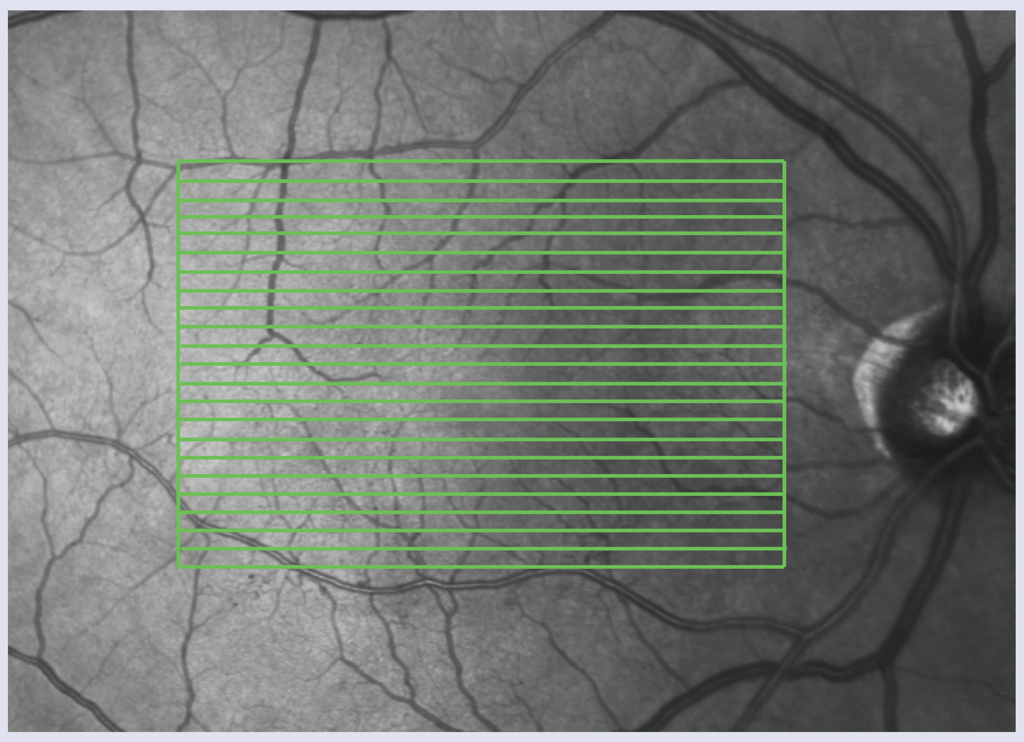
Segmentation of retinal layers in OCT-images using optimal graph search

