7 Layered surfaces

A SEGMENTATION PROBLEM can sometimes be geometrically constrained. We may for example be interested in segmenting a roughly horizontal layer or a roughly circular object. In 3D we may be interested in terrain-like surface, a tubular object, or a spherical object. Such topological constraints strongly reduce the solution space for the segmentation, and may turn an otherwise challenging problem into an easily solvable problem.

An example of segmentation problem which may benefit from constraining the solution, such that it consists of layers, is shown in Figure 7.1.

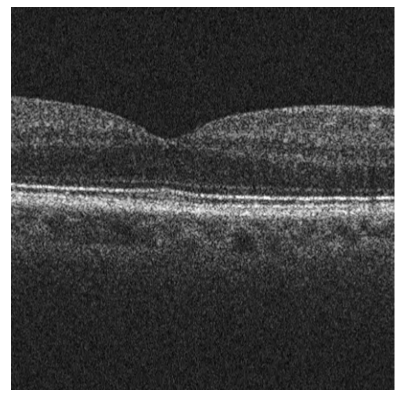

Figure 7.1: OCT (optical coherence tomography) image of retina. Quantifying the thickness of retinal layers informs about eye disease and is therefore of clinical importance.

In this exercise, we focus on optimal net surface detection via graph search originally suggested by Wu and Chen[1] and popularized by Li, Wu, Chen and Sonka[2]. To segment terrain-like surfaces, they construct a graph on a set of sample points from a volume, such that the roughness of possible solutions is constrained. The optimality of the solution is defined in terms of a volumetric cost function derived from the data. The algorithm can find multiple interrelated layered terrain-like and tubular surfaces, which made it applicable for medical image segmentation and led to numerous extensions. One important extension involves a cost function which determines an optimal placement of the surface: a cost function, originally defined only in terms of on-surface appearance, has been extended to incorporate appearance of the regions between surfaces [3].

[1] Xiaodong Wu and Danny Z Chen. Optimal net surface problems with applications. In *Automata, Languages and Programming*, pages 1029–1042. Springer, 2002

[2] Kang Li, Xiaodong Wu, Danny Z Chen, and Milan Sonka. Optimal surface segmentation in volumetric images – a graph-theoretic approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):119–134, 2006

[3] Mona Haeker, Xiaodong Wu, Michael Abràmoff, Randy Kardon, and Milan Sonka. Incorporation of regional information in optimal 3-d graph search with application for intraretinal layer segmentation of optical coherence tomography images. In *Information Processing in Medical Imaging*, pages 607–618. Springer, 2007

We will here review an algorithm for finding optimal layered surfaces in 3D, with focus on the inputs and the outputs. For details on *how* this algorithm works, the reader is referred to article by Li et al. Note that while the theory given in 7.1 covers the 3D case (surfaces in the volume), the exercise in 7.4 is on 2D case (curves in the image).

We also very briefly cover the principle of transforming the data into a volumetric cost, which is the input to the layered surface detection algorithm. Also this aspect of the layered surface detection is simplified for the exercise, where we only consider cost functions derived directly from pixel intensities.

7.1 Layered surface detection

In a discrete volume $x \in \{1, \ldots, X\}$, $y \in \{1, \ldots, Y\}$, $z \in \{1, \ldots, Z\}$, a terrain-like surface s defined by $z = s(x, y)$ satisfies a smoothness constraint (Δ_x, Δ_y) if

$$|s(x,y) - s(x-1,y)| \leq \Delta_x \quad \text{and} \quad |s(x,y) - s(x,y-1)| \leq \Delta_y \,. \tag{7.1}$$

For a cost volume $c(x, y, z)$, an *on-surface* cost of s is defined as

$$C_{\text{on}}(s,c) = \sum_{x=1}^{X} \sum_{y=1}^{Y} c(x,y,s(x,y))\,. \tag{7.2}$$

The *optimal net surface problem* is concerned with finding a terrain-like surface with a minimum cost among all surfaces satisfying the smoothness constraint.

