Exercises for
Introduction to Pattern Recognition (IntroPR)
Lennart Husvogt
Work sheet 4, 09.11.2015

k-Means Clustering and Histogram Equalization

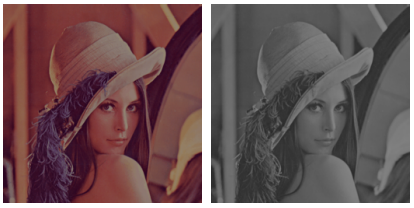
Exercise 11 Implement k-means clustering as it was shown in the lecture. Implement it for arbitrary values for k and arbitrary vector sizes.

Generate an array with randomized 2D vectors and perform k-means clustering. Plot the resulting clusters using matplotlib. Use a different color for each cluster and mark the mean.

Tip: Use Python dictionaries to store key-pair values such as a cluster index and a list of vectors. A Python list can be converted into an array with the command `numpy.asarray` (`http://docs.scipy.org/doc/numpy/reference/generated/numpy.asarray.html`).

You can generate random vectors using `numpy.random.rand` (`http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.rand.html`).

Exercise 12 Implement histogram equalization in Python and apply it to the images `Lena.png` and `LenaLowContrast.png`. Note that the first image is a 24-bit RGB image, whereas the second image is an 8-bit gray level image.



(a) Briefly explain the idea of histogram equalization and what kind of effect it has on the contrast within an image.

(b) Implement a function to calculate the cumulative density function for an input image.

(c) Compute the mapping for the histogram equalization. Note that the output intensities should be within the same range as the input intensities.

(d) Extend your methods to work also on multi-channel images (e.g. RGB) and apply the methods to the images.

(e) The histogram equalization changes the color balance if it is directly applied to RGB. How can this problem be addressed?