

# Variational Methods for Computer Vision: Exercise Sheet 1

---

Exercise: 4 November 2015

---

## Part I: Theory

The following exercises should be **solved at home**. You do not have to hand in your solutions, however, writing it down will help you present your answer during the tutorials.

1. Let  $X$  be an arbitrary set. A metric on  $X$  is a function

$$d: X \times X \rightarrow \mathbb{R} \tag{1}$$

satisfying for all  $x, y, z$  in  $X$  the following conditions:

$$d(x, y) = 0 \Leftrightarrow x = y \quad (\text{identity of indiscernibles.})$$

$$d(x, y) = d(y, x) \quad (\text{symmetry})$$

$$d(x, y) \leq d(x, z) + d(z, y) \quad (\text{triangle inequality})$$

Verify whether or not the following functions are metrics:

- (a) The Euclidean distance:  $D(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} \forall x, y \in \mathbb{R}^2$ .
  - (b) The so called Manhattan distance:  $D(x, y) = |(x_1 - y_1)| + |(x_2 - y_2)| \forall x, y \in \mathbb{R}^2$ .
  - (c)  $D(x, y) = |(x^2 - y^2)| \forall x, y \in \mathbb{R}$ .
2. Let  $f \in C^n(\mathbb{R})$  and  $g \in C^m(\mathbb{R})$  be real valued continuously differentiable mappings. Consider the convolution of these two functions:

$$(f * g)(x) = \int_{\Omega} f(y)g(x - y) dy$$

Let  $f \in L^1(\mathbb{R})$  and  $g \in L^1(\mathbb{R})$  be two absolutely integrable functions and let  $\mathcal{F}$  denote the Fourier transform operator:

$$\mathcal{F}\{f\} := \hat{f}(\nu) = \int f(x)e^{-2\pi i x \cdot \nu} dx.$$

Prove that the Fourier transform of the convolution of two functions is the same as the the point-wise multiplication of the respective Fourier transforms hence:

$$\mathcal{F}\{f * g\} = \mathcal{F}\{f\} \cdot \mathcal{F}\{g\}.$$

What implications does this have for computing the convolution?

## Part II: Practical Exercises

This exercise is to be solved **during the tutorial**.

1. Download the archive `vmcv_ex01.zip` and unzip it on your home folder. In there should be a file named `coins.png`.
2. Start Matlab and load the unzipped image using the following command:

```
f=double(imread('coins.png'))
```

Show the image using Matlab's command:

```
figure; imshow(uint8(f));
```

Also familiarize yourself with the following commands:

```
for, size, zeros, ones and matrix indexing
```

using Matlab's great documentation or the Tutorials listed below.

3. Compute a convolution of the image with the Gaussian kernel given in the theoretical exercise 2.b. In theory, the Gaussian distribution is non zeros everywhere, however in practice we restrict ourself to truncated kernels defined on so called masks. The discrete convolution is given as follows:

$$g(i, j) = (w * f)(i, j) := \sum_{m, n} w(m, n) f(i - m, j - n)$$

The discrete truncated Gaussian kernel can be written as follows:

$$w(m, n) = \exp\left(-\frac{m^2 + n^2}{2\sigma^2}\right)$$

In order to stay consistent with the continuous formulation of the Gaussian distribution make sure to normalize the kernel function such that the following holds:

$$\sum_{m, n} w(m, n) = 1.$$

An example of a 3x3 mask is the following:

$w(-1, 1)$	$w(-1, 0)$	$w(-1, -1)$
$w(0, 1)$	$w(0, 0)$	$w(0, -1)$
$w(1, 1)$	$w(1, 0)$	$w(1, -1)$

For simplicity ignore pixels where the convolution filter goes beyond the edge of the image.

4. Let  $W$  and  $H$  denote respectively the length and width of the input image  $f$ . Compute the the gradient  $\nabla f = (\partial_x f, \partial_y f)^T$  of the image using the discretization scheme of forward differences:

$$(\partial_x f)_{i,j} = \begin{cases} f_{i+1,j} - f_{i,j} & \text{if } i < H \\ 0 & i = H. \end{cases}$$

$$(\partial_y f)_{i,j} = \begin{cases} f_{i,j+1} - f_{i,j} & \text{if } j < W \\ 0 & j = W. \end{cases}$$

Notice that the boundary values of the gradient are set to zero.

5. Try solving exercise 4 by avoiding using any for loops this time. Can you tell the difference?
6. Let  $f_\sigma$  be the input image convolved with a Gaussian kernel of standard deviation  $\sigma$ . Compute the magnitude of the Gradient  $|\nabla f_\sigma|$  for different values for  $\sigma$ . What do you observe?

**Matlab-Tutorials:**

<http://www.math.utah.edu/lab/ms/matlab/matlab.html>

<http://www.math.ufl.edu/help/matlab-tutorial/>

<http://www.glue.umd.edu/~nsw/ench250/matlab.htm>