Incorporation of Regional Information in Optimal 3-D Graph Search with Application for Intraretinal Layer Segmentation of Optical Coherence Tomography Images

Mona Haeker[1,2], Xiaodong Wu[1], Michael Abràmoff[1,3], Randy Kardon[3], and Milan Sonka[1,3]

[1]Departments of Electrical & Computer Engineering, [2]Biomedical Engineering, and [3]Ophthalmology & Visual Sciences, The University of Iowa, Iowa City, IA 52242, USA
{mona-haeker, milan-sonka}@uiowa.edu

Abstract. We present a method for the incorporation of regional image information in a 3-D graph-theoretic approach for optimal multiple surface segmentation. By transforming the multiple surface segmentation task into finding a minimum-cost closed set in a vertex-weighted graph, the optimal set of feasible surfaces with respect to an objective function can be found. In the past, this family of graph search applications only used objective functions which incorporated "on-surface" costs. Here, novel "in-region" costs are incorporated. Our new approach is applied to the segmentation of seven intraretinal layer surfaces of 24 3-D macular optical coherence tomography images from 12 subjects. Compared to an expert-defined independent standard, unsigned border positioning errors are comparable to the inter-observer variability (7.8 ± 5.0 μm and 8.1 ± 3.6 μm, respectively).

1 Introduction

Even though medical images commonly exist in three or more dimensions, the ability to efficiently and accurately segment images in 3-D or 4-D remains a challenging problem. For example, in order to make many 3-D segmentation approaches practical, optimality of the resulting segmentation is often not guaranteed because of the computational demands in searching a large solution space [1, 2].

Nevertheless, Li *et al.* [3] recently presented a low-polynomial time graph-based approach for the *optimal* multi-surface segmentation of 3-D or higher dimensional data. This was an extension of the approach for the optimal detection of single surfaces presented by Wu and Chen [4] to the multiple surface case. In these approaches [3,4], the surface segmentation problem is transformed into that of finding a minimum-cost closed set in a constructed vertex-weighted geometric graph. The edges of the graph are defined so that each closed set in the graph corresponds to a feasible surface (or set of feasible surfaces). Furthermore, the vertex costs are assigned so that the cost of each closed set directly

N. Karssemeijer and B. Lelieveldt (Eds.): IPMI 2007, LNCS 4584, pp. 607–618, 2007.

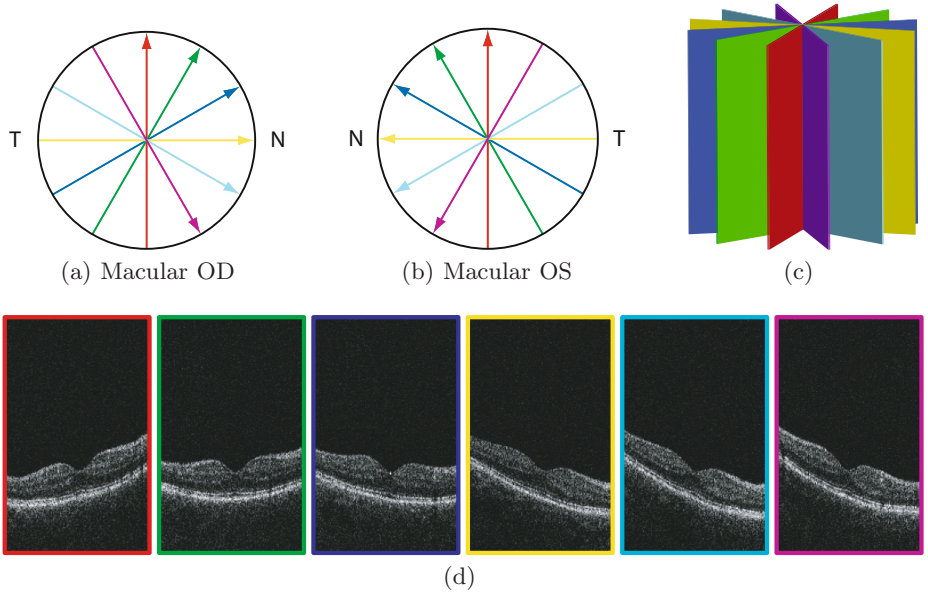
corresponds to the cost of the set of surfaces. The closed set with the minimum cost (corresponding to the optimal set of surfaces) is then determined by finding a minimum *s*-*t* cut in a closely-related graph. Note that even though the surfaces are ultimately found by finding a minimum-cost *s*-*t* cut in a constructed graph, these approaches [3,4] are fundamentally different than the "graph cut" methods of Boykov *et al.* (e.g., [5]).