The polynomial time solution presented in the work by Wu and Chen transforms the optimal net surface problem into a problem of finding a *minimum-cost closed set* in a node-weighted directed graph with nodes representing volume voxels. This is further transformed into a problem of finding a *minimum-cost s-t cut* in a related arc-weighted directed graph. Minimum-cost *s-t* cut can be solved in polynomial time and efficiently found using the algorithm of Boykov and Kolmogorov [4], a well known tool for many image segmentation tasks. While the optimal net surface problem is ultimately solved using the minimum-cost *s-t* cut algorithm, it should be noted that the graph constructed for surface detection is different from the graph used for Markov random fields.

[4] Y Boykov and V Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, 2004

The solution to the optimal net problem gives a practical tool for detecting a surface in a volume, or a curve in an image. To use the tool, we need to (somehow) transform the image data into a cost function with the property of having small values in regions where we expect to find the surface. The practical value of the solution to the optical net problem is further increased by two very useful extensions which we describe next.

7.1.1 Multiple surfaces

The extension to multiple surfaces developed in [5] may be exemplified by considering two terrain-like surfaces s_1 and s_2. The surfaces are said to meet an overlap constraint $(\delta_{\text{low}}, \delta_{\text{high}})$ if

[5] Kang Li, Xiaodong Wu, Danny Z Chen, and Milan Sonka. Optimal surface segmentation in volumetric images – a graph-theoretic approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):119–134, 2006

$$\delta_{\text{low}} \leq s_2(x,y) - s_1(x,y) \leq \delta_{\text{high}}\,. \tag{7.3}$$

Given two cost volumes c_1 and c_2 the total cost associated with surfaces s_1 and s_2 is

$$C_{\text{on}}(s_1, c_1) + C_{\text{on}}(s_2, c_2)$$

and the optimal surface detection will return a pair of surfaces with a minimum cost among all surfaces satisfying overlap and smoothness constraint. Depending on the problem at hand c_1 and c_2 may be different or identical, and likewise smoothness constrains may vary or be the same for the two surfaces.

7.1.2 *In-region cost*

When detecting only one surface s, instead of on-surface cost as in 7.2 we may define a cost for the two regions: the one below the surface and the one above the surface. The cost of the surface s is then given by

$$C_{\text{in}}(s, c_{\text{below}}, c_{\text{above}}) = \\ = \sum_{x=1}^{X} \sum_{y=1}^{Y} \left(\sum_{z=1}^{s(x,y)} c_{\text{below}}(x,y,z) + \sum_{z=s(x,y)+1}^{Z} c_{\text{above}}(x,y,z) \right) . \quad (7.4)$$

That is *one* surface may be found by defining *two* cost volumes: one with small values (dark) below and on the surface, and the other with small values above the surface. In-region cost is evaluated over larger area, making this approach very robust to noise. It is also practical in cases where the boundary between two regions is blurry.

Finally, two set of *layered* (i.e. non-intersecting and ordered) surfaces give rise to an *in-region* cost corresponding to the regions between two neighboring surfaces. So for two surfaces s_1 and s_2 we have

$$C_{\text{in}}(s_1, s_2, c_{1,2}) = \sum_{x=1}^{X} \sum_{y=1}^{Y} \sum_{z=s_1(x,y)+1}^{s_2(x,y)} c_{1,2}(x,y,z) . \quad (7.5)$$

This cost, together with the cost for the region under the surface s_1 and the region over the surface s_2, can be incorporated into the minimization problem [6]. In the text below, we use notation $c_{0,1}$ and $c_{k,k+1}$ for the cost volumes below the first and above the last (k-th) surface.

[6] Mona Haeker, Xiaodong Wu, Michael Abràmoff, Randy Kardon, and Milan Sonka. Incorporation of regional information in optimal 3-d graph search with application for intraretinal layer segmentation of optical coherence tomography images. In *Information Processing in Medical Imaging*, pages 607–618. Springer, 2007

7.2 *Summary*

To summarize, for finding K cost-optimal layered surfaces we need to define

- K on-surface cost volumes c_k, $k = 1, \ldots, K$, and/or
- $K{+}1$ in-region cost volumes $c_{k,k+1}$, $k = 0, \ldots, K$.

