



Published in final edited form as:

IEEE Trans Med Imaging. 2009 September ; 28(9): 1436–1447. doi:10.1109/TMI.2009.2016958.

Automated 3-D Intraretinal Layer Segmentation of Macular Spectral-Domain Optical Coherence Tomography Images

Mona Kathryn Garvin* [Member, IEEE],

Department of Electrical and Computer Engineering, The University of Iowa, Iowa City, IA, 52242 USA

Michael David Abramoff [Member, IEEE],

Department of Ophthalmology and Visual Sciences and the Department of Electrical and Computer Engineering, The University of Iowa, Iowa City, IA, 52242 USA. He is also with the VA Medical Center, Iowa City, IA 52246 USA (michael-abramoff@uiowa.edu)

Xiaodong Wu [Senior Member, IEEE],

Department of Electrical and Computer Engineering and the Department of Radiation Oncology, The University of Iowa, Iowa City, IA, 52242 USA (xiaodong-wu@uiowa.edu)

Stephen R. Russell,

Department of Ophthalmology and Visual Sciences, The University of Iowa, Iowa City, IA, 52242 USA (steve-rus-sell@uiowa.edu)

Trudy L. Burns, and

Department of Epidemiology, College of Public Health, The University of Iowa, Iowa City, IA, 52242 USA (trudy-burns@uiowa.edu)

Milan Sonka [Fellow, IEEE]

Department of Electrical and Computer Engineering, the Department of Ophthalmology and Visual Sciences, and the Department of Radiation Oncology, The University of Iowa, Iowa City, IA, 52242 USA (milan-sonka@uiowa.edu)

Abstract

With the introduction of spectral-domain optical coherence tomography (OCT), much larger image datasets are routinely acquired compared to what was possible using the previous generation of time-domain OCT. Thus, the need for 3-D segmentation methods for processing such data is becoming increasingly important. We report a graph-theoretic segmentation method for the simultaneous segmentation of multiple 3-D surfaces that is guaranteed to be optimal with respect to the cost function and that is directly applicable to the segmentation of 3-D spectral OCT image data. We present two extensions to the general layered graph segmentation method: the ability to incorporate varying feasibility constraints and the ability to incorporate true regional information. Appropriate feasibility constraints and cost functions were learned from a training set of 13 spectral-domain OCT images from 13 subjects. After training, our approach was tested on a test set of 28 images from 14 subjects. An overall mean unsigned border positioning error of $5.69 \pm 2.41 \mu\text{m}$ was achieved when segmenting seven surfaces (six layers) and using the average of the manual tracings of two ophthalmologists as the reference standard. This result is very comparable to the measured interobserver variability of $5.71 \pm 1.98 \mu\text{m}$.

© 2009 IEEE

*M. K. Garvin is with the Department of Electrical and Computer Engineering, The University of Iowa, Iowa City, IA, 52242 USA (monagarvin@uiowa.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>

Index Terms

Ophthalmology; optical coherence tomography; retina; segmentation; spectral-domain; three-dimensional (3-D) graph search

I. INTRODUCTION

Optical coherence tomography (OCT) is becoming an increasingly important modality for the noninvasive assessment of a variety of ocular diseases such as glaucoma, diabetic macular edema, and age-related macular degeneration. The previous generation of OCT images were acquired in the time domain, which only allowed up to approximately six images to be obtained during each scanning sequence (lasting no more than several seconds, to avoid eye motion artifacts). In the clinic, these images are typically evaluated individually, in 2-D, because the large interslice spacing prevents a reliable interpolation for 3-D display. With the recent development [1], [2] and commercialization of spectral-domain optical coherence tomography for use in ophthalmology, substantially more data can be acquired in the same time period for systems with the same sensitivity (e.g., the spectral-domain images used in this work contained $200 \times 200 \times 1024$ voxels, which is 52 times the number of voxels in a typical time-domain sequence with $6 \times 128 \times 1024$ voxels) [3]. As more data is acquired, the need for an automated 3-D segmentation approach is more urgent than before. However, most previously reported OCT segmentation approaches have been 2-D in nature (i.e., if multiple 2-D slices are available in a scanning sequence they are segmented independently) [4]–[11], or interactive [12].

Motivated by this need, we previously reported an automated approach for the 3-D segmentation of time-domain OCT images using a graph-theoretic segmentation technique [13]–[15]. The approach was based on the work by Li *et al.* [16] that allowed for the optimal and simultaneous segmentation of multiple 3-D surfaces by transforming the segmentation problem into the layered-graph-theoretic problem of finding a minimum-cost closed set in a vertex-weighted geometric graph. However, in its original formulation, the graph-theoretic approach did not allow for the incorporation of varying feasibility constraints or true regional information, two extensions which we have shown to be useful in our preliminary work for the segmentation of OCT images [17], [18]. Furthermore, and perhaps of more interest to the ophthalmology community, we had not yet applied our approach for the segmentation of spectral-domain images.

Thus, the major contributions of this work are two-fold.

1. We present a novel extended version of the general graph-based approach presented by Li *et al.* [16] that allows for the incorporation of varying feasibility constraints and true regional information. While such extensions were originally presented in a preliminary fashion in some of our prior work for the segmentation of time-domain images [17], [18], the developed methodology had not been fully utilized in the past (e.g., the sparse acquisition of the data did not allow for learning feasibility constraints in more than one direction and *both* edge and true regional information did not simultaneously contribute to the cost function terms). These extensions substantially increase the utility of the layered graph-based segmentation.
2. Based on these extensions, we present an automated approach for the segmentation of retinal layers in spectral-domain OCT images [1]–[3]. The ability to automatically segment the layers in such large volumetric datasets reflects a very important contribution to the ophthalmology community.

II. GRAPH-BASED SIMULTANEOUS SURFACE SEGMENTATION WITH VARYING CONSTRAINTS AND REGIONAL INFORMATION

A. Review of Optimal 3-D Graph Search Concepts

The optimal 3-D graph search approach originally presented in [16], [19] is designed to solve what we will call the “multiple surface segmentation problem.” An overview of the steps used in this approach is given in Fig. 1. In very general terms, the multiple surface segmentation problem can be thought of as an optimization problem with the goal to find the set of surfaces with the minimum cost such that the found surface set is feasible. Thus, there are two major components to the problem specification: 1) the specification of the constraints to require surface set feasibility and 2) the formalization of the cost of a set of surfaces. The first step in the graph search approach is to construct a graph the minimum-cost closed set of which corresponds to the set of surfaces with the minimum cost. (A closed set is a subset of the vertices of a graph such that no directed edges leave the set.) This is done by 1) ensuring that there is a one-to-one correspondence between each closed set in the constructed graph and each feasible surface set and 2) ensuring that the cost of each closed set in the graph corresponds (within a constant) to the cost of a set of (feasible) surfaces. Thus, it is the structure of the graph that reflects the feasibility constraints and vertex weights of the graph that reflect the cost functions. Finally, the actual minimum-cost closed set is found by finding minimum-cost s - t cut in a closely related graph [16], [19].

1) Original Limitations—In its original formulation, the 3-D optimal graph search method employed surface feasibility constraints that were constant in each direction. For example, the smoothness constraints for a particular surface $f(x, y)$ were represented by two parameters, Δ_x and Δ_y , reflecting the allowed change in surface height when moving from one neighboring surface point to the next in the x -direction and y -direction, respectively. Similarly, the surface interaction constraints (reflecting the allowed minimum and maximum distances between surface pairs) were constant. More flexibility in constraining surfaces to particular shapes would be obtained if varying constraints were allowed. Such a change would especially be important for surfaces in which the needed constraints are expected to change based on location (e.g., the foveal region in OCT images of the retina).

Furthermore, in its original formulation [16], [19], the cost of a set of surfaces was 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 reflected the unlikelihood that the voxel would be part of the surface). While such “on-surface” costs can incorporate both image edge and regional information (e.g., see [14], [15]), 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 unlikelihood of that voxel belonging to the region. We have extended the optimal layered graph search approach to incorporate varying constraints [18] and regional cost function terms [17], [20].

B. Extended Multiple Surface Segmentation Problem

1) Surface Set Feasibility With Varying Constraints—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. Associated with each (x, y) pair is a z -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. We use a four-neighbor

relationship in which the neighbors for the column associated with (x, y) are the columns associated with $(x + 1, y)$, $(x - 1, y)$, $(x, y + 1)$, and $(x, y - 1)$.

In the original formulation [16], [19], a single surface is considered feasible if the difference in z -values of neighboring surface points is less than or equal to a constant parameter (Δ_x in x -direction, Δ_y in y -direction). For example, for neighboring columns $\{(x_1, y_1), (x_2, y_2)\}$ in the x -direction, this requires that

$$-\Delta_x \leq f(x_1, y_1) - f(x_2, y_2) \leq \Delta_x. \quad (1)$$

A similar constraint exists for neighbors in the y -direction.

Here, we allow the smoothness constraints to vary as a function of the column neighborhood pair. For a given neighborhood pair $\{(x_1, y_1), (x_2, y_2)\}$, the constraints become

$$-\Delta_{\{(x_1, y_1), (x_2, y_2)\}}^u \leq f(x_1, y_1) - f(x_2, y_2)$$

and

$$f(x_1, y_1) - f(x_2, y_2) \leq \Delta_{\{(x_1, y_1), (x_2, y_2)\}}^l \quad (2)$$

where $\Delta_{\{(x_1, y_1), (x_2, y_2)\}}^u$ reflects the maximum allowed increase in z -value when moving on a surface from column (x_1, y_1) to column (x_2, y_2) and $\Delta_{\{(x_1, y_1), (x_2, y_2)\}}^l$ reflects the maximum allowed decrease in z -value.

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 (note the notational difference in δ used for surface interaction constraints and Δ used for smoothness constraints). Again, in the original formulation [16], these constraints were constant. Here, we allow these constraints to be a function of (x, y) so that a different interaction constraint can be used for each column. As an example, we will use the notation $\delta_{i-j}^l(x_0, y_0)$ ($\delta_{i-j}^u(x_0, y_0)$) to reflect the minimum (maximum) allowed distance between surface i and surface j for column (x_0, y_0) .

2) Cost of a Feasible Surface Set—Given a set of n non-intersecting surfaces $\{f_1(x, y), f_2(x, y), \dots, f_n(x, y)\}$, the surfaces naturally divide the volume into $n + 1$ regions (Fig. 2). Assuming the surfaces are labeled in “increasing” order, the regions can be labeled R_0, \dots, R_n , where R_j 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 than surface 1 and the region with higher z -values than surface n , respectively). Each voxel can thus have $2n + 1$ real-valued costs associated with it: n on-surface costs corresponding to the unlikelihood of belonging to each surface and $n + 1$ in-region costs associated with the unlikelihood 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 $C_{\{f_1(x, y), f_2(x, y), \dots, f_n(x, y)\}}$ associated with the set of surfaces can be defined as

$$C_{\{f_1(x, y), f_2(x, y), \dots, f_n(x, y)\}} = \sum_{i=1}^n C_{f_i(x, y)} + \sum_{i=0}^n C_{R_i} \quad (3)$$

where

$$C_{f_i(x,y)} = \sum_{\{(x,y,z)|z=f_i(x,y)\}} c_{\text{surf}_i}(x, y, z) \quad (4)$$

and

$$C_{R_i} = \sum_{(x,y,z) \in R_i} c_{\text{reg}_i}(x, y, z). \quad (5)$$

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 .