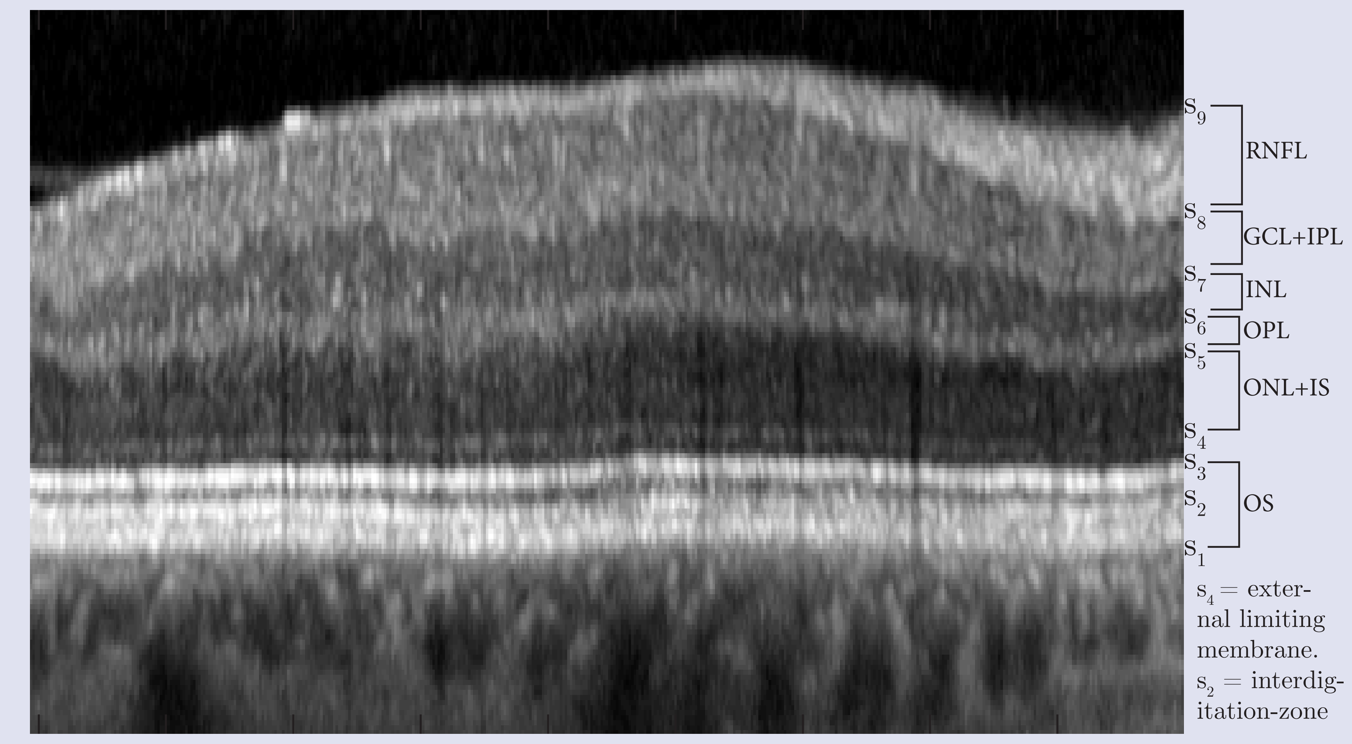
Søren Trads Steen



Introduction

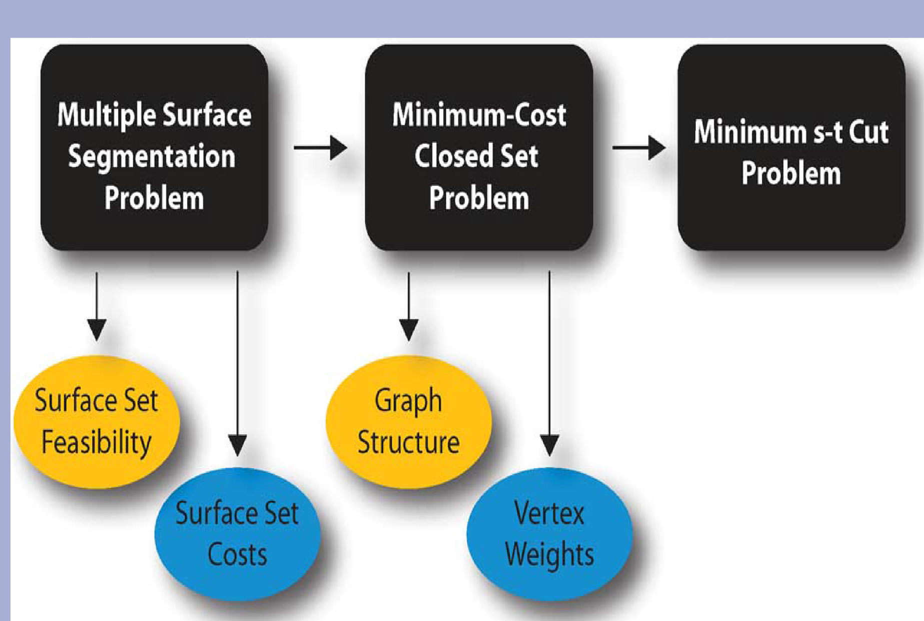
Many diseases threatening eyesight have root in the retina. Many of these diseases can be prevented if diagnosed early. This can be done using OCT images. However, a doctor has to shift through all the images to find symptoms manifesting in the different layers of the retina - if an algorithm can supply clues of where to look, it would be a great help.

Parallel scans are taken as seen by the horizontal lines to the left and could be flattened to look like the one to the right.