The set of feasible surfaces is given by

- K smoothness constraints $\left(\Delta_x^k, \Delta_y^k\right)$, $k = 1, \ldots, K$ and
- $K{-}1$ overlap constraints $\left(\delta_{\text{low}}^{k,k+1}, \delta_{\text{hight}}^{k,k+1}\right)$, $k = 1, \ldots, K{-}1$.

Layered surface detection has found an immediate use for detecting tubular surfaces. The main principle is the fact that a circle $x^2 + y^2 =$

ρ^2 appears as a straight line $r = \rho$ when represented in polar (r, θ) coordinates. Detecting a tubular surface is achieved by representing the volumetric data in a cylindrical coordinate system (r, θ, z) with the longitudinal axis $r = 0$ roughly aligned with the center of the tube. We call this transformation *unwrapping* the volume, and we also say that the volume is sampled along the radial rays. An important practical parameters for unwrapping are the radial and the angular resolution. In the unwrapped representation, the tubular surface is terrain-like and can be defined as $r = s(\theta, z)$. When using layered surface detection for detection of tubular surfaces, additional constraints are added to ensure a smooth transition over $\theta = 0$.

7.3 Constructing cost volumes

The surfaces returned by the layered surface detection algorithm are optimal in terms of the volumetric cost. Therefore, as mentioned, to detect a surface we need to define a cost volume which takes small values where the data $V(x, y, z)$ supports the surface k. This modelling step, crucial for the performance of the algorithm, is fully dependent on the data.

The transformation from image intensities to cost function often involves filtering, computing gradients or cumulative sums, truncating values etc. Finding a suitable cost function may require some expertise.

If the surface to be detected is characterized by a certain voxel intensity v_s, then the cost volume may be defined as $(V - v_s)^2$. More often, the surface divides two regions of different intensities, so cost volume needs to be defined in terms of change of intensity. When computing intensity changes for tubular surfaces, the best approach is to first unwrap the volume, and then compute the change in the r direction.

7.3.1 Examples

Figure 7.2 demonstrates the use of the layered surface detection for detecting a terrain-like curve in an image.