C. Transformation to Minimum-Cost Closed Set Problem

1) Graph Representation of Surface Set Feasibility—The structure of the constructed graph reflects the feasibility constraints. Recall that this means that there must be a one-to-one correspondence between each feasible surface set and a closed set in the constructed graph. The graph is constructed in a similar manner as reported in [16] with the exception that the edges of the graph must take into account the varying constraints. One subgraph is created for each surface to be found, with intracolumn and intercolumn edges being added to enforce the surface smoothness constraints [Fig. 3(a) and (b)]. The individual subgraphs are connected with intersurface edges to enforce the surface interaction constraints [Fig. 3(c) and (d)].

Let us consider the added edges for one vertex (x_1, y_1, z_1) 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 (x_2, y_2) : one directed to the vertex in the neighboring column that has a z -value that is $\Delta_{(x_1, y_1), (x_2, y_2)}^l$ smaller and one from the vertex in the neighboring column that has a z -value that is $\Delta_{(x_1, y_1), (x_2, y_2)}^u$ greater [Fig. 3(b)]. 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 $\delta^u(x_1, y_1)$ smaller and one from the vertex in the corresponding column with a z -value that is $\delta^l(x_1, y_1)$ smaller (assuming the surface for the given vertex is supposed to be “above” the interacting surface—see Fig. 3(c) and (d) with the blue surface subgraph being above the red surface subgraph; if not, the edge directions are reversed). Slightly different edges must be used in the boundary cases in which any of those vertices do not exist [16].

2) Graph Representation of Surface Set Costs—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 cost of a closed set is the summation of the costs of all the vertices.) The weight $w_i(x, y, z)$ of each vertex ($i = 1, 2, \dots, n$) 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 ($w_{\text{in-reg}_i}(x, y, z)$)

$$w_i(x, y, z) = w_{\text{on-surf}_i}(x, y, z) + w_{\text{in-reg}_i}(x, y, z). \quad (6)$$

For on-surface costs, the cost of each vertex is assigned the on-surface cost of the corresponding voxel minus the on-surface cost of the voxel below it [16], [19]

$$w_{\text{on-surf}_i}(x, y, z) = \begin{cases} c_{\text{surf}_i}(x, y, z), & \text{if } z=0 \\ c_{\text{surf}_i}(x, y, z) - c_{\text{surf}_i}(x, y, z-1), & \text{otherwise} \end{cases} \quad (7)$$

In addition, to ensure that a nonempty closed set is ultimately determined, the cost of one of the vertices of the graph is modified so that the summation of all of the base vertices is negative [16].

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

$$w_{\text{in-reg}_i}(x, y, z) = c_{\text{reg}_{i-1}}(x, y, z) - c_{\text{reg}_i}(x, y, z). \quad (8)$$

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

$$C(V_{CS_i}) = \sum_{(x,y,z) \in R_0 \cup \dots \cup R_{i-1}} c_{\text{reg}_{i-1}}(x, y, z) - \sum_{(x,y,z) \in R_0 \cup \dots \cup R_{i-1}} c_{\text{reg}_i}(x, y, z). \quad (9)$$

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

$$C(V_{CS}) = -K + \sum_{i=0}^n \sum_{(x,y,z) \in R_i} c_{\text{reg}_i}(x, y, z) \quad (10)$$

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

III. SPECTRAL OCT SEGMENTATION METHODS

Fig. 5 illustrates the seven surfaces we desired to find on macular spectral-domain OCT images. Even though the surfaces were the same as those segmented in our prior work for time-domain OCT images [14], [15], [17], [18], the larger coverage of the macular region created a greater variety of layer appearances (see example slices from a macular spectral-domain volumetric image in Fig. 6). The images were first flattened so that the surfaces near the RPE became approximately a flat plane (Section III-A). After flattening, the intraretinal layer segmentation of all seven surfaces was performed on a truncated full-resolution version of the 3-D image (Section III-B).

A. Flattening

Flattening the 3-D OCT volumes served multiple purposes: 1) to allow for smaller image sizes in the segmentation step, 2) to provide for a more consistent shape for segmentation purposes, and 3) to make visualization easier and conform to clinical practice. For example, having a consistent shape aided in both visualization and learning appropriate constraints for segmentation. In addition, flattening the image made it possible to truncate the image substantially in the axial direction (z -direction), thereby reducing the memory and time-requirements of the intraretinal layer segmentation approach. Note that the idea of flattening

OCT images has previously been reported by other groups as well (e.g., see [6] and [12]), mostly using 1-D or 2-D registration techniques; however in this work, we truly perform the flattening using a 3-D approach. Flattening an image involved the following steps:

- downsample image (by a factor of 4);
- simultaneously find surfaces 1, 6, and 7 on downsampled image using the optimal 3-D graph search approach;
- fit a (regularized) thin-plate spline [21] to an upsampled version of surface 7;
- translate all the columns of the (full-resolution) image so that the fitted surface becomes a flat plane;
- truncate the flattened image based on surface locations 1 and 7.

An example image before and after flattening is illustrated in Fig. 7.

B. Intraretinal Layer Segmentation

After applying a speckle-reducing anisotropic diffusion method [22], the surfaces on the flattened and truncated volumetric spectral images were segmented in two groups: surfaces 1, 6, and 7 were segmented simultaneously, followed by the simultaneous segmentation of surfaces 2, 3, 4, and 5. The cost functions used for surfaces 1, 6, and 7 were the same as those used in our prior work for time-domain images ([14], [15]). Thus, the cost functions for these surfaces were on-surface cost functions comprised of an edge-term (favoring a dark-to-bright transition using a Sobel kernel) and localized regional terms (cumulative term used for surface 1, terms favoring surfaces with a dark region above and bright region below used for surfaces 6 and 7).

The cost functions for the interior surfaces (surfaces 2, 3, 4, and 5) used a combination of on-surface and in-region cost functions. The on-surface cost functions for the interior surfaces were signed edge terms (using a Sobel kernel) to favor either a bright-to-dark transition (used for surfaces 2, 3, and 5) or a dark-to-bright transition (used for surface 4). The in-region cost terms were based on fuzzy membership functions in dark, medium, or bright intensity classes. Based on Gaussians, each membership function mapped a normalized image intensity 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. The dark membership function, $\text{dark}_{\text{mem}}(x)$, was defined as

$$\text{dark}_{\text{mem}}(x) = \begin{cases} 1, & \text{for } x \leq \Delta d \\ e^{-(x-\Delta d)^2/2\sigma^2}, & \text{for } x > \Delta d \end{cases} \quad (11)$$

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

$$\text{med}_{\text{mem}}(x) = \begin{cases} e^{-(x-(c_m-\Delta m))^2/2\sigma^2}, & \text{for } x < c_m - \Delta m \\ 1, & \text{for } c_m - \Delta m \leq x \leq c_m + \Delta m \\ e^{-(x-(c_m+\Delta m))^2/2\sigma^2}, & \text{for } x > c_m + \Delta m \end{cases} \quad (12)$$

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

$$\text{bright}_{\text{mem}}(x) = \begin{cases} e^{-(x-(1-\Delta b))^2/2\sigma^2}, & \text{for } x < 1 - \Delta b \\ 1, & \text{for } x \geq 1 - \Delta b. \end{cases} \quad (13)$$

The region between surfaces 1 and 2 used a bright membership function (adjustable by parameter Δb), the region between surfaces 2 and 3 used a medium membership function (adjustable by parameter c_m), the region between surfaces 3 and 4 used a dark membership function (adjustable by parameter Δd_1), the region between surfaces 4 and 5 used a medium membership function (adjustable by parameter c_m), and the region between surfaces 5 and 6 used a dark membership function (adjustable by parameter Δd_2). Note that there were actually two dark parameters, Δd_1 and Δd_2 , for the two dark regions.

The parameters Δb , c_m , and Δd_2 were automatically estimated for each volumetric image by computing the mean intensity in regions expected to belong to a specific layer. These regions were estimated by utilizing learned thickness constraints. For example, the RNFL (layer defined by surfaces 1 and 2) must contain the voxels between surface 1 and the surface defined by adding the minimum allowed distance between surfaces 1 and 2 to surface 1. The mean intensity in this region was thus used for estimating $(1 - \Delta b)$. Table I provides a summary of the bounding surfaces for each region. Columns for which the top surface would be located below the bottom surface were excluded from the estimation region. Δd_1 was estimated based on finding the relationship between Δd_2 and Δd_1 using training data (i.e., $\Delta d_1 = \Delta d_2 + c$, where c was a constant determined from the training data). The other parameters, σ and Δm , were set to reasonable fixed values based on our prior experiments with time-domain segmentations [17], [18].

C. Learning the Feasibility Constraints

Complete edited segmentations from the training set (see Section IV-B) were used for learning the surface interaction (thickness) and smoothness constraints for the interior surfaces. In general, the varying smoothness constraints for each pair of neighboring columns $\{(x_1, y_1), (x_2, y_2)\}$ in each surface were determined as follows. First, the mean and standard deviation of how the z -value changes when moving from column (x_1, y_1) to column (x_2, y_2) were determined. Let \bar{d} reflect the mean deviation (i.e., the mean of $f(x_1, y_1) - f(x_2, y_2)$ from the reference standard) and σ reflect the standard deviation. To allow for 99% of the expected changes in z -value when moving from column (x_1, y_1) to column (x_2, y_2) (assuming a normal distribution), the two parameters of the smoothness constraint, $\Delta_{\{(x_1, y_1), (x_2, y_2)\}}^l$ and $\Delta_{\{(x_1, y_1), (x_2, y_2)\}}^u$ were set as follows:

$$\Delta_{\{(x_1, y_1), (x_2, y_2)\}}^l = \bar{d} + 2.6 * \sigma \quad (14)$$

and

$$\Delta_{\{(x_1, y_1), (x_2, y_2)\}}^u = -(\bar{d} - 2.6 * \sigma). \quad (15)$$

Using a four-neighbor relationship, this meant that constraints were computed for each pair of neighboring columns in the x -direction $\{(x_1, y_1), (x_1 + 1, y_1)\}$ and y -direction $\{(x_1, y_1), (x_1, y_1 + 1)\}$.

The surface interaction constraints were learned in a similar fashion. For each surface pairing, the minimum allowed distance between the surfaces at each column was set as the mean thickness minus 2.6 times the standard deviation (but truncated so as not to go below 0). The value 2.6 was chosen to allow for 99% of the expected thickness values. The maximum allowed distance between the surfaces at each column was set as the mean thickness plus 2.6 times the standard deviation.

D. Learning the Cost Function Parameters

As mentioned in Section III-B, most of the in-region parameters were determined automatically in a subject-specific fashion during the segmentation method by using the learned thickness constraints to determine the regions of interest for estimation. However, the relationship between the two dark parameters was estimated based on the training data (see Section IV-B). In particular, it was assumed that this relationship could be described as $\Delta d_1 = \Delta d_2 + c$. The constant c was estimated by taking the difference of the mean intensity of the two dark regions based on the $10 \times 13 = 130$ manually traced slices (before segmentation/editing) in the training set.

