Foundations of Artificial Intelligence

Prof. Dr. B. Nebel, Prof. Dr. M. Riedmiller
S. Lange, J. Witkowski, D. Zhang
Summer Term 2010

University of Freiburg
Department of Computer Science

Exercise Sheet 8

Due: Tuesday, June 29, 2010

Exercise 8.1 (Programming Assignment K-Means Clustering)

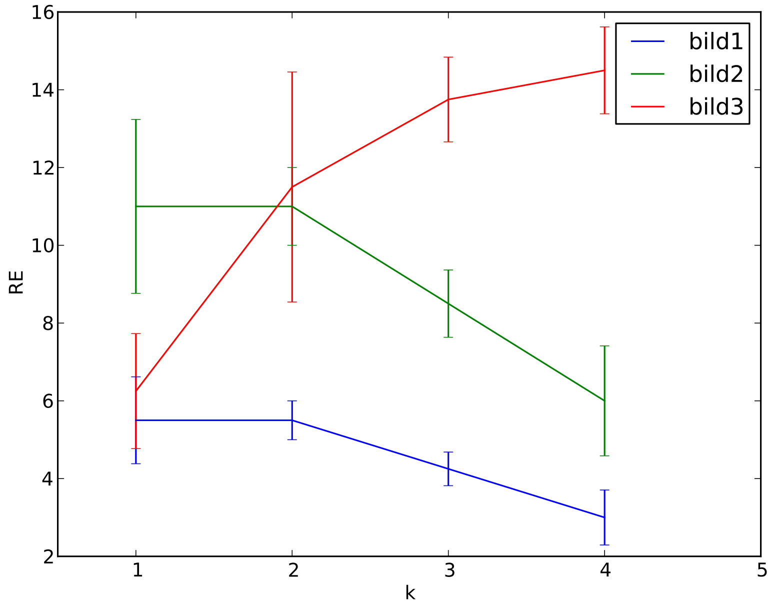
This programming assignment asks you to use K-Means for reducing the number of colours in an image, as was shown with the example images on slides 17/19 and 17/20 during the lecture of June, 15th.
Please process the following three colour images:



Until the deadline of this sheet, we provide these images in a package on the supporting website. The package contains JPEGs as well as PPMs (Portable Pixmap). PPMs are especially easy to read in from several programming languages. You can find a description of this format on Wikipedia: http://en.wikipedia.org/wiki/Netpbm_format#PPM_example. With the help of tools such as *convert* (from ImageMagick) you can convert these files to your prefered image format.
As you were asked on the last sheet, please hand in the source code of your program / script AND your images, plots and answers—any programming language will be accepted, but don't expect your tutor to debug and / or run your code.

(a) Write a program, that reads the colour values from an image file, represents them as RGB-Triples (red, green, blue channel) or vectors in a three-dimensional vector space, applies the K-Means algorithm in order to find clusters, replaces the original colour values by the values of the corresponding cluster centers and then write the resulting images with only k different colours. Implement either "soft" or "hard" K-Means clustering. Please hand in the three resulting images for $k = 6$ aside your sources. *Bonus task:* Repeat the clustering process for comparison reasons, this time using the YUV, HSV, or CIELAB colour space instead of using the RGB colour space.

(b) Conduct a series of experiments for each of the three images, where you apply K-Means with $k = 1, 2, 3, \ldots$ and calculate the reconstruction error as defined in the lecture. Depending on the choice of initial cluster positions, K-Means will converge to different local minima. Therefore, repeat each experiment for ten times with different starting positions of the cluster centers. Hand in a plot displaying the average reconstruction error and its standard deviation for each of the tested k (as depicted in the exemplary plot below with standard deviations displayed as error bars). Ideally, your program runs the series of experiments automatically. Depending on your implementation, you will notice the execution time of K-Means to rise significantly even for small k. Continue at least until $k = 8$, if possible until $k = 16$, $k = 32$, or even $k = 64$.



(c) Discuss how your implementation of the algorithm could be further improved concerning the execution time. Which method that has been presented a few lectures ago, could be of great help at what step(s) of the algorithm? How could modern parallel computer architectures help reduce the execution time and which particular techniques could be of use in this particular task? *Bonus task:* Speed up your implementation as much as possible.

Warning: Please start early to work on this sheet. Depending on your solution, the execution time of the necessary calculations might be quite high.

The exercise sheets may and should be handed in and be worked on in groups of three (3) students. Please fill the cover sheet[1] and attach it to your solution.

[1] `http://www.informatik.uni-freiburg.de/~ki/teaching/ss10/gki/coverSheet-english.pdf`