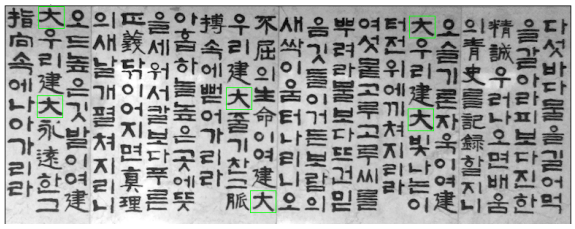
Assignment 4

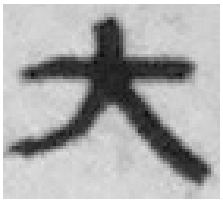
Welcome to the fourth assignment of the lecture *Visual Computing*. **Please read all instructions carefully!** This assignment covers Fourier transform, template matching, and image pyramids. Submission is due on Monday, Mai 12th, 2021 at 8pm. Please send your solutions via `read.mi.hs-rm.de`.

Exercise 1 (5 points)**.** Write a python program `findregions.py` that

1. allows you to mark a rectangular region in an image which serves as a template (see (b) in the example below)
2. and then to find and mark all identical regions in the image by a green rectangle (see (a) in the example depicted below) using normalized cross-correlation. You can find the example image with the korean signs on the lecture website.



(a) Image with marked korean signs



(b) Template

Exercise 2 (5)**.** Write a program `blendimages.py` that reads in two images and blends the left half of the first image with the right half of the second image using

1. *direct blending* (see example (c) below).
2. *pyramid blending* (see example (d) below). Therefore, you have to
   (a) generate the *Gaussian pyramid* for each image up to a prescribed level,
   (b) construct the *Laplacian pyramids* using the beforehand constructed *Gaussian pyramids*,
   (c) join the left half of one image with the right half of the other image in each level of the *Laplacian Pyramid*, and
   (d) reconstruct the blended image from this joint image pyramids.

(a) Zebra

(b) Horse

(c) Direct blending

(d) Pyramid blending

Exercise 3 (10 points)**.** Combining the low spatial frequencies of one picture with the high spatial frequencies of a second picture generates so called *hybrid images*. These hybrid images are perceived in two different ways depending on the distance between the spectator and the image. In the frequency domain a hybrid image h combined from the two images i_1, i_2 is simply given by

$$H = I_1 \cdot G_1 + I_2 \cdot (1 - G_2),$$

where I_1, I_2 are the Fourier transform of i_1, i_2 and $G_1, (1 - G_2)$ are a low-pass filter and a high-pass filter respectively. The hybrid image depicted below was generated using the same gaussian filter

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (x, y \text{ are distances to the image center})$$

for G_1 as well as G_2. The parameter σ controls the amount of information that contributes to the hybrid image from each of the source images.

For more details on hybrid images see e.g. the publication of *A. Oliva, A. Torralba, P. G. Schyns, Hybrid Images, Siggraph 2006, 527-532.*

Write a simple python program `hybrid.py` to generate hybrid images, that

1. (8 points) can read in two grayscale images and the parameter *sigma* of a gaussian filter kernel. After that, it should calculate, show and save the filtered images as well as the resulting hybrid image.

2. (2 points) can do the same as in 1. but for color images.