To learn an appropriate combination of the on-surface and in-region cost functions for the simultaneous segmentation of surfaces 2, 3, 4, and 5, first the overall cost function was described using one parameter (α) to indicate the relative weight of the on-surface and in-region cost terms ($0 \leq \alpha \leq 1$)

$$C_{\{f_2(x,y), f_3(x,y), \dots, f_5(x,y)\}} = \alpha \sum_{i=2}^5 C_{f_i(x,y)} + (1 - \alpha) \sum_{i=1}^5 C_{R_i} \quad (16)$$

where

$$C_{f_i(x,y)} = \sum_{\{(x,y,z) | z=f_i(x,y)\}} c_{\text{surf}_i}(x, y, z) \quad (17)$$

and

$$C_{R_i} = \sum_{(x,y,z) \in R_i} c_{\text{reg}_i}(x, y, z). \quad (18)$$

With this description, $\alpha = 1$ corresponded to only using on-surface cost functions and $\alpha = 0$ corresponded to only using in-region cost terms. Various values of α were tested on images in the training set. For each value of α , the mean unsigned border positioning error from the bronze standard (Section IV-B) for each surface was computed. Values of α in increments of 0.1 were first tested, followed by using smaller increments as deemed necessary for obtaining the best value.

Next, in order to allow for a different weighting factor for each surface, the cost function for the interior surfaces was written as

$$C_{\{f_2(x,y), \dots, f_5(x,y)\}} = \sum_{i=2}^5 c_i C_{f_i(x,y)} + \sum_{i=1}^5 C_{R_i} \quad (19)$$

where each c_i was the constant weighting factor used for the on-surface cost term $C_{f_i(x,y)}$ for surface i . Note that in this formulation the weighting factor for the in-region costs was set to 1 so that only the on-surface weights would change. Letting α_i be the best value of α for surface i , the weighting c_i for each on-surface cost function was set as follows:

$$c_i = \begin{cases} \alpha_i / (1 - \alpha_i), & \text{if } 1 - \alpha_i > 0 \\ C_{\text{large}}, & \text{if } 1 - \alpha_i = 0 \end{cases} \quad (20)$$

where C_{large} is a relatively large constant value. In this formulation, the desired ratio (determined from α_i) of the on-surface costs to the in-region costs was preserved for each

surface. However, the relative importance of each surface could change with respect to the others if the values of α varied greatly amongst the surfaces. Thus, this cost function formulation was run on the cases in the training set to ensure that it in fact produced a smaller overall error. [If not, then only one value of α would be used as in (16).]

IV. EXPERIMENTAL METHODS

A. Subject Data

Macula-centered 3-D OCT volumes ($200 \times 200 \times 1024$ voxels, $6 \times 6 \times 2$ mm³) were obtained from the right eyes of 27 normal subjects using three technically identical Cirrus HD-OCT machines (Carl Zeiss Meditec, Inc., Dublin, CA). The data (41 volumetric scans) were organized into two groups.

- *Training set:* a 3-D OCT volume of the right eye of each subject 1–13 was acquired using one machine (machine A). These volumetric scans were used for training the algorithm.
- *Test set:* a 3-D OCT volume of the right eye of each subject 14–27 was acquired twice using two different machines (machine B and machine C). These 28 volumetric scans were used for testing the algorithm.

Consequently, the training and testing sets were completely disjoint.

B. Reference Standards

Fig. 8 illustrates the use of reference standards in the training and testing process. Two reference standards were used: a “bronze” standard (used for training) and a “gold” standard (used for testing). The bronze standard involved creating complete single-observer-edited segmentations for all 13 volumes in the training set. The gold standard was created by having two retinal specialists trace 10 slices on each volume in the test set and then taking the average of these tracings as the reference standard. To prevent an unintended bias from influencing the final algorithm, no segmentations were performed on the test set until the training was complete.

1) Bronze Standard (Used for Training)—In creating the bronze standard, each volume was first divided into 10 regions (illustrated in Fig. 9) and one slice was randomly selected in each region. Using custom-designed manual tracing software, the surfaces were manually traced on each of the 10 slices. Although only 10 slices were traced in each volume, all 200 slices were available to provide 3-D context (typically viewed in a cine-like mode). In addition, for better visualization, the software allowed for translating the columns to flatten the image to a manually traced boundary of interest (typically boundary 6 or 7).

After the boundaries were manually defined on each of the selected slices, the volumes were flattened to the RPE by fitting a thin-plate spline to this manually traced surface and then translating the columns so that this surface became a flat plane. Next, using the manual tracings as anchor points, a preliminary version of the segmentation approach was performed to segment the remaining slices. The “anchoring” was done by modifying the on-surface cost functions to have a very large value for all pixels in the selected slices except for the pixels on the anchored surface. Because the constraints had not been learned yet, the smoothness and surface interaction constraints were set to values more permissive than were expected to be necessary.

The resulting surfaces were reloaded back into the manual tracings application to be edited on the flattened images. The final complete edited segmentations defined the bronze standard.

2) Gold Standard (Used for Testing)—For each 3-D OCT volume, 10 random slices were traced independently by two ophthalmologists (retinal specialists). The slices were selected from the same regions as was done for the bronze standard (illustrated in Fig. 9). The average of these tracings defined the gold standard.

C. Validation on the Test Set

After training was complete (learning the constraints and cost functions), the final algorithm was run on all images in the test set. Signed and unsigned border positioning differences were computed for the following comparisons (over the 10 selected slices for each volume):

- ophthalmologist 1 versus ophthalmologist 2;
- the algorithm versus the average of ophthalmologists 1 and 2 (gold standard);
- the algorithm versus ophthalmologist 1;
- the algorithm versus ophthalmologist 2.

In order to compare the interobserver variability of the ophthalmologists with the errors of the algorithm, differences in the “errors” (e.g., comparing the border positioning differences between the two ophthalmologists to the border positioning differences between algorithm and the gold standard) for each border were fitted using a univariate analysis of variance (ANOVA) model. The models were fitted using the GLM (general linear models) procedure from the Statistical Analysis System (SAS) version 9.1. Resulting p -values were adjusted using the SIMULATE option for pairwise comparisons among least squares method-specific means that were estimated using the GLM procedure. This adjustment approach provided family-wise error rate protection by computing adjusted p -values based on simulated data. The type I error rate was specified to be 0.05.

V. RESULTS

A. Training Set

1) Learned Constraints—As an example, Fig. 10 illustrates the resulting learned thickness (surface interaction) constraints for the layer bounded by surfaces 2 and 3. Example learned smoothness constraints in the x -direction (the maximum allowed decrease of z -value in the x -direction and the maximum allowed increase of z -value in the x -direction) for surface 2 are illustrated in Fig. 11. As a point of clarification, “increasing” or “decreasing” refers to increasing or decreasing the z -value when moving from one column to the next (e.g., moving from column (x, y_j) to column $(x + 1, y_j)$ for the x -direction constraints) in a coordinate system in which $z = 0$ was at the *top* of the images.

2) Cost Function Combination—Fig. 12 summarizes the computed unsigned positioning errors for each tested value of the on-surface—in-region cost weight α . Table II summarizes the resulting best values of α for each surface. The last column of this table provides the ratio of α_j to $1 - \alpha_j$, which was used as the on-surface cost weighting term [(19) and (20)].

B. Test Set

1) Example Results—Qualitatively, the algorithm performed well overall, with only minor local inaccuracies. The one exception might be surface 7, as it would sometimes “flip” between two probable surfaces in a localized region. Fig. 13 illustrates the segmented surfaces on five slices (used for validation) of the spectral image with the median overall unsigned border positioning error.

2) Border Positioning Errors—The resulting unsigned and signed border positioning errors are given in Tables III and IV, respectively. The overall mean unsigned border positioning error was $5.71 \pm 1.98 \mu\text{m}$ for the comparison between the two ophthalmologists, $5.69 \pm 2.41 \mu\text{m}$ for the comparison between the algorithm and the average ophthalmologist, $6.73 \pm 2.48 \mu\text{m}$ for the comparison of the algorithm versus ophthalmologist 1, and $5.89 \pm 2.74 \mu\text{m}$ for the comparison of the algorithm versus ophthalmologist 2. When looking at each border individually (using separate ANOVA models and adjusting the p -values to take into account the multiple comparisons), the unsigned errors of the algorithm (when compared to the gold standard defined as the average tracings of the two ophthalmologists) were significantly smaller than those between the two ophthalmologists for border 1 ($p < 0.001$), border 2 ($p = 0.01$), and border 5 ($p < 0.001$). The unsigned errors of the algorithm were significantly larger than those between the two ophthalmologists for border 4 ($p = 0.03$) and border 7 ($p = 0.04$) but the differences were small ($7.79 \pm 0.74 \mu\text{m}$ [algorithm] versus $7.06 \pm 1.41 \mu\text{m}$ [ophthalmologists] for border 4, 8.47 ± 2.29 [algorithm] versus 7.11 ± 2.07 [ophthalmologists] for border 7). The unsigned border positioning errors for border 3 and border 6 were not significantly different between the algorithm and the two ophthalmologists.

VI. DISCUSSION

A. Accuracy

With respect to accuracy, the unsigned border positioning errors were comparable to those of the two ophthalmologists (overall mean error of $5.69 \pm 2.41 \mu\text{m}$ for the algorithm versus the average of the ophthalmologists and $5.71 \pm 1.98 \mu\text{m}$ for differences between the two ophthalmologists).

It is important to note that the time required for each ophthalmologist to manually trace the scans was substantial—a minimum of 30 min was required to trace the 10 slices of each volume, but many volumes took longer than 30 min to trace. Thus, it would not be feasible to have an ophthalmologist trace all 200 slices in order to perform a thickness assessment (requiring approximately 10 h of devoted tracing time per dataset). In addition, even if an ophthalmologist (or other expert) were willing to trace more slices, it is likely that the accuracy could suffer with the additional slices to trace. Thus, having an automated (or semi-automated) approach is necessary if quantitative assessments based on the surfaces are desired.

For both the algorithm and the ophthalmologist, the position of surface 7 was difficult to define. This was because two boundaries (interfaces) were sometimes visible and it was not clear which boundary was the appropriate one. Thus, the algorithm would sometimes find a surface that would transition between both boundaries within each slice; whereas each ophthalmologist would stick to one boundary for a particular slice, but would sometimes pick the other boundary on the repeated scan (or in other slices within the volume). In future work, we plan on segmenting additional layers in this distal region to help alleviate this issue.

B. Reducing Memory and Time Requirements

The large amount of data in spectral OCT datasets makes the time and memory requirements more of an issue when compared to segmenting time-domain OCT images. In fact, the methods had to be implemented to run on a 64-bit operating system so that up to approximately 10 GB of RAM could be used during the segmentation process. Furthermore, the time requirements were substantial (requiring hours of processing time per volume). However, there are multiple ways the memory and time requirements could be reduced.

For example, one way of reducing the required processing time would be to try a different $s-t$ cut approach in the graph search (in this work we used an approach based on that developed by Boykov [23]). In fact, we have performed some preliminary experiments that indicate that substantial amounts of time could be saved by just changing this approach (e.g., for a case with a small number of slices, a reduction in running time from 2.5 h to 5 min was achieved by changing from Boykov's method to a push-relabel method). Using a parallel implementation once its development by Boykov's group is completed [24] would be another way of improving the processing time.

