Mathematisch-Naturwissenschaftliche
Fakultät

Fachbereich Informatik
Arbeitsbereich Visual Computing

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Bildverarbeitung I (Prof. Schilling)
WS 2022/2023

Assignment 2

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. Stick to the submission procedure described in Assignment 1.
Reminder: If there is a built-in function for an algorithm you are supposed to implement, do *not* use it.

Exercise 3: Multi-Image Denoising [1 point]

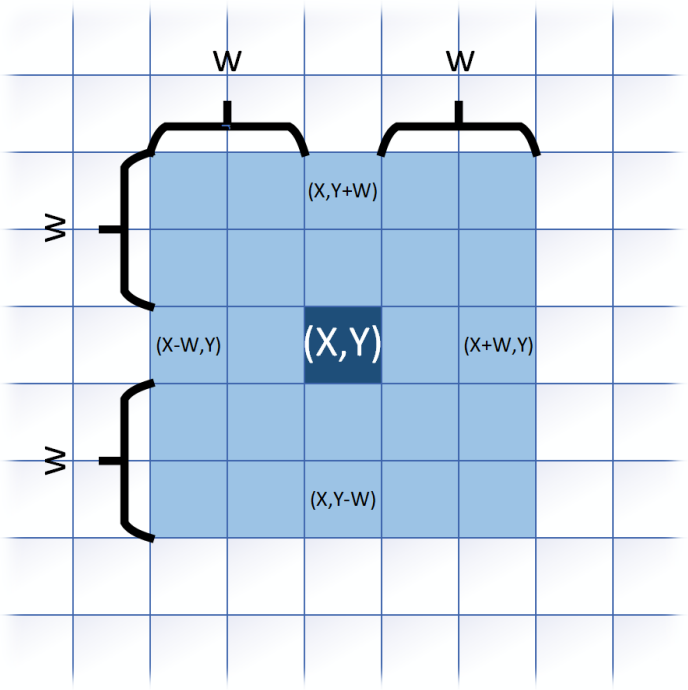
Use `exercise_03.py` for this exercise.

a) The provided images of the cup were shot with high ISO settings and contain serious amounts of noise. Complete the function `n_filter` to return a denoised image from a list of images.

b) Use the provided images of a tree for this exercise. The set contains five crops of an image sequence. You can find one of the original images on this page. Look at the result and discuss why the approach performs poorly for this sequence of images.

Exercise 4: Denoising [4 points]

The provided script `exercise_04.py` loads two image files (`cup_noisy.png` and `peppers.png`) and defines templates for the filter functions you are asked to implement in this exercise. You may vary the parameters `w` and `sigma` to explore the effect on the resulting filtered images. Hint: The parameter `w` defines the size of the window, as shown in the following figure.



a) Mean filter [1 point]: Complete the function `mean_filter` to implement a mean filter as described in the lecture.

b) Median filter [1 point]: Complete the function `median_filter` to implement a median filter as described in the lecture.

c) Gaussian filter [2 points]: Fill in the function `get_gauss_kern_2d` to make it return a two dimenensional Gaussian kernel. Use it to complete the function `gauss_filter` implementing a Gaussian filter as described in the lecture. Hint: You may use `gauss_function` implemented in `utils.py`.

Reminder: Do *not* use more than two nested for-loops in order to prevent excessive runtime of your functions! Use slicing to crop the required patch directly from the padded image.

Exercise 5: Bilateral Filtering [4 points]

Read the provided paper thoroughly. Complete the function `bilateral_filter` in `exercise_05.py` and implement bilateral filtering as described in the paper. Please use the RGB colorspace and don't convert to lab, even though the paper recommends this. Experiment with the parameters `w`, `sigma_d` and `sigma_r`.