While the edges of the graph are important in determining what it means for a surface to be feasible, it is the cost function that determines what set of surfaces is optimal. As originally presented in [3, 4] and used by all of the applications to date (e.g., [6, 7]), the cost of a set of surfaces is defined as a summation of cost values associated with voxels on the surfaces (i.e., the cost of a voxel with respect to a particular surface reflects the unlikeliness that the voxel would be part of the surface). While such "on-surface" costs can incorporate both image edge and regional information [7], the incorporation of regional information is often limited to a region immediately surrounding the voxel for which the cost is defined (especially in cases of multiple surface detection). In some applications, better cost functions could likely be defined if "true" regional information could be incorporated. This involves extending the definition of the cost of a set of surfaces to also include the summation of in-region cost values in addition to the on-surface cost values. The in-region cost value for a voxel associated with a particular region would reflect the unlikeliness of that voxel belonging to the region. Using the segmentation of the intraretinal layers of optical coherence tomography (OCT) images as an example application, this paper presents how true regional information can be incorporated into the 3-D graph search.

2 OCT Imaging Background

With its first introduction in 1991 [8], OCT is a relatively new imaging modality. One of its most common uses is within the ophthalmology community, where the high-resolution cross-sectional images resulting from OCT scanners are used for the diagnosis and management of a variety of ocular diseases such as glaucoma, diabetic macular edema, and optic neuropathy. The macula and region surrounding the optic nerve are two locations commonly scanned. For the images used in this work, a macular scanning protocol was used that involved the acquisition of six linear radial scans in a spoke pattern centered at the fovea (Fig. 1(a–c)). An example image series using this protocol is shown in Fig. 1(d).

Even though intraretinal layers are visible on these macular scans, current commercial systems (e.g., Stratus OCT-3, Carl Zeiss Meditec, Inc., Dublin, CA, USA) only segment and provide thickness measurements for the total retina. As each layer may be affected differently in the presence of ocular disease, an intraretinal layer segmentation approach is needed in order to correlate the individual layer properties with disease state. We have previously reported a method for the division of the retina into five intralayers (corresponding to six surfaces) using only on-surface costs in the graph search [7]. However, even though the graph search theoretically allowed for the simultaneous detection of many surfaces,



(a) Macular OD (b) Macular OS (c)

(d)

Fig. 1. Macular scanning protocol. (a) OD = right eye. (b) OS = left eye. (c) Schematic view of 3-D image in which each color reflects one 2-D scan. (d) Six example scans in one macular series.

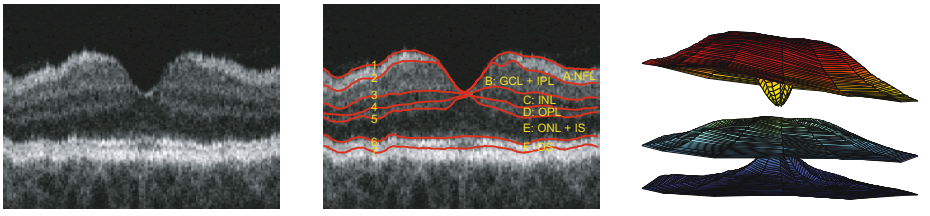
we found the three interior surfaces in a sequential fashion because of the limitations of incorporating regional information into on-surface costs. The method reported in this work for incorporating in-region costs allowed us to instead find four interior surfaces simultaneously. The surfaces we desired to find are shown in Fig. 2, with Fig. 2(c) providing an example 3-D view of three of the surfaces.

3 The Multiple Surface Segmentation Problem

In very general terms, the multiple surface segmentation problem can be thought of as an optimization problem with the goal being to find the set of surfaces with the minimum cost – such cost being edge and/or region based – so that the found surface set is feasible. In this section, we define what is meant by a feasible surface set and the cost of a set of surfaces.