In addition, there were a number of redundancies in the approach which provide additional opportunities of reducing the time requirements. For example, after the preliminary segmentation of surfaces 1, 6, and 7 was performed on the downsampled volume, this information was only used for flattening the image, but not used in the later segmentation steps. It may make sense to reuse this preliminary segmentation in order to reduce the size of the graph required in later steps (e.g., by defining a region of interest based on the preliminary result for the full-resolution segmentation). Interestingly, it was the segmentation of the first set of surfaces (1, 6, and 7) that took the most time (the segmentation of the interior surfaces took minutes instead of hours), so utilizing the preliminary segmentation would provide a substantial savings. Furthermore, reducing the size of the graph in this way would save on memory requirements as well. It may also be wise to try a more general multiresolution approach in order to save on time and memory requirements (e.g., by performing the segmentation on more than just the two resolutions used in this work). Even though the current implementation was relatively slow, our preliminary experience indicates that with some of the above modifications, the segmentation time could reasonably be expected to run in minutes or seconds instead of hours.

C. Importance of Intraretinal Layer Segmentation

Intraretinal layer segmentation is important for two main reasons. First, intraretinal segmentation will aid in an improved understanding of the pathophysiology of systemic diseases, especially those that also affect the brain. The retina is unique as a layered tissue: it is easily visible and also allows visualization of both the central nervous system and microvascular system. Some inner retinal layers have the same origin as brain cortical tissue, and damage to the nerve fiber layer has already been found to be a proxy for brain damage in multiple sclerosis, diabetes [25], and congenital connective tissue disease. Improved segmentation of retina layers can be expected to allow researchers to find retinal proxies for other systemic diseases, as well as serve as quantitative traits for diseases that are partially genetic in origin.

Second, intraretinal segmentation allows improved quantification of disease stages, because it can measure differences almost at the cellular level. For example, in glaucoma, retinal ganglion cells die, and the loss of axons can be quantified by measuring the nerve fiber layer thickness. Improved intraretinal segmentation will help clinicians better measure the effect of eye disease treatment.

VII. CONCLUSION

In addressing the need for a 3-D segmentation method for finding the intraretinal layers in macular spectral-domain OCT images, we have presented a method that allows for the optimal (with respect to a cost function) simultaneous segmentation of the intraretinal layer surfaces, achieving results comparable to those of two ophthalmologists. While the focus of this paper has been on the segmentation of spectral-domain OCT images, the two presented

extensions to the graph-theoretic approach (incorporating varying constraints and true regional information) are general and thus suitable for other applications.

Acknowledgments

This work was supported in part by the National Institutes of Health—National Institute of Biomedical Imaging and Bioengineering (NIH-NIBIB) under Grant R01-EB004640-01, in part by the National Institutes of Health—National Eye Institute (NIH-NEI) under Grant R01-EY017066, in part by the National Institutes of Health—National Cancer Institute (NIH-NCI) under K25-CA123112, in part by the National Science Foundation under Grant CCF-0830402, and in part by an unrestricted grant from the Research to Prevent Blindness, New York, NY.

REFERENCES

1. Wojtkowski M, Leitgeb R, Kowalczyk A, Bajraszewski T, Fercher AF. In vivo human retinal imaging by Fourier domain optical coherence tomography. *J. Biomed. Opt.* 2002 Jul; vol. 7(no. 3): 457–463. [PubMed: 12175297]
2. de Boer JF, Cense B, Park BH, Pierce MC, Tearney GJ, Bouma BE. Improved signal-to-noise ratio in spectral-domain compared with time-domain optical coherence tomography. *Opt. Lett.* 2003 Nov; vol. 28(no. 21):2067–2069. [PubMed: 14587817]
3. Drexler W, Fujimoto JG. State-of-the-art retinal optical coherence tomography. *Progress Retinal Eye Res.* 2008; vol. 27(no. 1):45–88.
4. Koozekanani D, Boyer K, Roberts C. Retinal thickness measurements in optical coherence tomography using a Markov boundary model. *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* 2000 Jun. vol. 2:363–370.
5. Koozekanani D, Boyer K, Roberts C. Retinal thickness measurements from optical coherence tomography using a Markov boundary model. *IEEE Trans. Med. Imag.* 2001 Sep; vol. 20(no. 9): 900–916.
6. Ishikawa H, Stein DM, Wollstein G, Beaton S, Fujimoto JG, Schuman JS. Macular segmentation with optical coherence tomography. *Invest. Ophthalmol. Vis. Sci.* 2005 Jun; vol. 46(no. 6):2012–2017. [PubMed: 15914617]
7. Chan A, Duker JS, Ishikawa H, Ko TH, Schuman JS, Fujimoto JG. Quantification of photoreceptor layer thickness in normal eyes using optical coherence tomography. *Retina.* 2006; vol. 26(no. 6): 655–660. [PubMed: 16829808]
8. Fernández DC. Delineating fluid-filled region boundaries in optical coherence tomography images of the retina. *IEEE Trans. Med. Imag.* 2005 Aug; vol. 24(no. 8):929–945.
9. Fernández DC, Salinas HM, Puliafito CA. Automated detection of retinal layer structures on optical coherence tomography images. *Opt. Express.* 2005; vol. 13(no. 25):10200–10216.
10. Shahidi M, Wang Z, Zelkha R. Quantitative thickness measurement of retinal layers imaged by optical coherence tomography. *Am. J. Ophthalmol.* 2005 Jun; vol. 139(no. 6):1056–1061. [PubMed: 15953436]
11. Baroni M, Fortunato P, Torre AL. Towards quantitative analysis of retinal features in optical coherence tomography. *Med. Eng. Phys.* 2007 May; vol. 29(no. 4):432–441. [PubMed: 16860587]
12. Fuller A, Zawadzki R, Choi S, Wiley D, Werner J, Hamann B. Segmentation of three-dimensional retinal image data. *IEEE Trans. Vis. Comput. Graphics.* 2007 Nov-Dec; vol. 13(no. 6):1719–1726.
13. Haeker, M.; Abràmoff, MD.; Kardon, R.; Sonka, M. Segmentation of the surfaces of the retinal layer from OCT images; Proceedings of the 9th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI 2006), Part I; 2006. p. 800-807.
14. Haeker M, Sonka M, Kardon R, Shah VA, Wu X, Abràmoff MD. Automated segmentation of intraretinal layers from macular optical coherence tomography images, in. *Proc. SPIE Med. Imag.* 2007: Image Process. 2007; vol. 6512
15. Garvin MK, Abràmoff MD, Kardon R, Russell SR, Wu X, Sonka M. Intraretinal layer segmentation of macular optical coherence tomography images using optimal 3-D graph search. *IEEE Trans. Med. Imag.* 2008 Oct; vol. 27(no. 10):1495–1505.

16. Li K, Wu X, Chen DZ, Sonka M. Optimal surface segmentation in volumetric images—A graph-theoretic approach. *IEEE Trans. Pattern Anal. Mach. Intell.* 2006 Jan; vol. 28(no. 1):119–134. [PubMed: 16402624]
17. Haeker, M.; Wu, X.; Abràmoff, MD.; Kardon, R.; Sonka, M. *Information Processing in Medical Imaging (IPMI)*. Vol. vol. 4584. New York: Springer; 2007. Incorporation of regional information in optimal 3-D graph search with application for intraretinal layer segmentation of optical coherence tomography images, in; p. 607-618. *Lecture Notes Computer Science*
18. Haeker, M.; Abràmoff, MD.; Wu, X.; Kardon, R.; Sonka, M. Use of varying constraints in optimal 3-D graph search for segmentation of macular optical coherence tomography images; *Proceedings of the 10th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI 2007)*; 2007. p. 244-251. *Lecture Notes Computer Science*
19. Wu, X.; Chen, DZ. Optimal net surface problems with applications; *Proceedings of the 29th International Colloquium on Automata, Languages, and Programming (ICALP), LNCS 2380*; p. 1029-1042.
20. Wu X, Chen DZ, Li K, Sonka M. The layered net surface problems in discrete geometry and medical image segmentation. *Int. J. Comput. Geometry Appl.* 2007; vol. 17(no. 3):261–296.
21. Donato, G.; Belongie, S. In: Heyden, A.; Sparr, G.; Nielsen, M.; Johansen, P., editors. *Approximate thin plate spline mappings*; *Proceedings of the 7th European Conference on Computer Vision (ECCV 2002), Part III, LNCS 2352*; New York: Springer. 2002. p. 21-31.
22. Yu Y, Acton ST. Speckle reducing anisotropic diffusion. *IEEE Trans. Image Process.* 2002 Nov; vol. 11(no. 11):1260–1270. [PubMed: 18249696]
23. Boykov YY, Jolly M-P. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images, in. *Proc. 8th IEEE Int. Conf. Comput. Vis. (ICCV)*. 2001; vol. 1:105–112.
24. DeLong A, Boykov Y. A scalable graph-cut algorithm for N-D grids, in. *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*. 2008:1–8.
25. van Dijk HW, Kok PHB, Garvin MK, Sonka M, Michels RPJ, Schlingemann RO, Verbraak FD, Abràmoff MD. Selective loss of inner retinal layer thickness in type 1 diabetic patients with minimal diabetic retinopathy. *Invest. Ophthalmol. Vis. Sci.* 2009; vol. 50:3404–3409. [PubMed: 19151397]

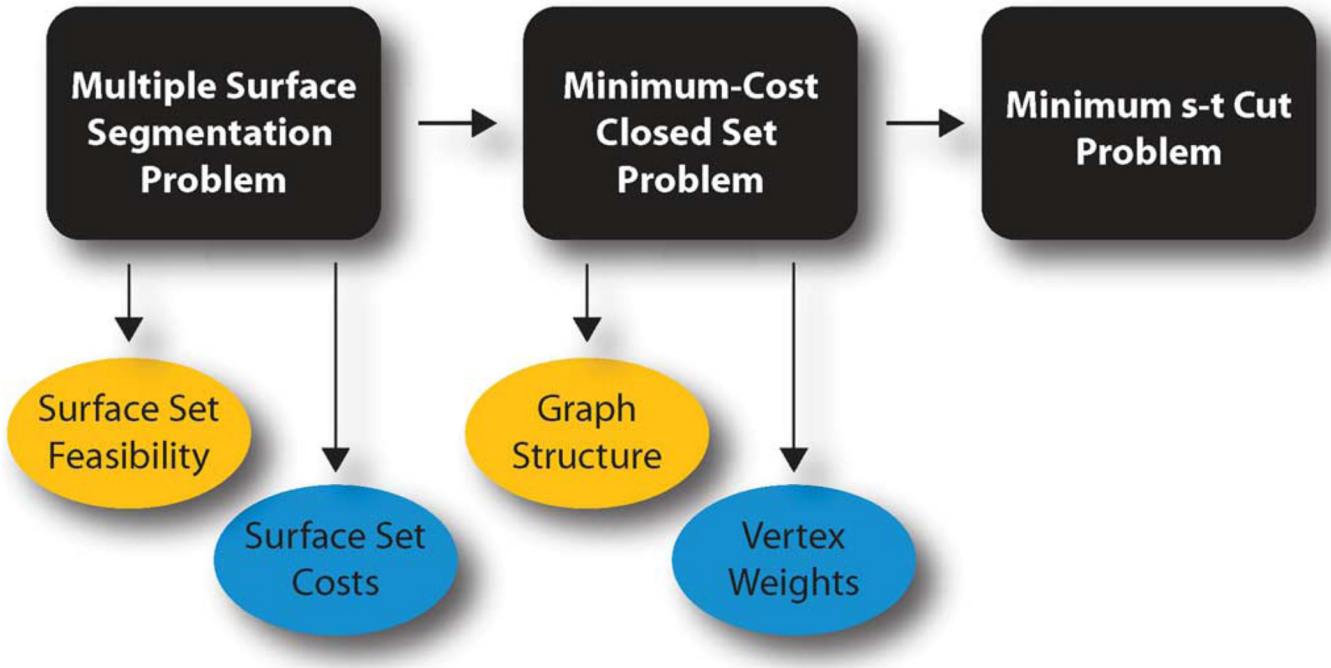


Fig. 1. Review of optimal 3-D graph search approach. The multiple surface segmentation problem is transformed into the graph-theoretic problem of finding a minimum-cost closed set in a geometric graph. The graph is constructed so that the structure of the graph reflects the feasibility constraints and the vertex weights of the graph reflect the cost function. The minimum-cost closed set is found in the constructed graph by finding a minimum $s-t$ cut in a closely related graph.

