# TDT4195: Visual Computing Fundamentals

**Digital Image Processing - Assignment 1**

August 29, 2016

Aleksander Rognhaugen
Department of Computer and Information Science
Norwegian University of Science and Technology (NTNU)

- **Delivery deadline: September 9, 2016** by 22:00.
- **This assignment counts towards 3 % of your final grade.**
- You can work on your own or in groups of two people.
- Deliver your solution on *itslearning* before the deadline.
- Please upload your report as a PDF file, and package your code into an archive (e.g. zip, rar, tar).
- The programming tasks may be completed in the programming language of your choice, however, it might be a good idea to select one that supports matrix and image processing operations, e.g. MATLAB or Python with NumPy. *The lab computers at IT-S 015 support Python and MATLAB.*
- Your code is part of your delivery, so please make sure that your code is well-documented and as readable as possible.
- For each programming task you need to give a brief explanation of what you did, answer any questions in the task text, and show any results, e.g. images, in the report.

**Learning Objectives:** Gain experience with (a) how to process images on a computer by using basic intensity transformations, and (b) how spatial filtering works by implementing two-dimensional convolution.

## 1 Theory [0.5 points]

1. [**0.1 points**] How can you see that an image has low contrast when looking at its histogram?
2. [**0.1 points**] Histogram equalization is an algorithm that creates a transformation $\mathcal{T}_{heq}$ by exploiting an image histogram. What effect does $\mathcal{T}_{heq}$ have when applied on an image? What happens when $\mathcal{T}_{heq}$ is applied multiple times on the same image?

3. [**0.1 points**] Perform histogram equalization by hand on the 3-bit (8 intensity levels) $3 \times 5$ image in Figure 1 (a). Your report must include all the steps you did to compute the transformation, the transformed image, and its histogram.

4. [**0.1 points**] What is the main difference between convolution and correlation? Could you mention one or more kernels that yield the same result for both convolution and correlation? What is common to these kernels?

5. [**0.1 points**] Perform spatial convolution by hand on the 3-bit image in Figure 1 (a) using the Laplacian kernel in Figure 1 (b). State how you handled the boundaries of the image. The convolved image should be $3 \times 5$.

| | | | | |
|---|---|---|---|---|
| 0 | 5 | 6 | 3 | 3 |
| 4 | 7 | 4 | 6 | 4 |
| 4 | 5 | 3 | 5 | 4 |

(a) A $3 \times 5$ image.

| | | |
|---|---|---|
| 0 | 1 | 0 |
| 1 | -4 | 1 |
| 0 | 1 | 0 |

(b) A $3 \times 3$ convolution kernel.

Figure 1: An image $I$ with intensities in the $[0, 7]$ range (3-bit) and a convolution kernel approximating the Laplacian $\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$. For the image, each square represents a pixel, where the number within is the intensity value.

# 2 Programming [2 points]

## Task 1: Greyscale Conversion [0.5 points]

Converting a colour image to a greyscale representation can be done by taking the average of the red (R), green (G), and blue (B) channels, as seen on the left-hand side of Equation 1. However, because different colours have different wavelengths it follows that each colour has a slightly different contribution to the image. Thus, to preserve the luminance, or lightness, of the original colour image a weighted average is typically used instead. One such weighted average – used by the sRGB colour space – can be seen on the right-hand side of Equation 1.[1]

$$\text{grey}_{i,j} = \frac{\text{R}_{i,j} + \text{G}_{i,j} + \text{B}_{i,j}}{3} \qquad \text{grey}_{i,j} = 0.2126\text{R}_{i,j} + 0.7152\text{G}_{i,j} + 0.0722\text{B}_{i,j} \tag{1}$$

a) [**0.5 points**] Implement *two* functions that manually convert a RGB colour image to a greyscale representation using the two methods outlined in Equation 1. Compare

[1] More information can be found here: `https://en.wikipedia.org/wiki/Greyscale`.

the output of the two functions on a colour image and show the result in your report. Briefly discuss the differences between the two outputs.