3.1 Feasible Surface Set

Consider a volumetric image $I(x, y, z)$ of size $X \times Y \times Z$. We focus on the case in which each surface of interest can be defined with a function $f(x, y)$ mapping (x, y) pairs to z-values; however, note that the graph search can be extended to work with closed surfaces as well [9]. Associated with each (x, y) pair is a column of voxels in which only one of the voxels – the voxel at $(x, y, f(x, y))$ – intersects the surface. Each column also has a set of neighbors. For example, a



(a) (b) (c)

Fig. 2. Intraretinal surfaces and layers of macular OCT images. (a) One 2-D image. (b) Seven surfaces (labeled 1-7) and six corresponding intralayers (labeled A through F). (c) Example 3-D visualization of surface 1, 3, and 5.

typical set of neighbors for the column associated with (x, y) are the columns associated with $(x+1, y)$, $(x-1, y)$, $(x, y+1)$, and $(x, y-1)$. Other neighborhood relationships are also possible. One common example is to add a "circularity" neighbor relationship for images that were unwrapped from a cylindrical coordinate system. An example of this would be if the column associated with $(0, y_0)$ was considered to be a neighbor to the column associated with $(X-1, y_0)$. A single surface is considered feasible if it satisfies certain smoothness constraints. In particular, if (x_1, y_1, z_1) and (x_2, y_2, z_2) are voxels on the surface from neighboring columns in the x-direction, then $|z_1 - z_2| \leq \Delta_x$, where Δ_x is a specified smoothness parameter. A similar constraint exists for neighboring columns in the y-direction ($|z_1 - z_2| \leq \Delta_y$).

For a set of surfaces, additional constraints are added to model the desired relationships between the surfaces. For example, it may be known that one surface is always above another surface and that the distance between the surfaces is at least δ^l voxels, but no more than δ^u voxels. More specifically, for each pair of surfaces $f_i(x, y)$ and $f_j(x, y)$, a constraint may be added to require that $\delta^l \leq f_i(x, y) - f_j(x, y) \leq \delta^u$ for all (x, y), where δ^l and δ^u are specified surface interaction parameters associated with the pair of surfaces. While in general a pair of surfaces may be allowed to cross, having crossing surfaces does not make sense when defining regional costs, and thus we will also assume that δ^l and δ^u have the same sign.

In summary, a set of surfaces are considered feasible if each individual surface in the set satisfies the given smoothness constraints for the surface and if each pair of surfaces satisfies the surface interaction constraints.

3.2 Cost of a Feasible Surface Set

Given a set of n non-intersecting surfaces $\{f_1(x, y), f_2(x, y), \ldots, f_n(x, y)\}$, the surfaces naturally divide the volume into $n + 1$ regions (Fig. 3). Assuming the surfaces are labeled in "increasing" order, the regions can be labeled $R_0, \ldots, R_n$, where R_i reflects the region that lies between surface i and surface $i + 1$ (with region boundary cases R_0 and R_n being defined as the region with lower z-values

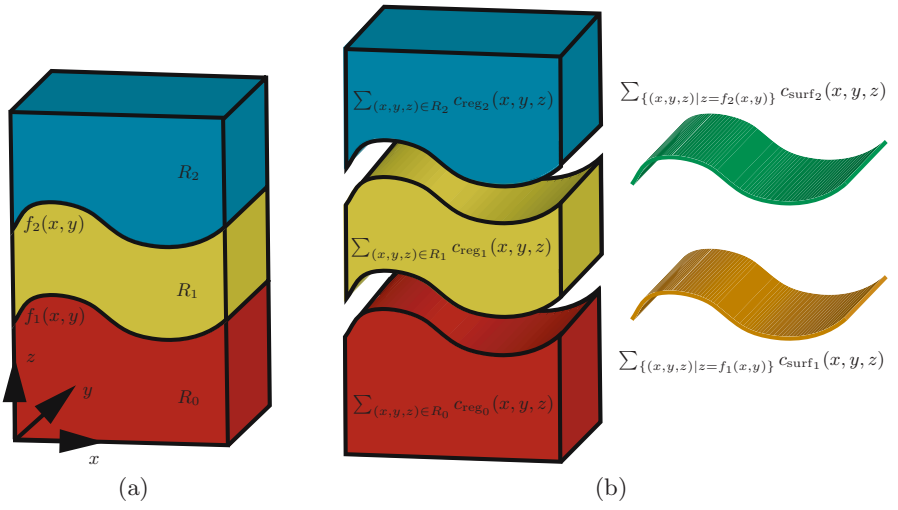


Fig. 3. Example schematic cost of two surfaces for the multiple surface segmentation problem. The two surfaces divide the volume into three regions.

