Exercises Inverse Problems, SS 2010

Exercise Sheet 1 (March 16, 2010)

Markus Haltmeier

Exercise 1 (Discrete Convolution). A discrete image of size $n_1 \times n_2$ is a family

$$u := \big(u(i_1, i_2)\big)_{(i_1,i_2)\in I} \in \mathbb{R}^{n_1\times n_2}\,,$$

where $I = \{1,\dots,n_1\} \times \{1,\dots,n_2\}$. The discrete convolution $u * v \in \mathbb{R}^{n_1\times n_2}$ of two images $u \in \mathbb{R}^{n_1\times n_2}$ and $v \in \mathbb{R}^{m_1\times m_2}$ is defined by

$$(u * v)(i_1, i_2) := \sum_{j_1=1}^{m_1}\sum_{j_2=1}^{m_2} u\left(i_1 - j_1 + \frac{m_1+1}{2}, i_2 - j_2 + \frac{m_2+1}{2}\right) v(j_1, j_2)\,. \tag{1}$$

Here m_1, m_2 are assumed to be odd numbers, and .

(a) Illustrate the definition of the discrete convolution (for simplicity take $n_2 = m_2 = 1$).

(b) Is the discrete convolution as defined in (1) symmetric, associative, and/or distributive (with respect to addition)?

(c) Write a Matlab function `conv_same(u,v)` that computes the discrete convolution as defined in (1).

(d) Test the function `conv_same(u,v)` with

`u = double(imread('lena512.bmp'))` and `v = [-1;0;1]`.

Exercise 2 (Gauss Filter). A basic method for denoising an image u is to compute the discrete convolution with the Gaussian kernel $k_{\sigma,m} \in \mathbb{R}^{m\times m}$,

$$k_{\sigma,m}(j_1, j_2) := \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(j_1 - (m+1)/2)^2 + (j_2 - (m+1)/2)^2}{2\sigma^2}\right)\,, \qquad \text{for } (j_1, j_2) \in \{1,\dots m\}\,.$$

Write a Matlab function

`filt_gauss(u,sig,m)`

that convolves an image u with the Gaussian kernel.

Exercise 3 (Edge Detection). One widely used method for edge detection is to appyl the Sobel operator:

$$G : u \mapsto \sqrt{(u * G_x)^2 + (u * G_y)^2}\,,$$

where

$$G_x := \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}, \quad G_y := \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}\,.$$

Write a Matlab function

`edge_sobel(u)`

that applies the Sobel operator to an image u. Test `edge_sobel(u)` with

`u = double(imread('lena512.bmp'))`.