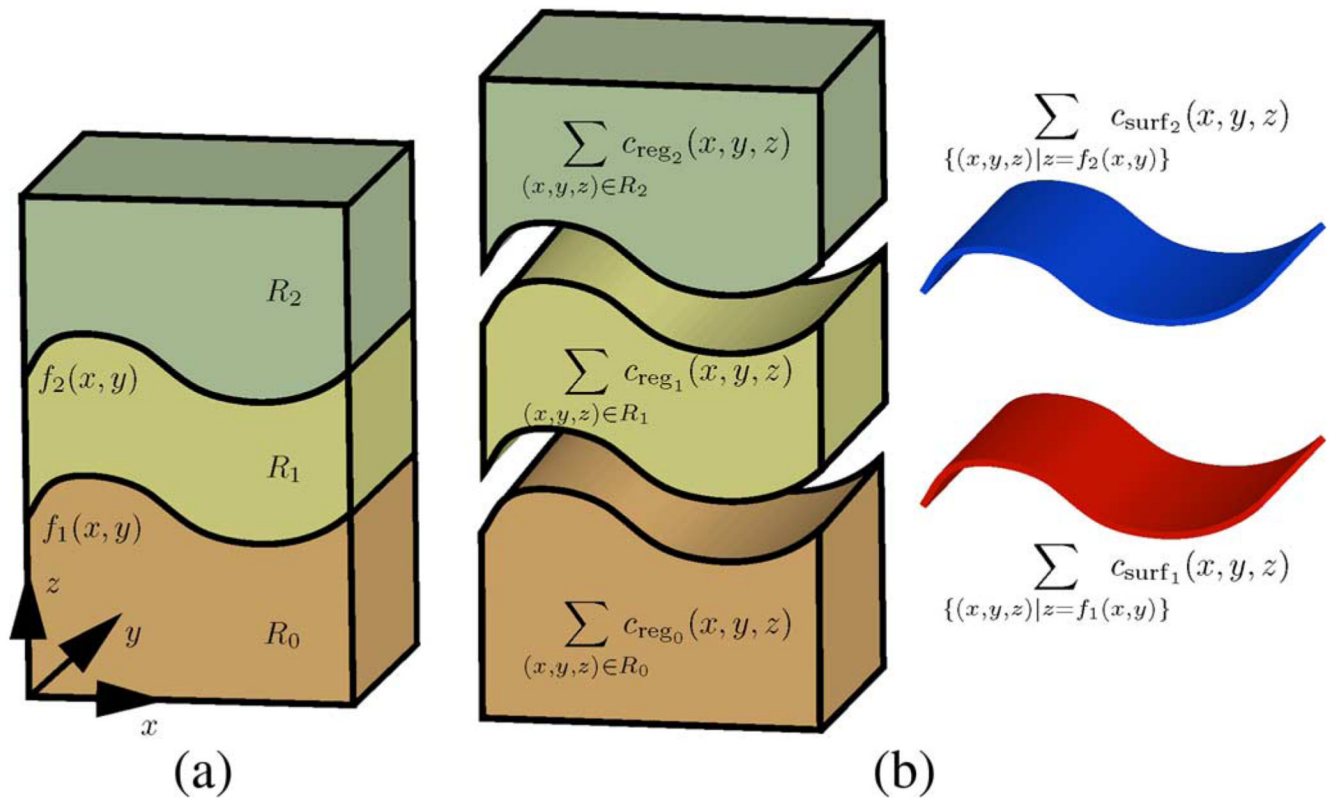


Fig. 2. Example schematic cost of two surfaces for the multiple surface segmentation problem. The two surfaces divide the volume into three regions.

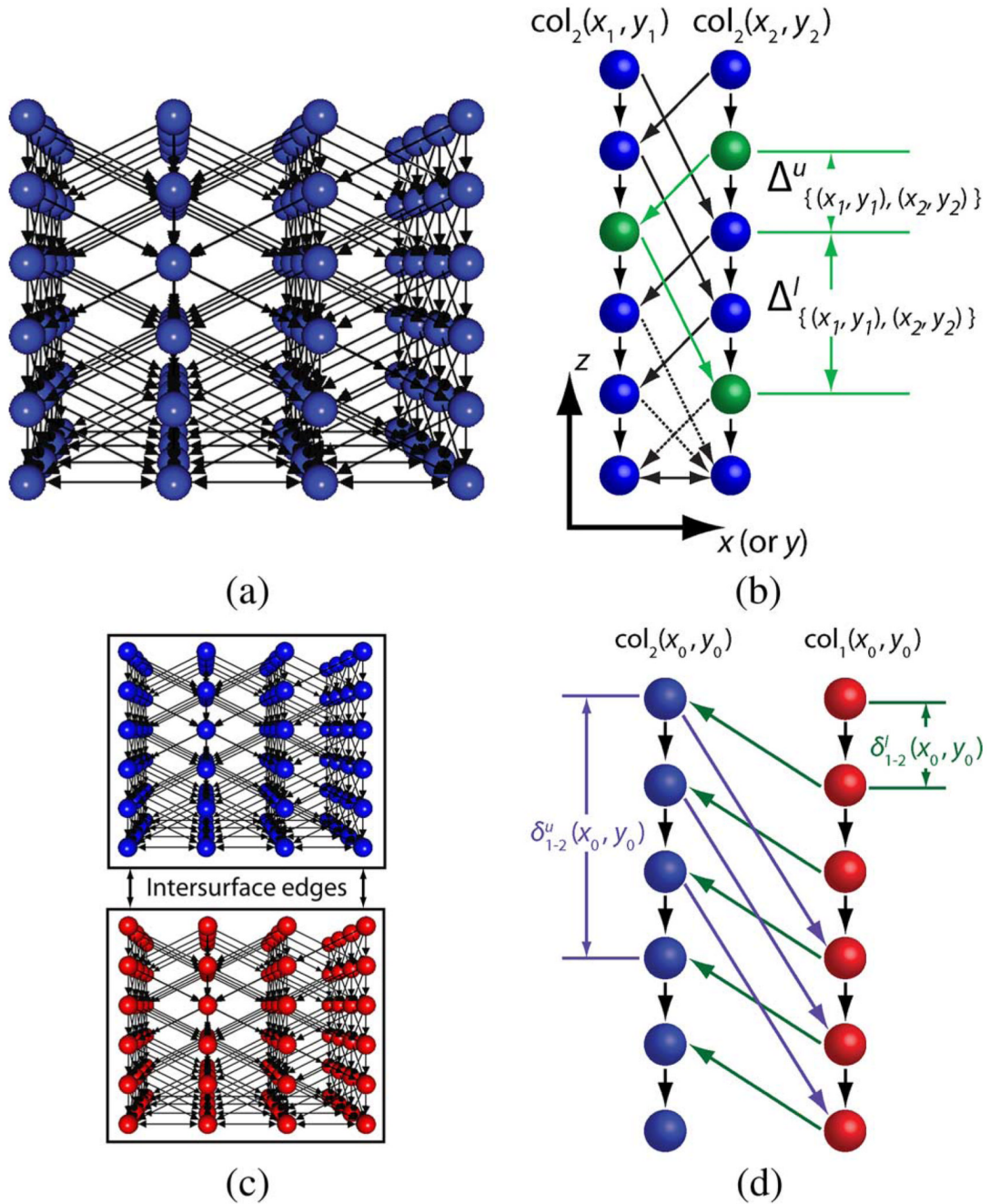


Fig. 3. Graph representation of feasibility constraints. (a), (b) One subgraph is created for each surface to be found with the added edges enforcing the surface smoothness constraints. In this example, the column denoted $col_2(x_1, y_1)$ is a neighbor to the column denoted $col_2(x_2, y_2)$. Both columns belong to the graph for surface 2. (c), (d) Intersurface edges are added between the surface subgraphs to enforce the surface interaction constraints. Here, $col_2(x_0, y_0)$ is the column at (x_0, y_0) belonging to the subgraph for surface 2 and $col_1(x_0, y_0)$ is the column at (x_0, y_0) belonging to the subgraph for surface 1.

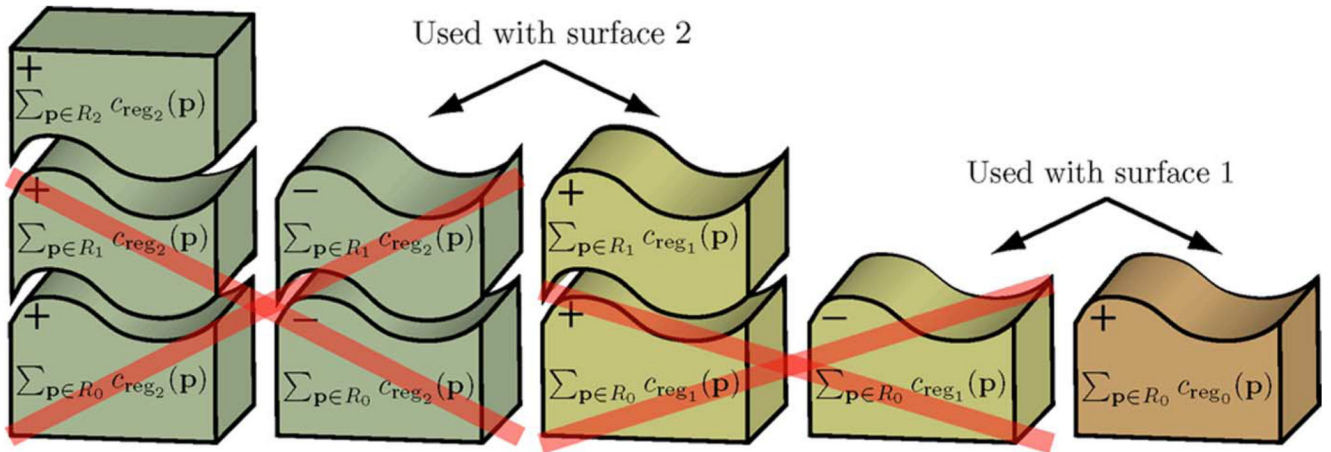


Fig. 4. Schematic showing how the assignment of in-region costs to vertices produces the desired overall cost.