than surface 1 and the region with higher z-values than surface n, respectively). Each voxel could thus have $2n+1$ real-valued costs associated with it: n on-surface costs corresponding to the unlikeliness of belonging to each surface and $n+1$ in-region costs associated with the unlikeliness of belonging to each region. Let $c_{\text{surf}_i}(x,y,z)$ represent the on-surface cost function associated with surface i and $c_{\text{reg}_i}(x,y,z)$ represent the in-region cost function associated with region i. Then, the cost $C_{\{f_1(x,y),f_2(x,y),\ldots,f_n(x,y)\}}$ associated with the set of surfaces can be defined as

$$C_{\{f_1(x,y),f_2(x,y),\ldots,f_n(x,y)\}} = \sum_{i=1}^{n} C_{f_i(x,y)} + \sum_{i=0}^{n} C_{R_i} \ , \tag{1}$$

where

$$C_{f_i(x,y)} = \sum_{\{(x,y,z)|z=f_i(x,y)\}} c_{\text{surf}_i}(x,y,z) \ , \tag{2}$$

and

$$C_{R_i} = \sum_{(x,y,z)\in R_i} c_{\text{reg}_i}(x,y,z) \ . \tag{3}$$

Note that $C_{f_i(x,y)}$ reflects the cost associated with voxels on surface i and C_{R_i} reflects the cost associated with voxels belonging to region i.

Thus, the multisurface segmentation problem becomes to find the surface set $\{f_1(x,y),\ f_2(x,y),\ \ldots,\ f_n(x,y)\}$ that minimizes Equation (1) such that each surface individually satisfies the smoothness constraints and each pair of surfaces satisfies the given surface interaction constraints.

4 Transforming the Multiple Surface Segmentation Problem into the Minimum-Cost Closed Set Problem

As was described in [3], a directed graph $G = (V, E)$ can be defined such that each feasible surface set corresponds to a closed set in the graph. A closed set is subset V_{CS} of the vertices V such that no edges leave the closed set. The cost of a closed set is the summation of the costs of all the vertices. Because of the direct correspondence between the vertices in the graph and voxels in the image, it is easiest to visualize the graph vertices as being organized as n volumes of vertices, one for each surface to be found. First, edges are added to each volume of vertices such that each closed set within this volume corresponds to a surface satisfying the given surface smoothness constraints. Essentially, the corresponding closed set includes all the vertices corresponding to the surface voxels plus all the "lower" vertices. This is done by adding two types of edges: intracolumn edges and intercolumn edges. The intracolumn edges ensure that all vertices below a given vertex (within one column) are also included in the closed set. The intercolumn edges ensure that the smoothness constraints are satisfied. Next, intersurface edges are added between the volumes of vertices to enforce the surface interaction constraints. This makes each non-empty closed set in the entire graph correspond to a feasible set of surfaces.

As an example, we will consider the added edges for one vertex associated with a voxel towards the center of the image (i.e., a vertex not involved in boundary conditions). It will be associated with two intracolumn directed edges: one directed towards the vertex below it in the column and one from the vertex above it. Two intercolumn edges will also exist for each neighboring column in the x-direction (y-direction): one directed to the vertex in the neighboring column that has a z-value that is Δ_x (Δ_y) smaller and one from the vertex in the neighboring column that has a z-value that is Δ_x (Δ_y) greater. Finally, for each corresponding column in the volume associated with a surface interaction constraint, two intersurface edges are associated with the vertex: one to the vertex in the corresponding column with a z-value that is δ^u smaller and one from the vertex in the corresponding column with a z-value that is δ^l smaller. Slightly different edges must be used in the "boundary cases" in which any of those vertices do not exist [3].

The cost of each vertex in the graph is set such that the cost of each closed set corresponds to the cost (within a constant) of the set of surfaces. The weight $w_i(x, y, z)$ of each vertex ($i = 1, 2, \ldots, n$) can be defined as the summation of a term related to the on-surface costs ($w_{\text{on-surf}_i}(x, y, z)$) and a term related to the in-region costs ($w_{\text{in-reg}_i}(x, y, z)$):

$$w_i(x, y, z) = w_{\text{on-surf}_i}(x, y, z) + w_{\text{in-reg}_i}(x, y, z) \; . \tag{4}$$

For on-surfaces costs, the basic idea is to assign the cost of each vertex the on-surface cost of the corresponding voxel minus the on-surface cost of the voxel