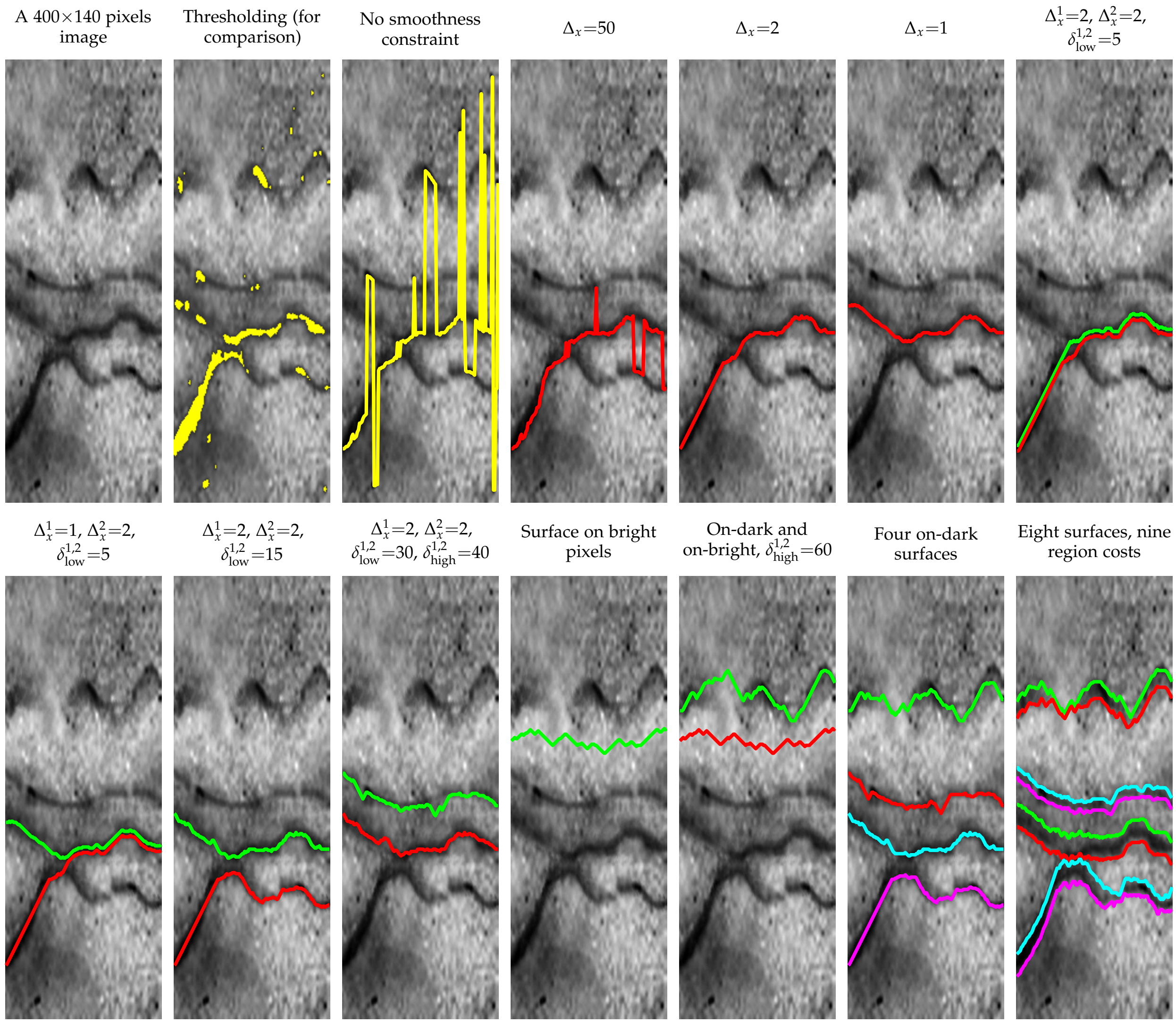

Figure 7.2: Output of layered surface detection. First three images serve to illustrate the problem). Images 4–6 show how changing the smoothness constraint influences the result. Images 7–10 demonstrate the use of the overlap constraint. Images 11–12 demonstrate the use of different cost functions. Image 13 is a four-surface detection, while image 14 uses region costs.

7.4 Exercise: Layered surfaces in 2D

In this exercise you will get familiar with layered surfaces. We will only consider layered surfaces in 2D.

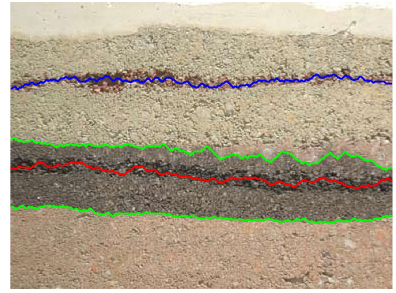

Figure 7.3: Layers in limestone.

Tasks

1. Run the script `on_surface_cost_example` which demonstrates the use of *on-surface* cost. In this case we want to detect dark lines, so for on-surface cost we use image intensities. Get familiar with the functions for computing the optimal solution.

2. Run the script `in_region_cost_example` which demonstrates the use of *in-region* cost. In this case we want to separate the dark and the brigth regions, so for in-regions cost of dark region we use image intensities I, while for bright region we use $255 - I$. Get familiar with the functions for computing the optimal solution and passing the region costs to the solver.

3. Inspect the image `rammed-earth-layers-limestone.jpg`. Use the layered surface detction to detect the darkest line in the image, shown red in Figure 7.3. Then, detect two dark lines (blue and red line in the figure). Finally, detect the lines partitioning the dark regions, as shown in green in Figure 7.3.