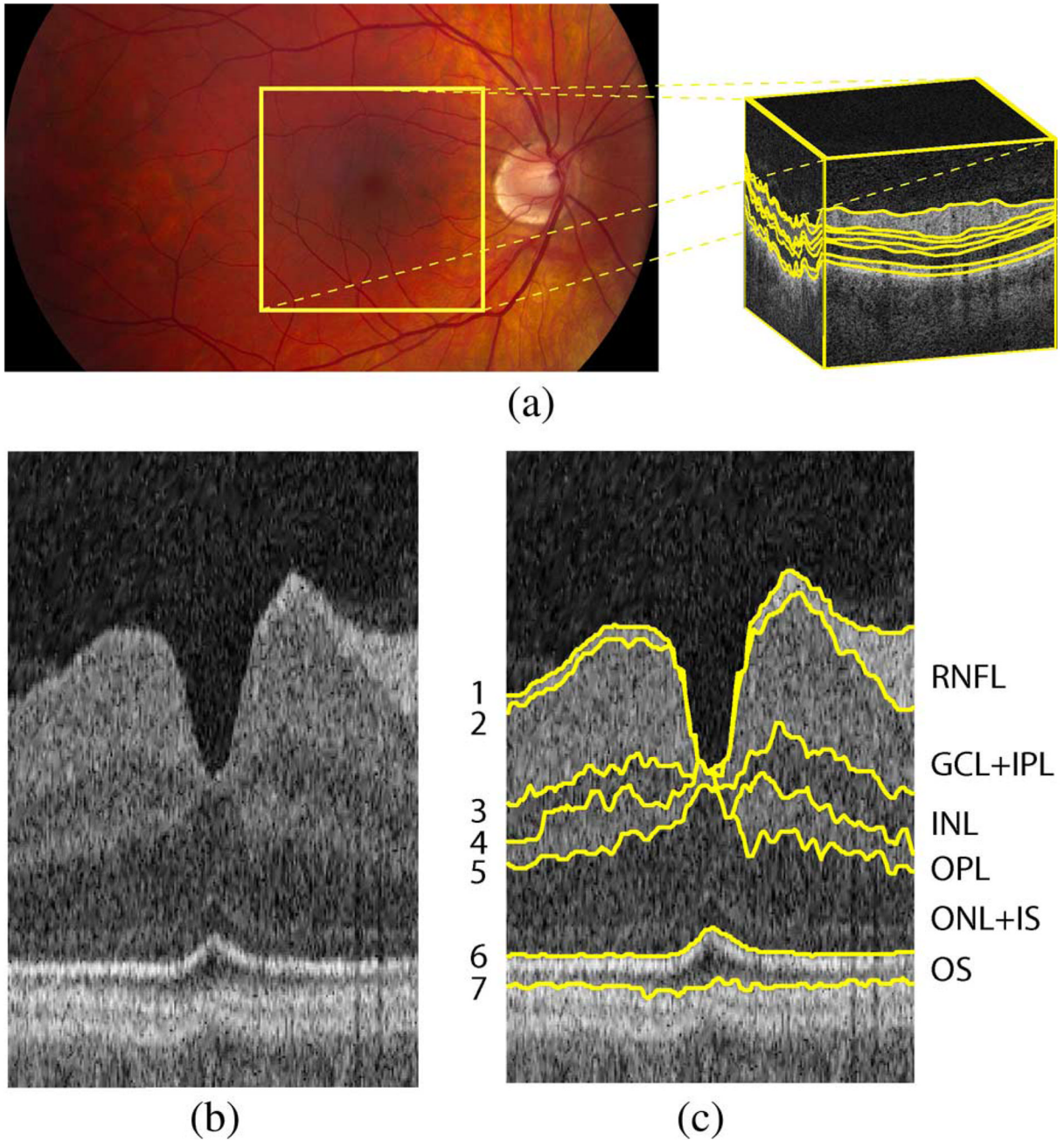


Fig. 5. Illustration of surfaces to be segmented on macular spectral-domain OCT images. (a) Fundus photograph with schematic OCT volume (size $6 \times 6 \times 2 \text{ mm}^3$) with surfaces indicated with yellow lines. (b) Example slice (flattened/truncated) from the center of an OCT volume. (c) Seven surfaces (and corresponding six layers) on example slice. (RNFL = retinal nerve fiber layer, GCL + IPL = ganglion cell layer and inner plexiform layer, INL = inner nuclear layer, OPL = outer plexiform layer, ONL + IS = outer nuclear layer and photoreceptor inner segments, and OS = photoreceptor outer segments). Note that the anatomical correspondence is our current presumption, but the precise correspondence of the outermost layers still remains a subject of discussion [3].

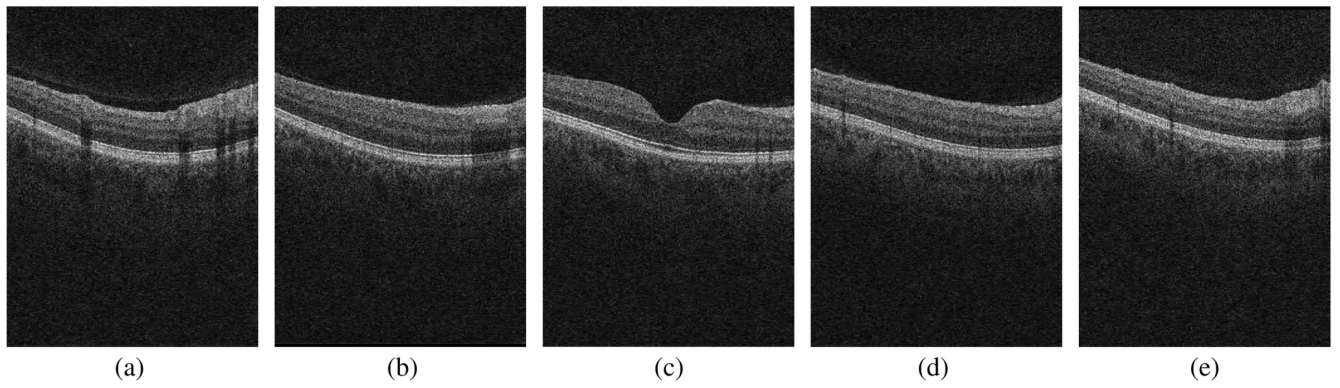


Fig. 6. Example slices from a macular spectral-domain OCT volume. (a) Slice 0. (b) Slice 55. (c) Slice 99. (d) Slice 154. (e) Slice 198.

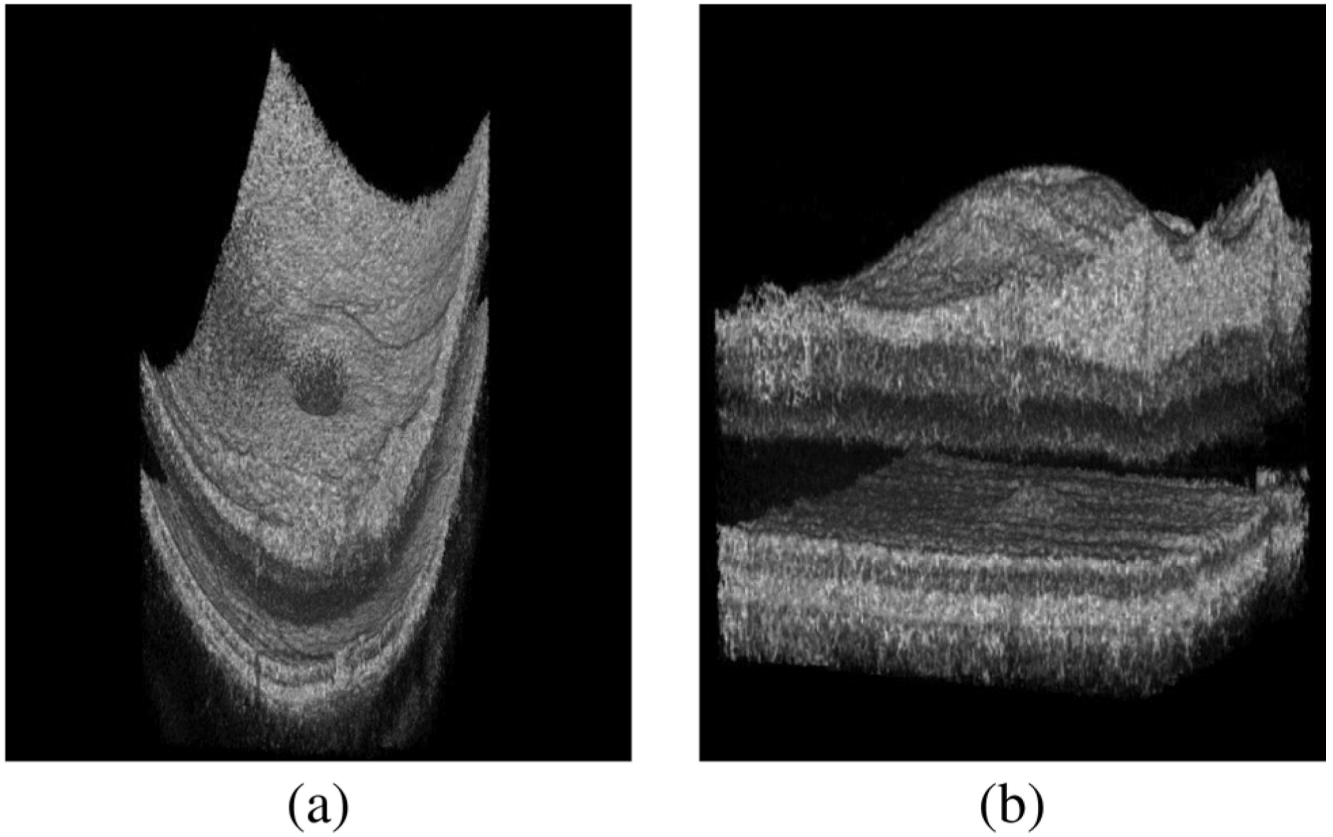


Fig. 7. Rendering of spectral OCT image before and after flattening. (a) Volume rendering before flattening. (b) Volume rendering after flattening.

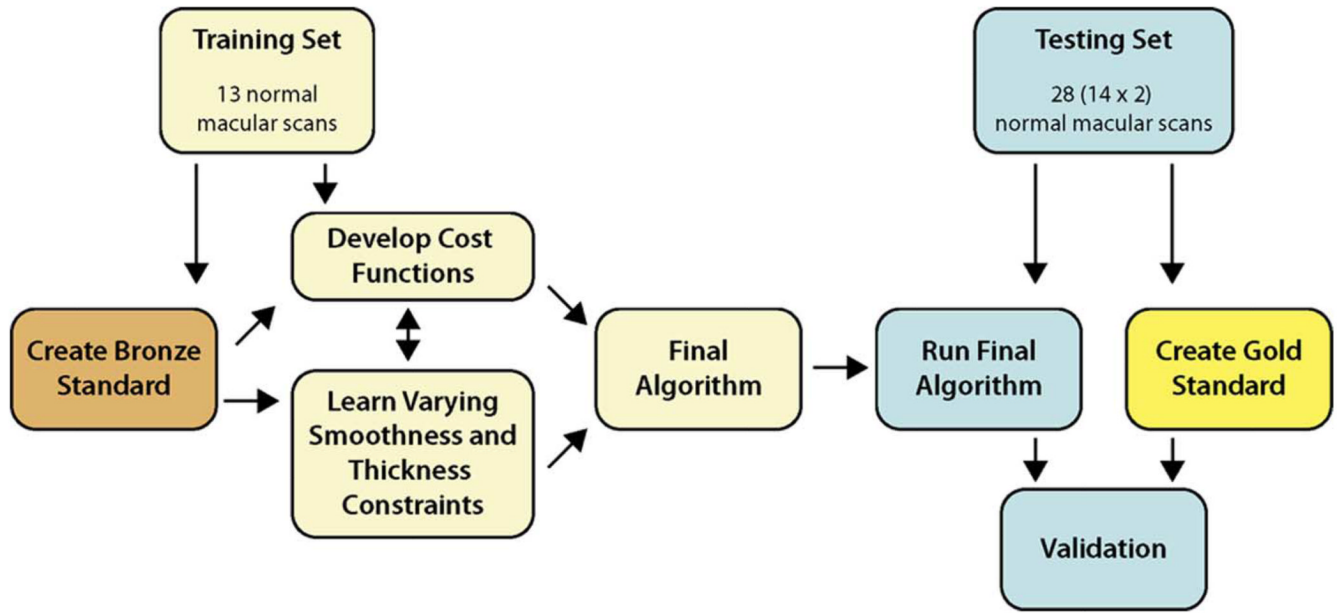


Fig. 8.
Spectral training/testing flowchart.