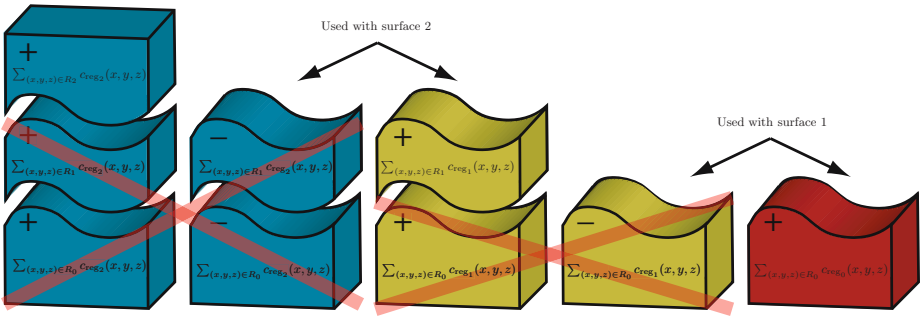


Fig. 4. Schematic showing how the assignment of in-region costs to vertices produces the desired overall cost

below it [4, 3]:

$$w_{\text{on}-\text{surf}_i}(x,y,z) = \begin{cases} c_{\text{surf}_i}(x,y,z) & \text{if} \quad z = 0 \\ c_{\text{surf}_i}(x,y,z) - c_{\text{surf}_i}(x,y,z-1) & \text{otherwise} \end{cases} . \tag{5}$$

For in-region costs, the cost of each vertex is assigned the in-region cost of the region below the surface associated with the vertex minus the in-region cost of the region above the surface associated with the vertex:

$$w_{\text{in}-\text{reg}_i}(x,y,z) = c_{\text{reg}_{i-1}}(x,y,z) - c_{\text{reg}_i}(x,y,z) . \tag{6}$$

Because the use of in-region costs is new and perhaps less intuitive, Fig. 4 illustrates why such a transformation works. The cost of the closed set $C(V_{CS_i})$ associated with surface i using the in-region costs becomes

$$C(V_{CS_i}) = \sum_{(x,y,z)\in R_0 \bigcup \cdots \bigcup R_{i-1}} c_{\text{reg}_{i-1}}(x,y,z) - \sum_{(x,y,z)\in R_0 \bigcup \cdots \bigcup R_{i-1}} c_{\text{reg}_i}(x,y,z) . \tag{7}$$

Recognizing that many of costs associated with each individual region cancel when added together and the fact that $\sum_{(x,y,z)\in R_0 \bigcup \cdots \bigcup R_n} c_{\text{reg}_n}(x,y,z)$ is a constant K, the cost for the closed set associated with the entire set of surfaces $C(V_{CS})$ reduces to

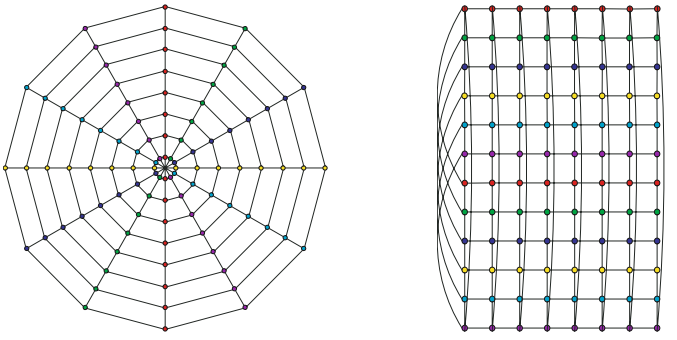
$$C(V_{CS}) = -K + \sum_{i=0}^{n} \sum_{(x,y,z)\in R_i} c_{\text{reg}_i}(x,y,z) , \tag{8}$$

which, within a constant, is equivalent to the desired in-region component of the cost of the set of surfaces.

5 Application to OCT Intraretinal Layer Segmentation

5.1 Overall Segmentation Approach

To increase the signal to noise ratio on the macular OCT images, up to six raw macular series were first aligned and registered together using the methods



(a) (b)

Fig. 5. Base graph showing neighborhood relationship. The edges indicate neighborhood connectivity of one column of z-values at a (r, θ) pair to another. For each edge shown, smoothness constraints existed between the corresponding columns. (a) Base graph using cylindrical coordinates. (b) Base graph using unwrapped coordinate system (as was stored in the computer).

described in [7]. This left one composite 3-D image for each eye. As a preprocessing step, a speckle-reducing anisotropic diffusion method was applied [10]. Boundaries 1, 6, and 7 were simultaneously found first using the graph search with only on-surface costs. The remaining boundaries were then simultaneously found using only in-region costs.