### Task 2: Intensity Transformations [0.5 points]

A greyscale intensity, or pixel brightness, transformation $\mathcal{T}$ maps the intensity values $p \in [p_0, p_k]$ in the original image into a new intensity range $q \in [q_0, q_k]$:

$$q = \mathcal{T}(p) \Leftrightarrow J_{i,j} = \mathcal{T}(I_{i,j}) \tag{2}$$

a) **[0.2 points]** Implement a function that takes a greyscale image and applies the following intensity transformation: $\mathcal{T}(p) = 255 - p$. What would you call this transformation? Apply the function on an image and show the result in your report. *Tip: If you have normalised the image between* $[0, 1]$*, then the transformation must be changed to* $\mathcal{T}(p) = 1.0 - p$*.*

A basic definition of the gamma transformation can be seen in Equation 3. It adjusts the overall intensity of an image by squeezing pixel intensities to either low or high intensities.

$$\mathcal{T}(p) = cp^{\gamma}, \qquad c > 0 \land \gamma > 0 \tag{3}$$

b) **[0.3 points]** Implement a function that takes a greyscale image and applies the gamma transformation from Equation 3. You can let $c = 1$. Test out different $\gamma$-values on a greyscale image and show the result in your report. What happens with the image intensity values when $\gamma > 1$ and $\gamma < 1$? *Tip: For this task, it is important that you normalise the image between* $[0, 1]$ *before applying the gamma transformation.*[2]

### Task 3: Spatial Convolution [1 point]

Convolution $(f * h)$ is an operation on two functions $f$ and $h$. When $f$ is defined on a spatial variable, the operation is called spatial convolution. Formally, for functions $f(x, y)$ and $h(x, y)$ of two discrete variables $x$ and $y$, convolution is defined as:

$$(f * h)(x, y) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} f(i, j)h(x - i, y - j) \tag{4}$$

In Equation 4, $*$ is the convolution operator, $h$ is the convolution filter/kernel/mask, and, for our purpose, $f$ is an image. Following the equation, we slide (or convolve) a kernel across the width and height of the input image; as the convolution kernel moves across the image in scanline, an output image is created. Intuitively, the centre of the kernel is placed on a pixel, and a linear combination of the local neighbourhood and the convolution kernel is determined. The linear combination is evaluated and the result is assigned to the output image. This operation is repeated for every pixel in the input image.

[2] Assuming we have an 8-bit image, with intensity values between 0 and 255, all we have to do is change the data type to `float` and divide by 255.

a) **[0.4 points]** Implement a function that takes a greyscale image and an arbitrary linear convolution kernel and performs 2D spatial convolution. Assume that the size of the input convolution kernel is odd numbered, e.g. $3 \times 3$, $5 \times 5$, or $7 \times 7$. You *must* implement the convolution procedure yourself from scratch. You are not required to implement procedures for adding and removing padding.

Equation 5 shows two convolution kernels that are used for smoothing images. The kernel on the left is a $3 \times 3$ averaging convolution kernel, while on the right is a $5 \times 5$ Gaussian convolution kernel approximated using binomial coefficients.

$$h_a = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad h_g = \frac{1}{256}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix} \tag{5}$$

b) **[0.2 points]** Test out the convolution function you made in task (a) by convolving a colour image with the smoothing kernels in Equation 5.[3] Show the smoothed images in your report. *Tip: To convolve a colour image, convolve each channel separately and layer them afterwards.*

The Sobel operator is a directional convolution kernel that approximates the image gradient in the horizontal and vertical direction. The convolution kernels can be seen in Equation 6.

$$s_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \qquad s_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{6}$$

c) **[0.4 points]** Use the convolution kernels in Equation 6 to approximate the horizontal $I_x$ and vertical $I_y$ gradient of a greyscale image $I$. Calculate the magnitude of the gradients $|\nabla| = \sqrt{I_x^2 + I_y^2}$ and show the result in your report. What does $|\nabla|$ tell us about the image? *Tip: Make sure your spatial filtering function outputs images with signed numbers.*

## 3 Exam Style Q&A [0.5 points]

Create *one* "exam" style question and answer with the following theme: *Spatial domain.* It should be possible to answer the question in 5-7 minutes. The answer *must* contain all the

[3] Remember that if you want to try 2D Gaussian kernels with different sizes you can easily generate one by taking the outer product of two 1D Gaussian kernels. For example, let $A = \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$, then $\frac{1}{16} A \otimes A$ is a $3 \times 3$ Gaussian convolution kernel.

steps required to get to a solution, and not just the answer itself. Challenging questions are more likely to receive the full amount of points.

You will be asked to add your question and answer – corrected based on any feedback – to a separate document later in the semester.