Technische Universität München
Fakultät für Informatik
PD Dr. Slobodan Ilic

WS 16/17
Exercise Sheet 1
October 27, 2016

Exercises in Tracking & Detection

Exercise 1 Convolution

In the lecture you learned about convolution. Given an image I and a convolution mask H, you obtain a convolved image J with the following formula:

$$J(x,y) = H * I = \sum_{i=-\frac{M}{2}}^{\frac{M}{2}} \sum_{j=-\frac{N}{2}}^{\frac{N}{2}} H(i,j)I(x-i,y-j) \; . \tag{1}$$

Since the dimensions of the mask are usually larger than 1, the resulting image J would shrink by $2\left\lfloor\frac{M}{2}\right\rfloor$ in x and by $2\left\lfloor\frac{N}{2}\right\rfloor$ in y direction compared to the original image (assuming that the mask dimensions are odd and the convolution center is in the middle of the mask). To avoid that, we use a special border treatment. One option is to *mirror* the pixel intensities across the border. Another one is to set the pixels outside the image to the intensity of the corresponding border pixel.

a) Write a generic Matlab function that takes an image and a given general $M \times N$ mask and returns the convolved image. The function should also have a parameter with which you can select the border treatment. Implement the two border treatments mentioned above.

b) Apply the mean filter from Eq. 2 to an image of your choice.

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{2}$$

Exercise 2 Separability of Filters

In the lecture you have seen one special filter (see Eq. 3):

$$G(u,v) = \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2}\frac{u^2+v^2}{\sigma^2}} \tag{3}$$

This filter is called the Gaussian filter. The Gaussian filter is often used for removing noise within the image by smoothing it. The Gaussian filter can be separated into one horizontal and one vertical Gaussian filter (see Eq. 4).

$$G(u) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\frac{u^2}{\sigma^2}} \tag{4}$$

There are several aspects of this separability that we want to investigate in this exercise.

a) Implement a method that generates a 2D Gaussian filter with respect to a parameter σ. Apply the Gaussian masks with $\sigma = 1.0$ and $\sigma = 3.0$ to an image of your choice with the filter function from the previous exercise. The mask should have the size $3\sigma \times 3\sigma$. What do you conclude?

b) Implement a method that generates a horizontal and a vertical 1D Gaussian filter with respect to a parameter σ. Apply the horizontal Gaussian mask and the vertical Gaussian mask with $\sigma = 1.0$ and $\sigma = 3.0$ successively to an image of your choice. The masks should have the sizes $1 \times 3\sigma$ and $3\sigma \times 1$. Compare the result of the outcome of the 2D Gaussian filtered image with the result of the outcome of the 1D Gaussian filtered image by computing the sum of squared differences for $\sigma = 1.0$ and $\sigma = 3.0$ respectively. What do you conclude?

c) Compare the time (Matlab provides *tic* and *toc* for time measurements) of the Gaussian filtering made with one 2D filter and the one made with two 1D filters. What do you conclude?

Exercise 3 Image Gradients

In this exercise we compute the image gradients using a convolution and visualize the magnitude and orientation of the gradients.

a) First, use the basic derivative filters from the lecture to compute the gradient images in x- and y-direction. The basic derivative filters are:

$$D_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \quad D_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}. \tag{5}$$

These filter masks are applied separately to the original image, such that we get two seperate gradient images as result.

b) From these two gradient images we now compute the gradient magnitudes and orientations using:

$$|\nabla I(x,y)| = \sqrt{(D_x * I)^2 + (D_y * I)^2}, \tag{6}$$

$$\psi(\nabla I) = \arctan\left(\frac{D_y * I}{D_x * I}\right) \tag{7}$$

Visualize the gradient magnitudes and orientations using gray value images.

c) Finally, we want to reduce the noise in the image before computing the gradients:

$$I_x = D_x * (G_x * I), \tag{8}$$

$$I_y = D_y * (G_y * I). \tag{9}$$

But, instead of processing the input image twice we make use of the associativity law. For this, first filter the Gaussian convolution mask using the basic derivative filters and then convolve the image with the resulting mask:

$$I_x = (D_x * G_x) * I, \tag{10}$$

$$I_y = (D_y * G_y) * I. \tag{11}$$