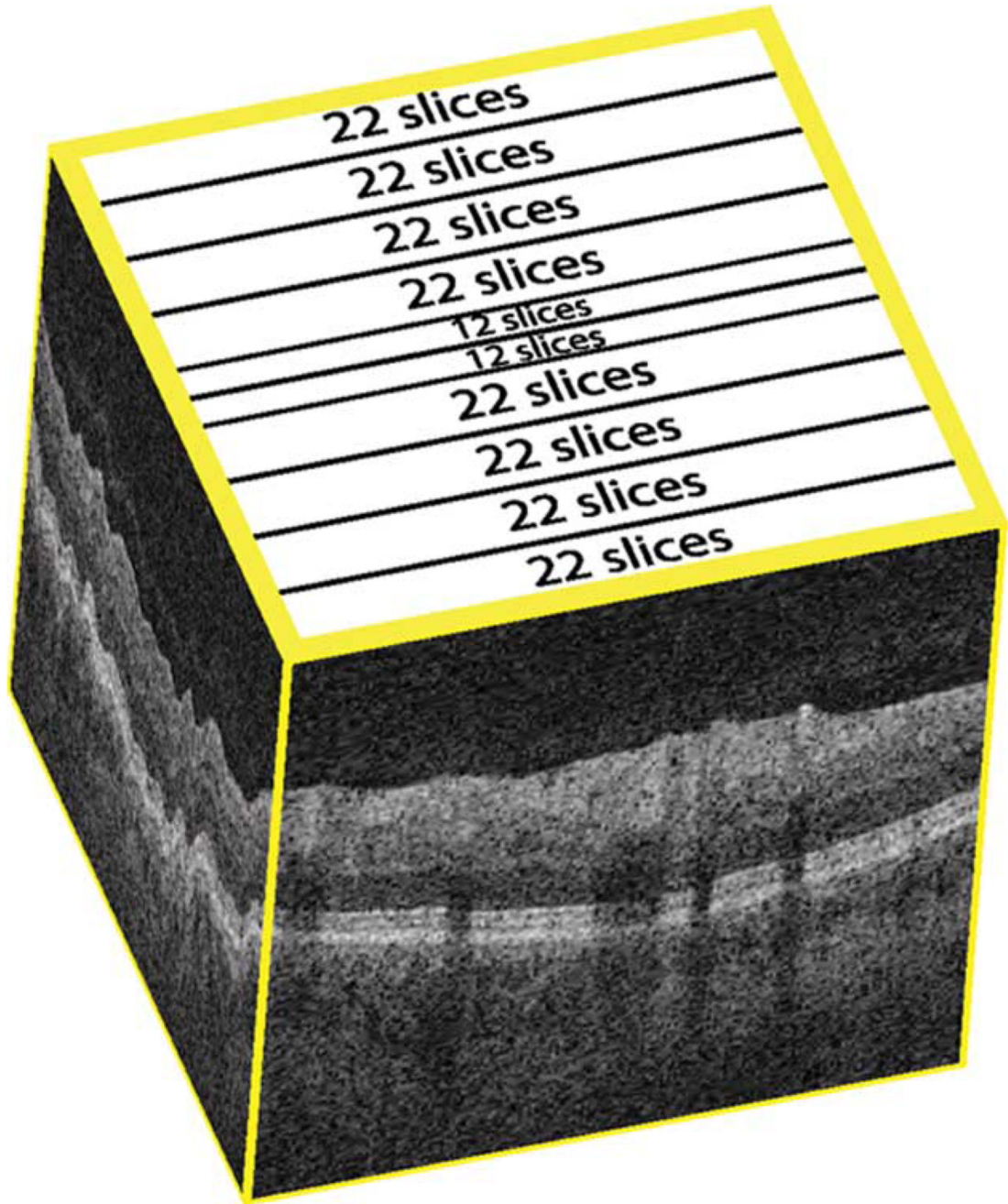


Fig. 9. Regions for randomly selecting slices for the macular spectral OCT independent standard (one slice was randomly chosen from each region).

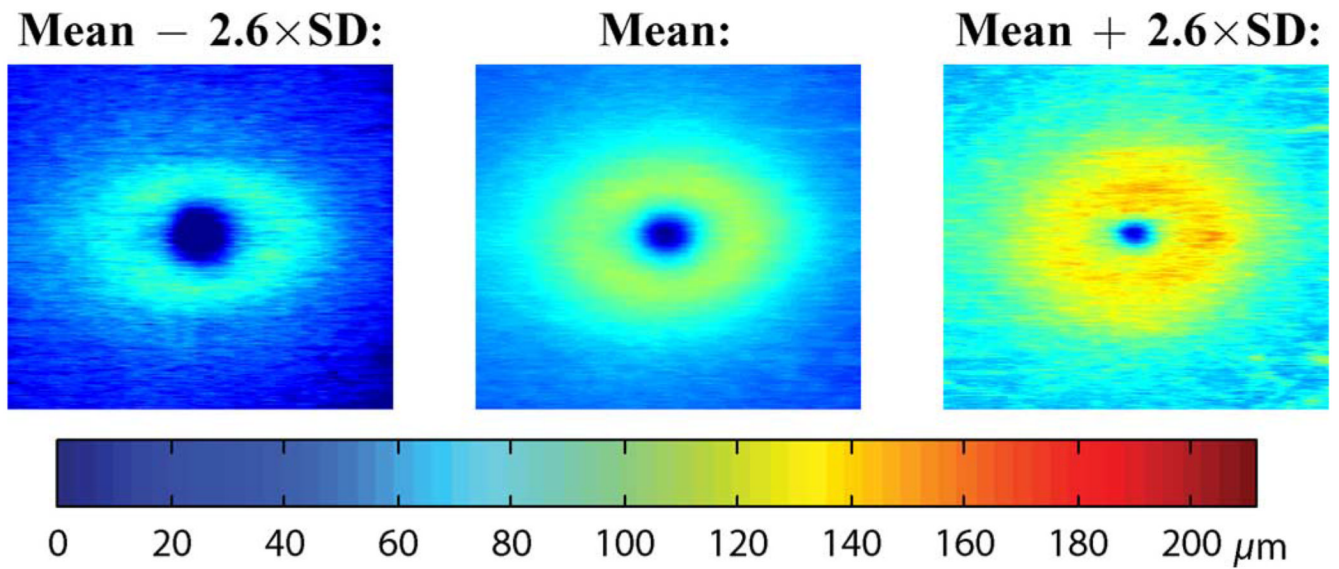


Fig. 10.

Visualization of learned thickness constraints for the layer bounded by surfaces 2 and 3. The minimum and maximum thickness constraints are illustrated in the first and third column, respectively, while the mean thicknesses are illustrated in the second column. Each image is oriented so that the temporal side is on the left and the nasal side is on the right.

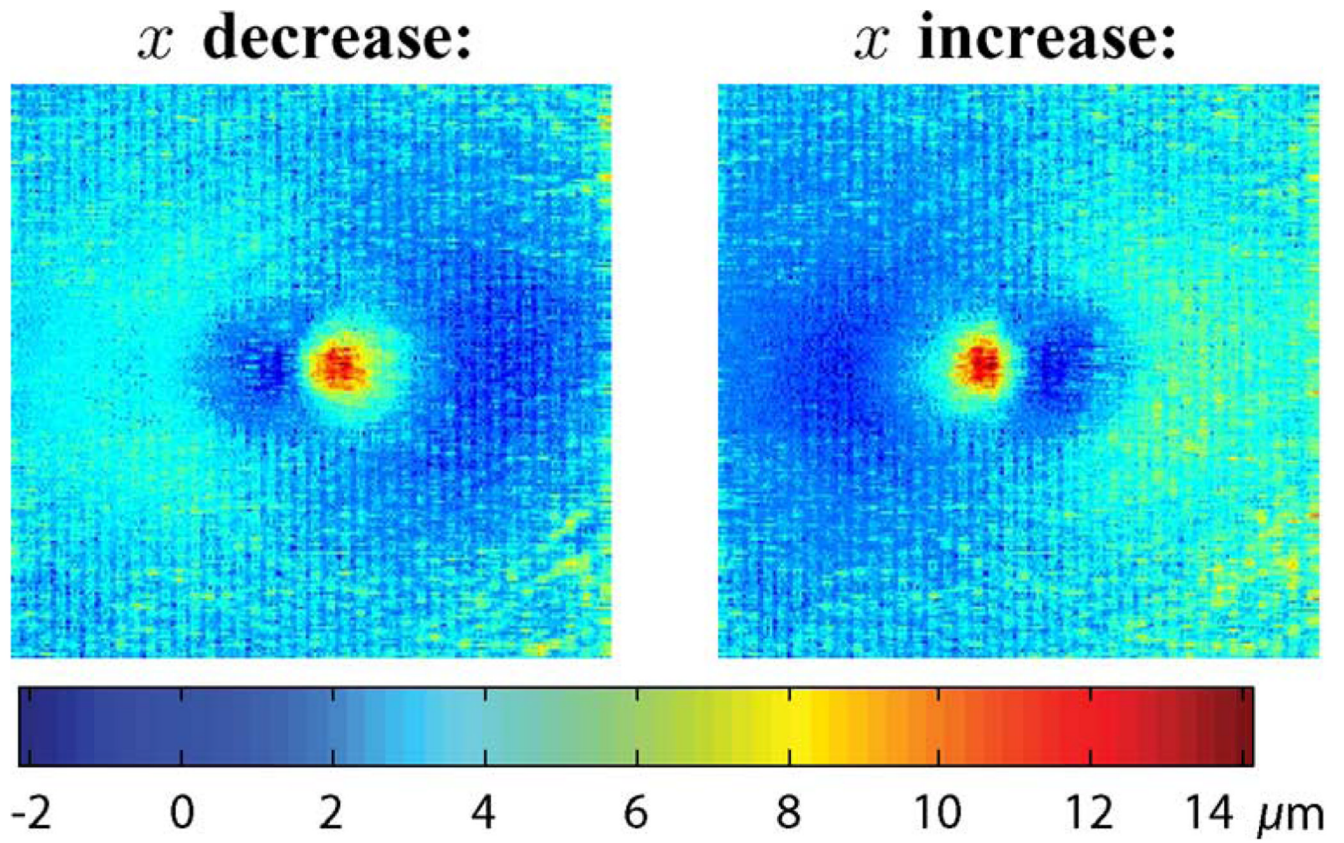


Fig. 11.
Learned smoothness constraints in x -direction for surface 2.