5.2 Surface Set Feasibility for Macular OCT Images

The acquisition setup of the macular scans (Fig. 1) made it natural to use a discrete cylindrical coordinate system when working with each 3-D image (the z-axis coincided with the intersection of the six 2-D composite scans). The coordinates of each voxel could thus be described with the triple (r, θ, z), where r reflected the distance of the voxel from the z-axis, θ reflected the angular location of the voxel (12 values in 30 degree increments), and z reflected the row of the voxel in the corresponding 2-D image. Note that with this coordinate system, voxels in the left half of each 2-D image had a different θ value than those in the right half (for example, for the vertical 2-D scan shown in red in Fig. 1, voxels in the right half of the image had a θ value of 90 while those in the left half had a θ value of 270). Each surface could be defined with a function $f(r, \theta)$, mapping (r, θ) pairs to z-values. The base graph in Fig. 5 schematically shows the neighborhood relationship for the columns and the corresponding smoothness constraints.

In addition, surface interaction constraints were added between each pair of surfaces $f_i(r, \theta)$ and $f_{i+1}(r, \theta)$. Because of the different nature of the surfaces near the fovea (layers often become much thinner), the surface interaction constraints towards the center of the image (r values less than 15) were correspondingly defined to allow for smaller distances between surfaces.

5.3 Surface Set Costs for Macular OCT Images

The on-surface cost functions for surfaces 1, 6, and 7 were the same as those used in our prior work [7] (each involved the combination of an edge term and a localized region-based term) and thus we will focus on the use of in-region cost terms for the simultaneous detection of the remaining four interior surfaces (surfaces 2, 3, 4, and 5).

Motivated by the observation that the intensity of each of the five interior regions could be described as being dark, medium, or bright (region A was bright, region B was medium, region C was dark, region D was medium, region E was dark), the in-region cost values were set based on fuzzy membership functions. Based on Gaussians, each membership function mapped a normalized image intensity value to a value between 0 and 1, with higher values reflecting a greater likelihood of belonging to the particular intensity group. The corresponding cost value was then defined as 1 minus the membership value. Fig. 6 shows an example plot of these membership functions and their corresponding cost values. More specifically, the dark membership function, $\text{dark}_{\text{mem}}(x)$, was defined as

$$\text{dark}_{\text{mem}}(x) = \begin{cases} 1 & \text{for } x \leq \Delta d \\ e^{-(x-\Delta d)^2/2\sigma^2} & \text{for } x > \Delta d \end{cases}, \tag{9}$$

the medium membership function, $\text{med}_{\text{mem}}(x)$, was defined as

$$\text{med}_{\text{mem}}(x) = \begin{cases} e^{-(x-(c_m-\Delta m))^2/2\sigma^2} & \text{for } x < c_m - \Delta m \\ 1 & \text{for } c_m - \Delta m \leq x \leq c_m + \Delta m \\ e^{-(x-(c_m+\Delta m))^2/2\sigma^2} & \text{for } x > c_m + \Delta m \end{cases}, \tag{10}$$

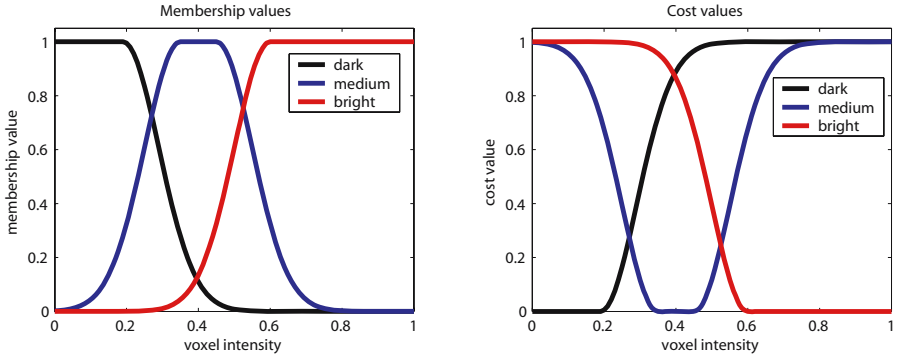
and the bright membership function, $\text{bright}_{\text{mem}}(x)$, was defined as

$$\text{bright}_{\text{mem}}(x) = \begin{cases} e^{-(x-(1-\Delta b))^2/2\sigma^2} & \text{for } x < 1 - \Delta b \\ 1 & \text{for } x \geq 1 - \Delta b \end{cases}. \tag{11}$$

Note that the precise membership functions used could be described by the five parameters Δd, Δm, and Δb, c_m, and σ. To allow for varying membership functions for each image, Δd, c_m, and Δb were estimated from the image data by computing the mean intensity value of regions that were assumed to have a dark, medium, or bright intensity. The assumed dark region was taken as 50–70 μm above surface 7, the assumed medium region was taken as a 40–60 μm below surface 1 (not including the middle voxels closest to the fovea), and the assumed bright region was taken as 0–24 μm below surface 7. These estimates could be taken because surfaces 1, 6, and 7 had already been determined.

