Assignment 3

Welcome to the third assignment of the lecture *Visual Computing* in summer term 2020. **Please read all instructions carefully!** The goal of this exercise is to familiarize you with some technical aspects of digital cameras, light and colors as well as image filtering. Submission is due on Wednesday, May 5th, 2021 at 8pm. Please submit your solutions via `read.mi.hs-rm.de`.

Aufgabe 1 (1 point)**.** Using the simple one-dimensional example $\mathbf{I}^E = [1\ 2\ 1]$ and $\mathbf{I}^{F_1} = \mathbf{I}^{F_2} = [-1\ 0\ 1]$, show that the cross-correlation is not associative.

Aufgabe 2 (2 points)**.** Calculate the convolution between the filter matrix $\mathbf{I}^F$ and the image $\mathbf{I}^E$.

$$\mathbf{I}^F = \begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 1 & 0 & 1 \\ \hline 0 & 0 & 0 \\ \hline \end{array} \quad \text{and} \quad \mathbf{I}^E = \begin{array}{|c|c|c|c|}\hline 0 & 1 & 1 & 1 \\ \hline 1 & 1 & 0 & 1 \\ \hline 0 & 0 & 1 & 1 \\ \hline 1 & 1 & 0 & 0 \\ \hline \end{array} \quad \text{(periodically continued)}$$

Aufgabe 3 (4 points)**.** Determine the unknown color components of the camera sensor with Bayer mosaic shown below using bilinear interpolation.

R = 40	G = 80	R = 60	G = 100
G = 80	r = ? g = ? B = 100	r = ? G = 100 b = ?	B = 40
R = 40	r = ? G = 100 b = ?	R = 100 g = ? b = ?	G = 25
G = 25	B = 20	G = 75	B = 20

Aufgabe 4 (1 point)**.** Calculate the H value of the RGB color vector $(1, 1/2, 0)^\top$ (this corresponds to the color orange).

Aufgabe 5 (1 point)**.** Specify the RGB color tuple of the brightest color (color with the highest luminance) to the rg color tuple $(1/2, 1/4)^T$. Which colour does this correspond to?

Aufgabe 6 (3 points)**.** What is the distance between the two pixels with location coordinates $(17, 42)$ and $(289, 68)$

- using the 4-way neighborhood?
- using the 8-way neighborhood?

Also compare the results with the Euclidean distance between the two pixels.

Aufgabe 7 (10 points)**.** Write a simple Python program that reads an image and performs the following functions:

1. (4 points) *Low-pass filtering of an image*: It should be possible to apply the OpenCV filters
   (a) *boxFilter(...)*,
   (b) *GaussianBlur(...)*,
   (c) *bilateralFilter(...)* und
   (d) *medianBlur(...)*

   to an image. The parameters of the individual filters should be adjustable via the GUI using *sliders*.
2. (3 points) *High-pass filtering of an image*: It should be possible to apply a *Sobel-* or *Sharr filter* to an image. The following should be visualized
   (a) the image showing the gradients in x direction,
   (b) the image showing the gradients in y direction,
   (c) the image showing the length of the gradients.
3. (3 points) *Canny edge filter* (*cv2.Canny(...)*). *Compare the result* with the image of the gradient lengths determined using the Sobel or Scharr filter.