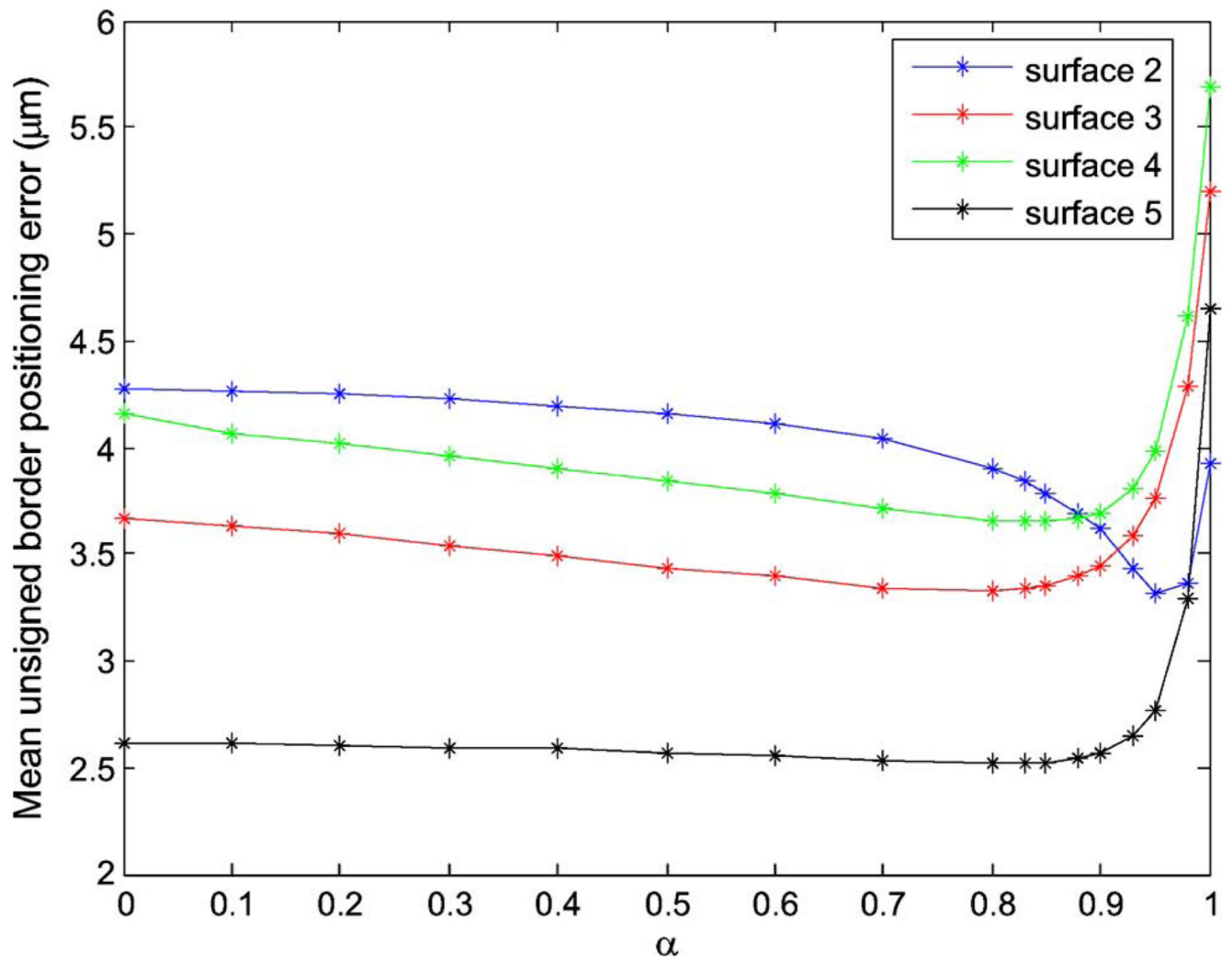


Fig. 12. Unsigned border positioning error results for different combinations of edge and regional information on training set.

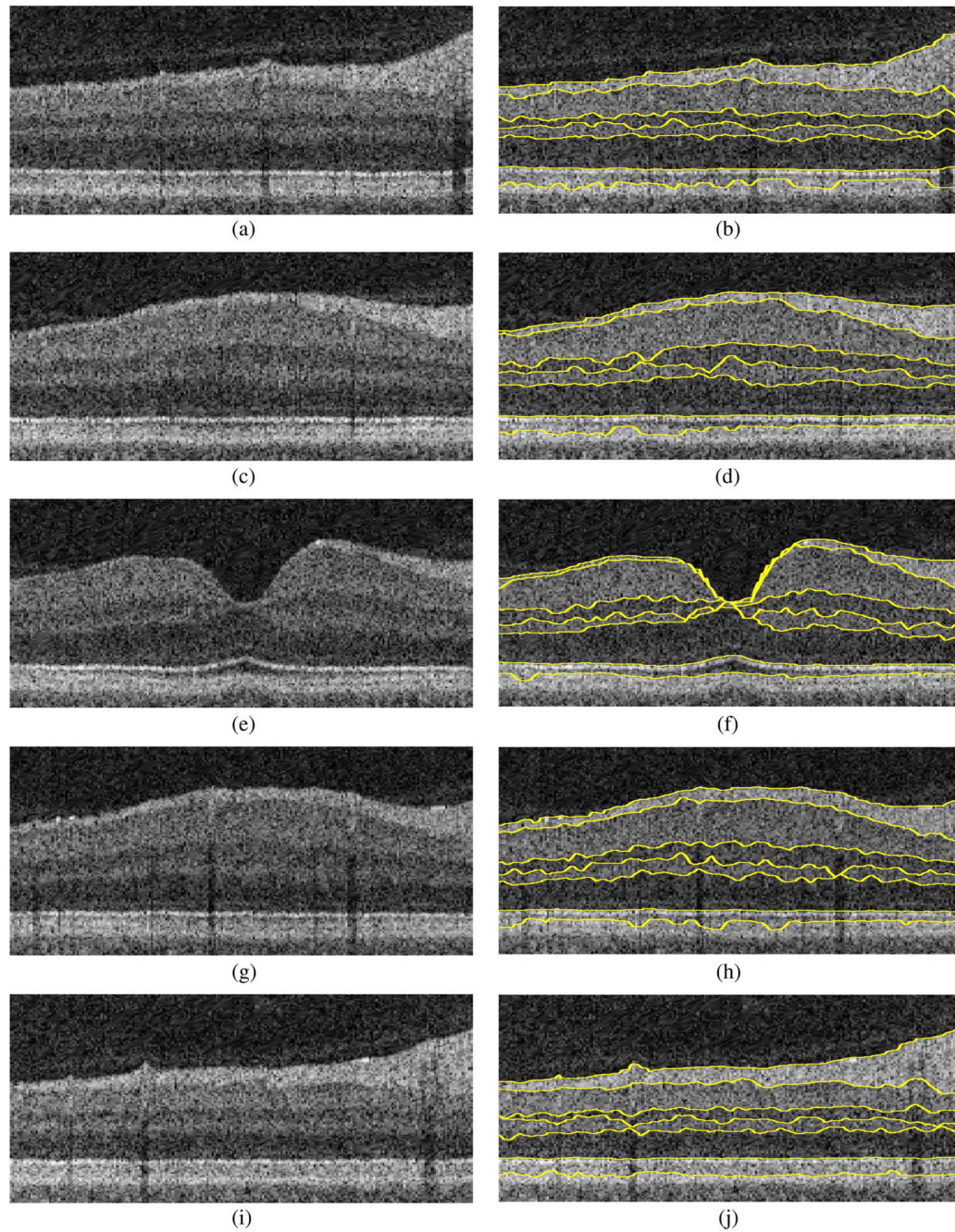


Fig. 13.

Example 7-surface 3-D segmentation results shown on five slices used for validation on one spectral-domain image in the test set (median case according to overall mean unsigned positioning error). (a) Slice 25. (b) Slice 25. (c) Slice 68. (d) Slice 68. (e) Slice 101. (f) Slice 101. (g) Slice 141. (h) Slice 141. (i) Slice 181. (j) Slice 181.

TABLE I

Bounding surfaces for estimating in-region cost parameters

Param.	Top est. surface	Bottom est. surface
Δb	$f_1(x,y)$	$f_1(x,y) + \delta_{1-2}^l(x,y)$
c_m	$f_1(x,y) + \delta_{1-2}^u(x,y)$	$f_1(x,y) + \delta_{1-2}^l(x,y) + \delta_{2-3}^l(x,y)$
Δd_2	$f_6(x,y) - \delta_{5-6}^l(x,y)$	$f_6(x,y)$

\dagger (x,y) columns for which the top estimation surface would go below the bottom estimation surface were excluded

TABLE IIBest values of α for each surface

Surface	α_i	$1 - \alpha_i$	$\frac{\alpha_i}{1 - \alpha_i}$
2	0.95	0.05	19.0
3	0.80	0.20	4.0
4	0.83	0.17	4.9
5	0.83	0.17	4.9

TABLE IIISummary of mean unsigned border positioning errors[†] for test set

Border	Obs. 1 vs. Obs. 2	Algo. vs. Avg. Obs.	Algo. vs. Obs. 1	Algo. vs. Obs. 2
1	3.48 ± 0.41	2.85 ± 0.32	3.70 ± 0.44	2.93 ± 0.34
2	5.49 ± 0.90	4.98 ± 1.24	6.28 ± 1.43	5.09 ± 1.25
3	6.68 ± 1.19	7.25 ± 1.10	8.30 ± 1.42	7.54 ± 1.18
4	7.06 ± 1.41	7.79 ± 0.74	9.59 ± 1.25	7.35 ± 0.54
5	6.16 ± 1.10	5.18 ± 0.82	6.50 ± 1.24	5.46 ± 0.75
6	3.99 ± 2.28	3.30 ± 1.60	4.28 ± 1.93	3.09 ± 2.03
7	7.11 ± 2.07	8.47 ± 2.29	8.48 ± 1.74	9.77 ± 2.76
overall	5.71 ± 1.98	5.69 ± 2.41	6.73 ± 2.48	5.89 ± 2.74

[†]Mean ± standard deviation in μm .

TABLE IVSummary of mean signed border positioning errors[†] for test set

Border	Obs. 1 vs. Obs. 2	Algo. vs. Avg. Obs.	Algo. vs. Obs. 1	Algo. vs. Obs. 2
1	0.20 ± 1.02	-1.52 ± 0.45	-1.62 ± 0.78	-1.42 ± 0.56
2	-0.34 ± 1.52	-2.04 ± 1.55	-1.87 ± 1.72	-2.21 ± 1.73
3	0.08 ± 2.12	5.02 ± 1.41	4.98 ± 1.78	5.06 ± 1.75
4	3.74 ± 2.47	-3.53 ± 2.18	-5.40 ± 3.07	-1.66 ± 1.76
5	1.88 ± 2.03	-1.19 ± 1.66	-2.14 ± 2.25	-0.25 ± 1.59
6	1.10 ± 2.57	-1.55 ± 1.78	-2.10 ± 2.09	-1.00 ± 2.29
7	4.10 ± 3.75	6.25 ± 3.37	4.20 ± 3.64	8.30 ± 4.07

[†]Mean ± standard deviation in μm .