INF-MAT 5360: Obligatory project 1

To be handed in by October 5

This obligatory exercise consists of two parts, one part focusing on algorithms and their implementation, and a second part with a few theoretical exercises.

1 Image segmentation using minimum spanning trees

We will consider a simplified model for image segmentation using graphs and minimum spanning trees. For the purposes of this exercise we consider an image to be represented as an $m \times n$-matrix $I = [p_{i,j}]$ where the pixels $p_{i,j}$ take on integer values in the range $[0, P]$. Let $N = mn$ denote the total number of pixels in the image. The following is an example of a 4×4-image.

$$\begin{bmatrix} 0 & 1 & 1 & 3 \\ 1 & 2 & 3 & 5 \\ 0 & 4 & 3 & 7 \\ 5 & 3 & 3 & 1 \end{bmatrix}$$

In graph-based approaches to image segmentation the image is used to define an undirected graph $G = (V, E)$ where each image pixel $p_{i,j}$ has a corresponding vertex $v_{i,j}$. We add an edge between each pair of neigbouring pixels both vertically and horizontally. The weight (or length) of the edge is set to the squared value of the difference between the two pixel values, e.g., $w_e = (p_{i,j} - p_{i,j+1})^2$ for the edge between nodes $v_{i,j}$ and $v_{i,j+1}$.

A *segmentation* is a partition of the nodes V into sets $S_1, \ldots, S_k$ such that $V = \cup S_i$ and $S_i \cap S_j = \emptyset$ for $i \neq j$. The following is a simple algorithm to find a segmentation:

1. Find a minimum spanning tree T of G
2. Remove the $k-1$ longest edges in T
3. Let the segmentation be given by the components of the remaining forest.

Exercises

1. Construct the graph G for the example image above, and find the minimum spanning tree.

2. Implement the algorithm above using either Prim or Kruskal's method for finding a minimum spanning tree. Use a programming language of your own choice. Test your algorithm on the image that can be downloaded from the course web pages. Let $k = 3$.

3. Analyze your algorithm to find the running time with respect to the number of pixels in the image. Verify the claim by constructing some random images of various sizes and log the running time.

2 Additional exercises

Exercises from Schrijver's lecture notes: *1.10, 2.2*

Please send the answers to the exercises and a running version of your program with commented code to *Truls.Flatberg@sintef.no*.

Good luck!