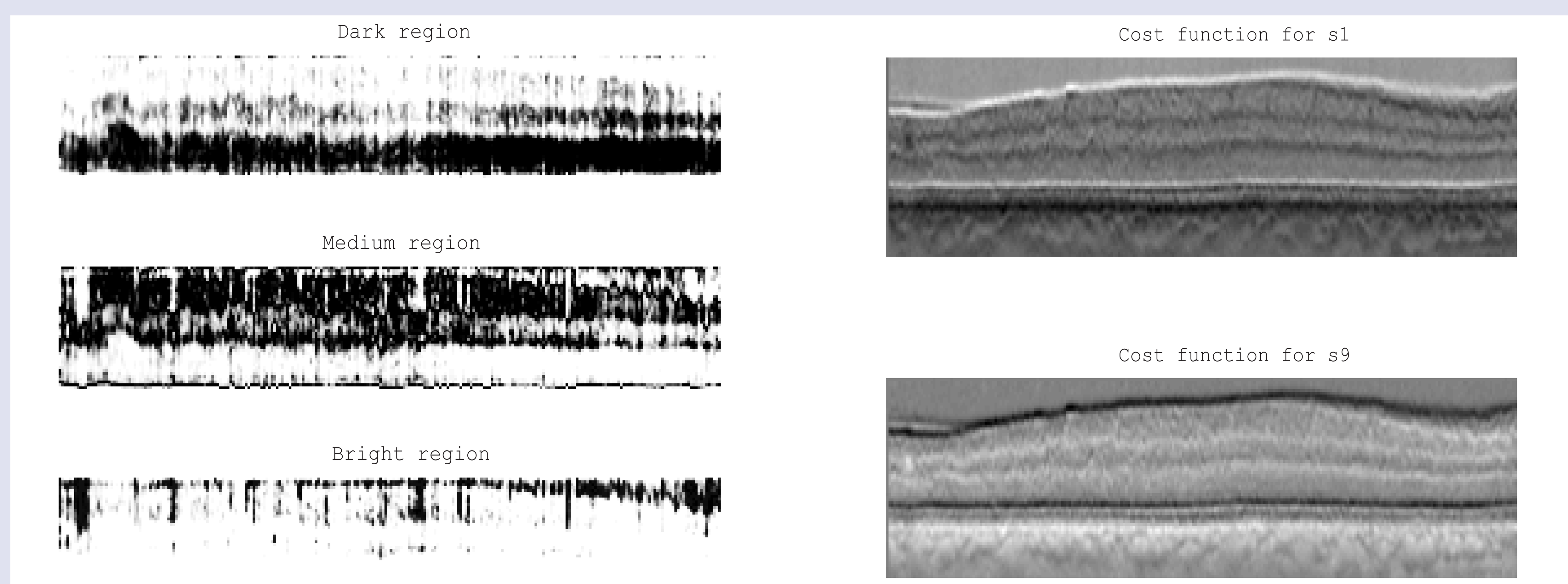
Graph search

The algorithm uses a transformation from the multiple surface segmentation problem to a problem of finding a mimum-cost closed set in a closed graph by constructing a graph with each voxel becoming a vertice. The node weights will then correspond to cost functions while the structure of the edges will reflect the feasibility constraints(such as a known interval of distances of the surfaces or how quickly a surface is allowed to rise and fall.). The minumum cost closed set is found by finding a minimum *st*-cut in a closely related graph.
The algorithm works in 3D, so instead of segmenting surfaces in individual 2D images, we can perform the segmentation with volumetric data, and not lose 3D information.



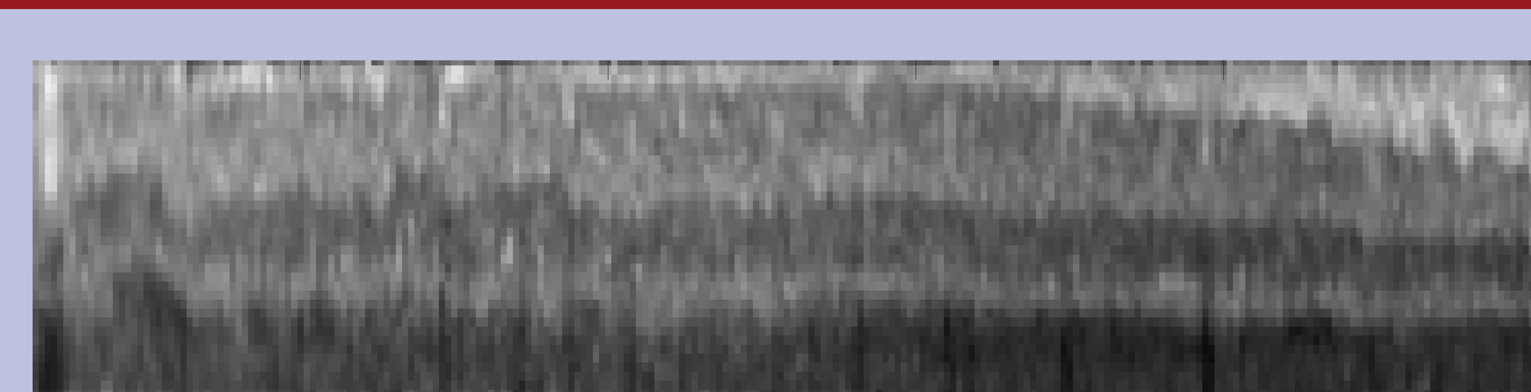
Cost functions

The algorithm takes two types of cost functions:
- Surface cost functions describing the reverse probability of a pixel being on that surface. Examples of surface cost functions are seen on the right. A convolution is used to construct cost functions favouring gradient shifts as well as favouring a number of dark or bright pixels above/below the surface.
- Regional cost functions describe the probability of a pixel beloning to a specific region, where a region is the area between two surfaces. If there are K surfaces, there are $K+1$ regions. To construct the regional costs, fuzzy membership functions has been used. We discern between three different regions - dark, medium and bright regions.



Splitting up the problem

The problem is split up - first we find surface 1, 3, and 9 using only on surface cost functions in an image downsampled by a factor 2.
Next we transform the area between surface 3 and 9 into a rectangular area using linear interpolation of the downsampled image(see an example on the left). Now surface 5,6,7, and 8 are found using both surface costs and regional costs.
At last a smaller part can be made of the area between surface 1 and 3, in which surface 2 is found using a surface cost function, however now the fullsize image is used, as not enough information is present in the downsampled image to discern the small detail required to find this surface. A smaller area between surface 3 and 5 is made in a similar fashion to find surface 4.



Results

The algorithm works. A slice from the final segmentation is shown below. The thickness of the different layers are shown beside it.