Algorithms for Image Processing

Prof. Andreas Weinmann

h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbmn
FACHBEREICH MATHEMATIK
UND NATURWISSENSCHAFTEN

Problem Sheet 1

Problem 1 (First week)
Write a procedure which determines the minimum of a convex function f in the interval $[a, b]$ using "trisection of the interval." Use the function

$$f(x) = e^{-\alpha x} + x^{\beta} \tag{1}$$

on the interval $[0, 1]$ as a function for testing and employ different values $\alpha, \beta \in [1, 5]$.

Problem 2 (First week)
Write a procedure which determines the interquartile range of a set of numbers. (Base your algorithm on the quicksort algorithm and only sort that parts necessary.) Test your program. Plot run time versus data size.

Problem 3
The *entropy* measures the information of the outcome of a random process. Assume that the random process has n states and that each state has probability p_i, $i = 0, \ldots, n-1$, then the entropy is given by

$$E = \sum_i p_i \mathrm{ld}(1/p_i) = -\sum_i p_i \mathrm{ld}(p_i).$$

For estimating the entropy, one frequently employs the relative frequency. For instance, in image processing, one may employ the relative frequency of gray values in a neighborhood to estimate the information in an image locally.
The task is to compute the local entropy for a circle of radius r, and more generally, a v- normal neighborhood at every coordinate in the image, thus producing an "entropy" image. To speed-up the algorithm, use the ideas of median filtering a la Huang.
Remark. There is only a small number of relative frequencies p_i which may appear. Using this fact in connection with a lookup table containing $p_i \mathrm{ld}(p_i)$ may yield a further speedup of the algorithm.

Problem 4 (Optional, extra credit) a) Implement a maximum filter for a line (1d data) for a window of size A using Gil's method (see lecture notes).

b) Based on this filter, implement a method to realize a maximum filter for the special windows fitting in one line of width A which works in a 2d image of arbitrary size.

Hint. Using the reshape command allows us to consider the 2d array as a 1d array. (Note that, physically, a 2d array is stored linearly anyway.) Then we my apply the previous filter and reshape to a 2d array again. Doing so, we may get problems at the boundary: Why? Please explain and resolve them.

c) Employ the previous part to implement a maximum filter for rectangular neighborhoods.

d) Employ the previous results within a wrapper routine which computes a maximum filter for rectangular neighborhoods with suitable boundary treatment.

e) Compare the results and runtimes with the results and runtimes of the method implemented in Scipy. Concerning the time comparision, generate a filter-size vs. time plot for sizes of A starting from 3×3 to 69×69.