Mathematisch- Naturwissenschaftliche
Fakultät

Fachbereich Informatik
Arbeitsbereich Visual Computing

Eberhard Karls
Universität
Tübingen

Bildverarbeitung I (Prof. Schilling)

WS2017/2018

Assignment 1, Due: November 2th, 2017

Remarks

Please submit your exercises in ILIAS before 23:50 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.

Matlab

- **You should avoid the use of unnecessary loops in Matlab since the processing time will increase drastically. Instead, use built-in Matlab functions that work on vectors or matrices 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 directly into your code.**
- **If there is a built-in function for an algorithm you are supposed to implement, do *not* use it.**

!!!!: Do not change the content of the `exercise\_XX` files. If you feel the need to change them for debugging purpose remember to revert your changes before submission.

Hint: The exercise files (`exercise_XX.m`) are structured in sections marked by `%%`. You can run a section by selecting it and pressing the `Run Section` button. For incremental exercises you need to run the section one after another.

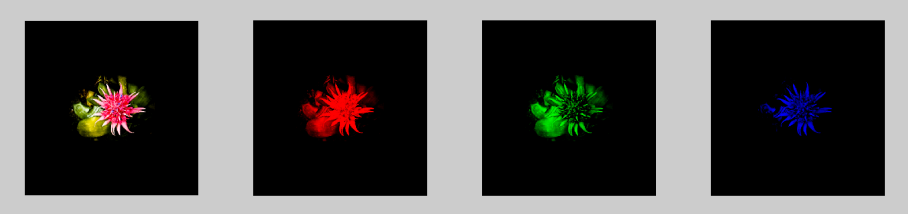
Hint: There is a binary encoded solution function for each exercise `exercise_XX(...)`, which helps you to validate your results.

Exercise 1: Working with images [4 points]

Use the script file `exercise_01` to solve this exercise. You can validate your results by running each section of the script. (Do the same for the following exercises.)

a) Write a function `my_loadImage` that loads an image and display it (`my_showImage`). Convert the image from `uint8` to floating point representation (command `double`). Values should be in between 0 and 1. Use the functions `imagesc` and set the `colormap` to `gray` or use `imshow`.

b) Write two functions `my_RGBSplit` and `my_plotRGBSplit` that split the R, G and B channels of an image and display them separately. The plots should all be included in a single figure (use `subplot`). Each channel should be plotted in its primary color.



c) Implement a gamma correction for images (`my_gammaCorrection`).

d) Write a function `my_RGB2Gray` that converts an image to grayscale. Apply your gamma correction before converting the image to grayscale.

Exercise 2: Histograms [6 points]

a) Create Histograms [2 points]: Implement a function `my_hist` which computes the grayscale histogram of an image. Implement a plotting function for your histogram (`my_plotHist`).
Double-check your result using the Matlab functions `hist` or `histc`.
What can be read out of the histogram – in general as well as for the given image?

b) Contrast stretching [1 point]: Implement a function `my_maxContrast` that maximizes the contrast of an image.
Look at the plot and discuss the result.

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