat{k})$$

returns the true image without a blur.

Now compute

$$\hat{u} = \texttt{fft}^{-1}\left(\texttt{fft}(f + \sigma \cdot \text{noise})/\hat{k})\right)$$

for a tiny $\sigma > 0$. What do you observe?

Exercise 5 (4 Points)**.** Implement

$$\min_u \frac{1}{2}\|Au - f\|^2 + \alpha R(u)$$

for a regularization R of your choice from the previous exercises (Huber-TV, nonlocal regularization) and A being the matrix representation of the blur kernel, given by `convmtx2`. Generate data f by $A * u$, blurring the vectorized image u, and add noise. Are you able to stabilize the inverse process?

Exercise 6 (4 Bonus Points)**.** Repeat exercise 5, using the circular convolution operation directly and its adjoint operation as computed in exercise 3.