Mathematisch- Naturwissenschaftliche
Fakultät

Fachbereich Informatik
Arbeitsbereich Visual Computing

Eberhard Karls
Universität
Tübingen

Bildverarbeitung I (Prof. Schilling)
WS2017/2018

Assignment 4, Due: December 14th, 2017

Remarks

You are allowed to use the built-in functions `fft2` and `fftshift` for the following exercises. Grayscale images are sufficient when plotting in the frequency spectrum.

Exercise 8: 2D Filtering [4 points]

a) Implement a function `my_box2DFFT` that takes an image and a filter width as arguments and applies a 2D box filter in the frequency domain to the image. The function should return the filtered image and the kernel function (in frequency space). Also, implement a function `my_compareBox` that filters an image with `my_box2DFFT` and the built-in function imfilter using a box kernel. The function should produce a single figure that contains the following:
   - the original image
   - both versions of the filtered image
   - the original image and your filtered image in the frequency domain (use displayfft2)
   - an image that shows the differences between the results of your own implementation of the filter and the built-in version. If there are any differences, you should be able to explain them!

b) Implement two functions `my_gauss2DFFT` and `my_compareGauss` analogous to the above that apply a 2D Gaussian filter in the same way.

c) Write a function `my_plotKernels` that plots the kernel functions of two filters in the frequency domain. Use it to compare the box and Gaussian filters. Use a view that directly illustrates the behavior (appearing artifacts) of the two kernels.
   Explain how the differences affect the filtering results!

Exercise 9: Unsharp Masking [3 points]

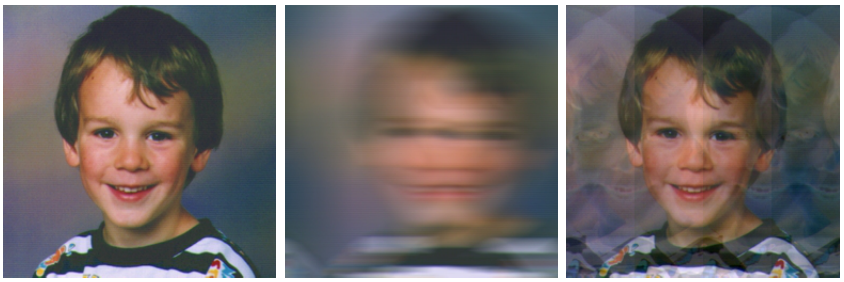
A low pass filter can be used to sharpen an image. This is called unsharp masking:

$$I_{sharpened} = I_{original} + k * (I_{original} - I_{lowpass})$$

Implement a function `my_unsharpMasking` that uses a 2D Gaussian filter to sharpen an image. Find a value for k that produces the best results.

Exercise 10: Inverse Filtering [8 points]

Use the fourier transformation in combination with the inverse filtering technique to implement a function `my_inverseFilter`. Use it to reconstruct the original image for 'TimHBox.hdr'. Find out what the size of the original filter was. Your submission should include a pdf-file which describes your approach and how you found your answer. Include images to illustrate the intermediate steps.



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

HINTS:

- *the image was filtered with a horizontal box filter of width n*
- *look at the properties of the altered image and the box filter in the frequency domain*
- *try to reproduce the effect of the filter on an image of your choice*
- *experiment!*