7.5 Exercise: Quantifying dental tomograms

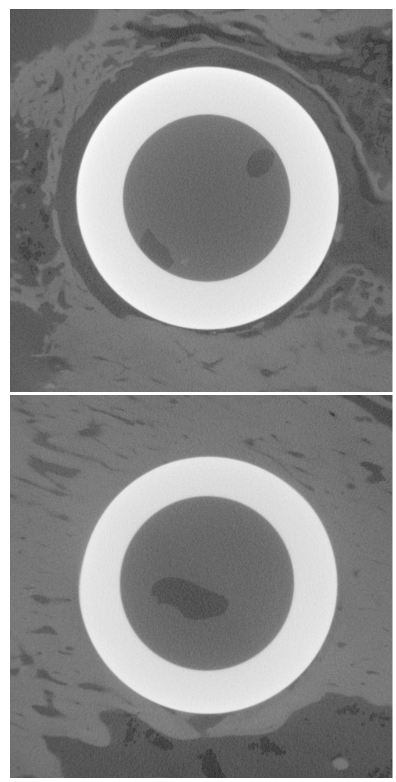

Figure 7.4: Two slices from a dental tomogram with different measure of osseointegration.

In this exercise, you will use layered surface detection to solve a concrete image analysis problem. The problem may be solved using different approaches, and you are encouraged to find your own solution. You can therefore interpret tasks below as hints, and you don't need to solve all the tasks.

We will address the problem of quantifying dental tomograms. The success of the dental implants depends on osseointegration, the formation of a direct interface between an implant and bone. An experiment was conducted to experiment how different conditions and treatments affect ossointegration. To assess the outcome of the experiment, we want to measure the interface between an implant and the bone.

For example, consider the two slices in Figure 7.4.

Tasks/hints

1. Inspect the data.

2. Use unwapping to handle tubular (circular) data.

3. Detect the layer corresponding to the surface of the dental implant. You may treat the slices individually or as a 3D volume.

4. For one measure of osseointegration consider a curve displaced 20 pixels from the surface of the implant. Express the measure of osseointegration as the percentage of this curve which is passing trough bone. You may use thresholding to divide bone from air (threshold is around 110). You should get the result similar to Figure 7.5.

5. Optional challenge. For another measure of osseointegration detect a surface (curve) defining the transition from bone to air in vicinity of the implant surface. Quantify the distance between the bone and the implant, for example as the mean distance. You may also produce a histogram of distances.

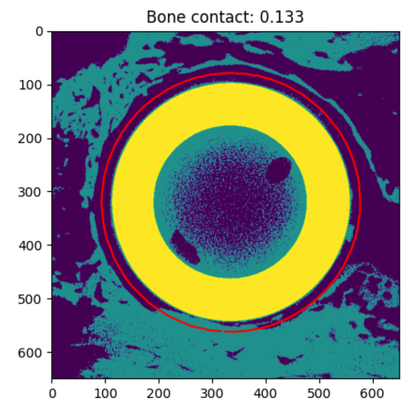

Figure 7.5: Quantification of bone contact for one slice from dental tomogram.

7.6 Exercise: Quantifying abdominal fat

This is another quantitative image analysis problem which may be solved using layered surfaces. The task is to measure the thickness of abdominal fat from the MRI (magnetic resonance imaging) scans. For example, consider the image in Figure 7.6.

The MRI images share similarities with the slices from dental implants, so you can re-use some parts of the code developed for the previous exercise. However, abdominal cross-section is less circular than dental implant, so you might need to adjust the parameter δ.

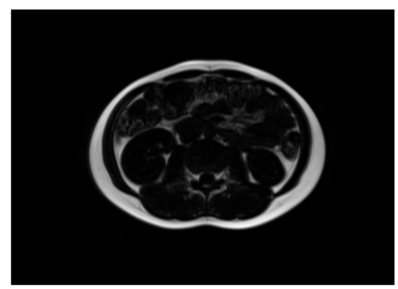

Figure 7.6: A slice from abdominal MRI.

Tasks/hints

1. Inspect the data.

2. Use unwapping to handle tubular (circular) data.

3. Detect two layer corresponding to the outer and inner surface of the abdominal fat.

4. Estimate the average thickness of the abdominal fat (in pixels). One way of achieving this is to compute the radial distances between the inner and the outer surface and average these distances. Alternatively, you can compute the fat area, and divide with abdominal circumference.