Mathematisch-Naturwissenschaftliche
Fakultät

Fachbereich Informatik
Arbeitsbereich Visual Computing

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Bildverarbeitung I (Prof. Schilling)

WS 2023/2024

Assignment 4

Remarks

Please submit your exercises in ILIAS before 23:55 on the closing date. At least one member of the group must be able to present at our biweekly tutorial, beeing prepared to explain *each* exercise. Random groups will be asked to present their solutions. Stick to the submission procedure described in Assignment 1.
4 points are counted as a bonus for this assignment. *Hint:* You are allowed to use the `numpy` functions `np.fft.fft2`, `np.fft.ifft2`, `np.fft.fftshift` and `np.fft.ifftshift` to complete this assignment. Set their `axes`-argument to $(0, 1)$ to tansform all channels of an image with only one line of code.

Exercise 8: Image Sharpening

[7 points]

This exercise demonstrates two different approaches to sharpen an image. Use the provided file `exercise_08.py` to solve the following tasks:

a) Image Blurring [2 Points]: Complete the function `gauss_filter_freq` that blurrs an image by applying a two dimensional gaussian filter in frequency space. You may use the function `get_gauss_kern_2d` which is already imported from `utils.py`.

b) Inverse Filtering [2 Points]: Complete the function `inverse_gauss_filter_freq` that sharpens an image by inverting a two dimensional gaussian filter in frequency space.

c) Unsharp Masking [3 Points]: A low pass filter can be used to sharpen an image I_{orig}. This is called unsharp masking:

$$I_{\text{sharp}} = I_{\text{orig}} + \alpha(I_{\text{orig}} - I_{\text{blurr}})$$

Complete the function `unsharp_masking` that uses the Gaussian filter from a) to generate a sharpened image I_{sharp}.

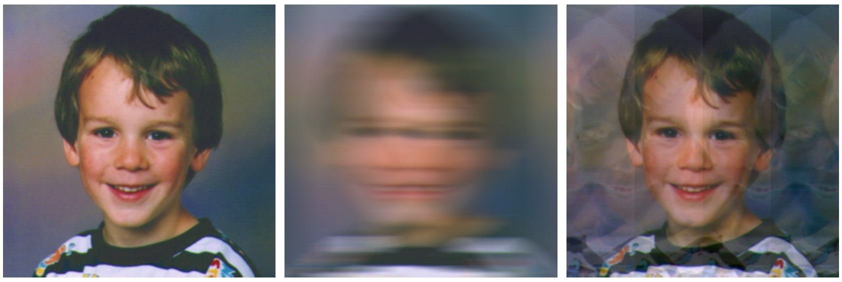
Exercise 9: Inverse Filtering

[7 points]

The left image (a) in the figure below was horizontally blurred by a box-filter of an unknown width n resulting in the middle image (b). Use the fourier transformation in combination with the inverse filtering technique to correct the altered image and obtain a result similar to (c).

a) Find the Width [4 Points]: Investigate the properties of the altered image and various (horizontal) box-filters in the frequency domain. Use your findings to determine the filter's width n and insert it to the `main`-function of the provided scipt `exercise_09.py`. **Your submission must include a pdf-file, describing your approach and reasoning your answer. Include images to illustrate the intermediate steps.**

b) Reconstruct the Original Image [3 Points]: Complete the function `reconstruct_image` in `exercise_09.py` that applies an inverse horizontal box-filter in the frequency domain. *Hint:* Use boolean masking of `numpy`-arrays to avoid divisions by zero.



(a) Original Image (b) Altered Image (c) Reconstructed Image