6 Experimental Methods for OCT Intraretinal Segmentation

The intraretinal layer segmentation algorithm was applied to fast macular scans from 12 subjects with unilateral chronic anterior ischemic optic neuropathy. Note



(a) (b)

Fig. 6. Example dark, medium, and bright membership functions and corresponding cost values

that the unilateral nature of the disease meant that we had data for 24 eyes, 12 of which were affected by optic neuropathy, 12 of which were not. In almost all cases (21/24 eyes), six repeated series ($6 \times 6 = 36$ raw scans) were used to create the 3-D composite image for each eye. (Each of the remaining three eyes used fewer than six repeated series to create the 3-D composite image.) The resulting 24 3-D composite images were each comprised of 6 composite 2-D scans (144 total composite 2-D scans) of size 128 × 1024 pixels. The physical width and height of the 2-D raw scans (and thus also the composite scans) was 6 mm × 2 mm, resulting in a pixel size of approximately 50 μm (horizontally) × 2 μm (vertically).

One raw scan from each eye was independently traced by two human experts with the average of the two tracings being used as the reference standard. The experts did not attempt to trace borders that were not considered visible. The algorithmic result on the corresponding composite 2-D scan was converted into the coordinate system of the raw scan (inversely transforming the alignment/registration) and the mean and the maximum unsigned border positioning errors for each border were computed (the middle 30 pixels were not included to exclude the fovea). The unsigned border positioning errors were also computed using one observer as a reference standard for the other. For each border, a paired t-test was used to test for significant differences in the computed mean border positioning errors (p-values < 0.05 were considered significant).

7 OCT Intraretinal Segmentation Results

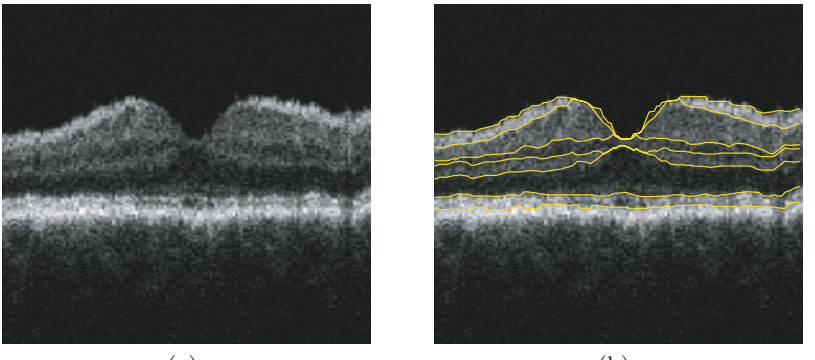
The computed unsigned and maximal border position errors are summarized in Table 1. Except for the unsigned border positioning errors for surface 2 and surface 4 (which both were significantly greater, $p < 0.001$ and $p = 0.04$, respectively), the computed mean errors for all the surfaces were significantly lower or not significantly different from that between the human observers ($p < 0.001$ for

Table 1. Summary of unsigned border positioning errors[†] for 24 scans

	Algorithm vs. Avg. Observer		Observer 1 vs. Observer 2	
Border	Mean	Maximum	Mean	Maximum
1	4.0 ± 1.2	16.9 ± 9.0	5.9 ± 1.3	16.4 ± 5.0
2[*]	11.2 ± 5.2	37.1 ± 11.9	5.8 ± 1.2	21.5 ± 8.6
3[*]	10.0 ± 4.7	29.0 ± 9.3	8.4 ± 3.3	26.0 ± 11.8
4[*]	10.4 ± 5.1	31.4 ± 14.3	7.7 ± 2.1	22.7 ± 6.6
5	9.1 ± 6.5	27.1 ± 13.2	9.4 ± 4.4	28.5 ± 12.5
6	3.5 ± 2.0	13.1 ± 7.5	7.8 ± 2.8	19.3 ± 5.6
7	7.8 ± 2.5	22.5 ± 7.2	11.5 ± 4.6	24.8 ± 5.8

[†] Mean ± SD in μm. For each boundary, differences were not computed for the middle 30 pixels (out of 128) to exclude the fovea.

