Mathematisch-Naturwissenschaftliche
Fakultät

Fachbereich Informatik
Arbeitsbereich Visual Computing

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Bildverarbeitung I (Prof. Schilling)

WS 2022/2023

Assignment 1

Remarks

Please submit your exercises in ILIAS before 23:55 on the closing date. *Each* member of the group must be able to explain *each* exercise. Groups and members will be chosen at random and asked to present an exercise as a representative of the whole group. You should be prepared to explain any exercise at our biweekly tutorial.

Python

- **You should avoid the use of unnecessary loops in Python since the processing time will increase drastically. Instead, use functions that work on numpy arrays directly.**
- **Use the script file that is provided on ILIAS. Use the exact filenames for your functions and scripts that are specified in the framework.**
- **Please put the answers to comprehension questions into a separate PDF file.**
- **If there is a built-in function for an algorithm you are supposed to implement, do *not* use it.**

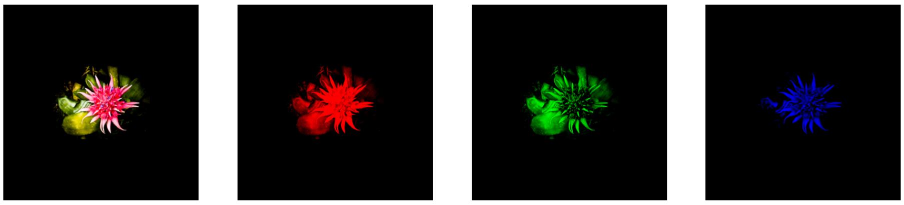
How to work on these exercises: The programming assignments in this course consist of multiple exercise files named `exercise_XX.py`. These files contain some incomplete definitions of python functions which you should complete. Running an exercise script with `python exercise_XX.py` (where `XX` refers to the two-digit exercise number) from your anaconda environment promts the results of your implementation. There are also some test cases logged to the console to provide feedback to you. Please ensure that your implementation passes all test cases. Note that we will grade your submissions using further test cases with multiple different inputs.

Submission: Create a folder `lastname1_..._lastnameN` with your team members lastnames in alphabetical order. Copy your `exercise_XX.py`-files into this folder. Create a zip file `lastname1_..._lastnameN.zip` from the folder. Submit the zip-file to Ilias.

Exercise 1: Working with images [4 points]

This exercise teaches some fundamentals about how images can be represented and manipulated in a computer using `python` with the libraries `numpy`, `matplotlib` and `pillow`. Complete the missing code in `exercise_01.py` to solve the following subtasks.

a) Complete the function `rgb_split` that receives an input image and returns a tuple containing its R, G and B channels as separate images. Each channel should be represented in its primary color like shown in the figure below.



b) Implement a gamma correction for images (`gamma_correction`).

c) Fill in the function `rgb_to_gray` that converts an rgb-image to grayscale. Do not weight all colors with the same weight, use the predefined array `weights`. It already provides weighting factors proportional to the human eye's color sensivity.

Exercise 2: Histograms [6 points]

Add the missing functionalities to the script `exercise_02.py` analogous to the previous exercise.

a) Create Histograms [2 points]: Implement the function `get_hist` which computes the histogram of a grayscale image. Do not use the numpy function `np.histogram`.
What can be read out of the histogram — in general as well as for the given image?

b) Contrast stretching [1 point]: Implement the function `max_contrast` that maximizes the contrast of an image. How does the histogram change? How does it affect the image?

c) Histogram Equalization [3 points]: Calculate a normalized accumulated histogram (`accumulate_hist`). Implement histogram equalization (`equalize_hist`) that receives the grayscale image and the accumulated histogram as parameters.
Discuss the result.