[*] Errors were not computed for those scans in which boundary was determined to not be visible by at least one expert.



(a) (b)

Fig. 7. Example result shown on a 2-D scan from one of the 3-D images

surface 1; $p = 0.11$ for surface 3; $p = 0.80$ for surface 5; $p < 0.001$ for surface 6; $p = 0.004$ for surface 7). The overall mean (all borders combined) unsigned border positioning error for the algorithm was 7.8 ± 5.0 μm with an overall maximum unsigned border positioning error of 24.7 ± 12.9 μm. This was comparable to the overall mean and maximum border positioning errors computed between the observers which were 8.1 ± 3.6 μm and 22.8 ± 9.2 μm, respectively, and compared well with the true 9–10 μm resolution of the OCT imaging system reported in the literature [11]. An example result is shown in Fig. 7.

8 Discussion and Conclusion

We have presented how in-region cost terms may be added to the optimal 3-D graph search approach and demonstrated its applicability to the intraretinal

layer segmentation of macular OCT images. In fact, we chose to only use in-region cost terms for the interior surfaces to help to show how surfaces may be found using only in-region cost terms. With the resulting segmentation errors being similar to that found between two observers, our results were very good overall. Nevertheless, we anticipate that incorporating both on-surface and in-region cost terms will help to provide a better segmentation than that from using either type of cost alone.

References

1. McInerney, T., Terzoppoulos, D.: Deformable models in medical image analysis: A survey. Medical Image Analysis 1(2), 91–108 (1996)
2. Suri, J.S., Liu, K., Singh, S., Laxinarayan, S.N., Zeng, X., Reden, L.: Shape recovery algorithms using level sets in 2-D/3-D medical imagery: A state-of-the-art review. IEEE Trans. Inform. Technol. Biomed. 6(1), 8–28 (2002)
3. Li, K., Wu, X., Chen, D.Z., Sonka, M.: Optimal surface segmentation in volumetric images – a graph-theoretic approach. IEEE Trans. Pattern Anal. Machine Intell. 28(1), 119–134 (2006)
4. Wu, X., Chen, D.Z.: Optimal net surface problems wih applications. In: Widmayer, P., Triguero, F., Morales, R., Hennessy, M., Eidenbenz, S., Conejo, R. (eds.) ICALP 2002. LNCS, vol. 2380, pp. 1029–1042. Springer, Heidelberg (2002)
5. Boykov, Y.Y., Jolly, M.P.: Interactive graph cuts for optimal and region segmentation of objects in N-D images. In: Proc. of the Eighth IEEE International Conference on Computer Vision (ICCV), vol. 1, pp. 105–112. IEEE Computer Society, Washington, DC, USA (2001)
6. Zhao, F., Zhang, H., Wahle, A., Scholz, T.D., Sonka, M.: Automated 4D segmentation of aortic magnetic resonance images. In: British Machine Vision Conference (BMVA). vol. 1, pp. 247–256 (2006)
7. Haeker, M., Sonka, M., Kardon, R., Shah, V.A., Wu, X., Abràmoff, M.D.: Automated segmentation of intraretinal layers from macular optical coherence tomography images. In: Proceedings of SPIE Medical Imaging 2007: Image Processing. vol. 6512., SPIE (2007)
8. Huang, D., Swanson, E.A., Lin, C.P., Schuman, J.S., Stinson, W.G., Chang, W., Hee, M.R., Flotte, T., Gregory, K., Puliafito, C.A.: Optical coherence tomography. Science 254(5035), 1178–1181 (1991)
9. Li, K., Millington, S., Wu, X., Chen, D.Z., Sonka, M.: Simultaneous segmentation of multiple closed surfaces using optimal graph searching. In: Christensen, G.E., Sonka, M. (eds.) IPMI 2005. LNCS, vol. 3565, pp. 406–417. Springer, Heidelberg (2005)
10. Yu, Y., Acton, S.T.: Speckle reducing anisotropic diffusion. IEEE Trans. Image Processing 11(11), 1260–1270 (2002)
11. Costa, R.A., Skaf, M., Melo, L.A.S., Calucci, D., Cardillo, J.A., Castro, J.C., Huang, D., Wojtkowski, M.: Retinal assessment using optical coherence tomography. Prog. Retin. Eye Res. 25(3), 325–353 (2006)