Journal of Biomedical Optics

BiomedicalOptics.SPIEDigitalLibrary.org

Length-adaptive graph search for automatic segmentation of pathological features in optical coherence tomography images

Brenton Keller
David Cunefare
Dilraj S. Grewal
Tamer H. Mahmoud
Joseph A. Izatt
Sina Farsiu

SPIE.

Brenton Keller, David Cunefare, Dilraj S. Grewal, Tamer H. Mahmoud, Joseph A. Izatt, Sina Farsiu, "Length-adaptive graph search for automatic segmentation of pathological features in optical coherence tomography images," *J. Biomed. Opt.* **21**(7), 076015 (2016), doi: 10.1117/1.JBO.21.7.076015.

Length-adaptive graph search for automatic segmentation of pathological features in optical coherence tomography images

Brenton Keller,[a,*] David Cunefare,[a] Dilraj S. Grewal,[b] Tamer H. Mahmoud,[b] Joseph A. Izatt,[a,b] and Sina Farsiu[a,b]
[a]Duke University, Department of Biomedical Engineering, 101 Science Drive, Campus Box 90281, Durham, North Carolina 27708, United States
[b]Duke University, Department of Ophthalmology, Duke University Medical Center, Durham, North Carolina 27710, United States

Abstract. We introduce a metric in graph search and demonstrate its application for segmenting retinal optical coherence tomography (OCT) images of macular pathology. Our proposed "adjusted mean arc length" (AMAL) metric is an adaptation of the lowest mean arc length search technique for automated OCT segmentation. We compare this method to Dijkstra's shortest path algorithm, which we utilized previously in our popular graph theory and dynamic programming segmentation technique. As an illustrative example, we show that AMAL-based length-adaptive segmentation outperforms the shortest path in delineating the retina/vitreous boundary of patients with full-thickness macular holes when compared with expert manual grading. [DOI: 10.1117/1.JBO.21.7.076015]

Keywords: image segmentation; graph search; algorithm.

Paper 160132RR received Mar. 3, 2016; accepted for publication Jul. 11, 2016; published online Jul. 28, 2016.

1 Introduction

Optical coherence tomography (OCT) is widely utilized for many medical imaging applications, especially for diagnosing retinal diseases.[1] The thickness of individual layers in the retina is an important quantitative biomarker for diagnosis, management, and prognosis of ocular and neurodegenerative diseases. As manual segmentation of retinal layer boundaries visualized on OCT images is often too time-consuming and subjective for utilization in a clinical setting, several automated segmentation algorithms have been developed.[2–24] Despite progress and development of algorithms applicable to images for normal eyes and for eyes with limited pathology, segmentation of retinal layers with significant pathology is still a challenging problem. In addressing the problem of dealing with diseased retinas with severe deformation, graph-based segmentation frameworks have been among the most promising.[7,11–16,20]

Finding the shortest path through a graph, often implemented via Dijkstra's algorithm,[25] is the core engine behind many graph-based automated segmentation methods, including our popular graph theory and dynamic programming (GTDP) framework.[2,7,11] For healthy retinas [Fig. 1(a)][26] with layers that cross each A-scan only once the shortest path metric works well. However, the shortest path metric cannot faithfully represent certain classes of pathology in retinal images. An example is the case of a full-thickness macular hole, which is defined as a lesion with interruption of all layers of the neurosensory retina that develops in the fovea[27] [Fig. 1(b)]. In these pathologic eyes, the retina/vitreous boundary can cross an A-scan more than once, negating the applicability of Dijkstra's algorithm (Fig. 2).

The main contributions of this paper are to

1. demonstrate the shortcomings of the popular shortest path metric and the importance of utilizing length-adaptive metrics in design of graph-based layer segmentation algorithms,
2. introduce a new length-adaptive graph search metric, applicable to a wide variety of layer segmentation problems in medical images,
3. introduce an algorithm for implementing this metric in the graph-search framework, and
4. show, as a practical example, how this metric can be utilized in an automatic algorithm for segmenting the retina/vitreous boundary in OCT images of eyes with full-thickness macular holes.

2 Methods

When applying a general graph search technique to image boundary segmentation, each pixel in the image is converted to a node in the graph. Nodes in our graph are connected to their surrounding eight neighbors via a positive weighted edge. Edge weights between neighbors are assigned based on a feature between the two nodes (pixels) the edge connects. This feature is then converted into a weight via a mapping function. The cost of traversing an edge expresses how likely this edge corresponds to a boundary.[28] Examples of features include, but are not limited to, pixel intensity and the gradient between the neighboring pixels. Once the graph of nodes and edges is constructed, boundaries (layers) can be found by searching for paths through the graph. In order to find a path through the graph, search algorithms seek paths that minimize a cost metric (e.g., sum of the edge weights in a path also known as the total path length).

*Address all correspondence to: Brenton Keller, E-mail: brenton.keller@duke.edu

1083-3668/2016/$25.00 © 2016 SPIE

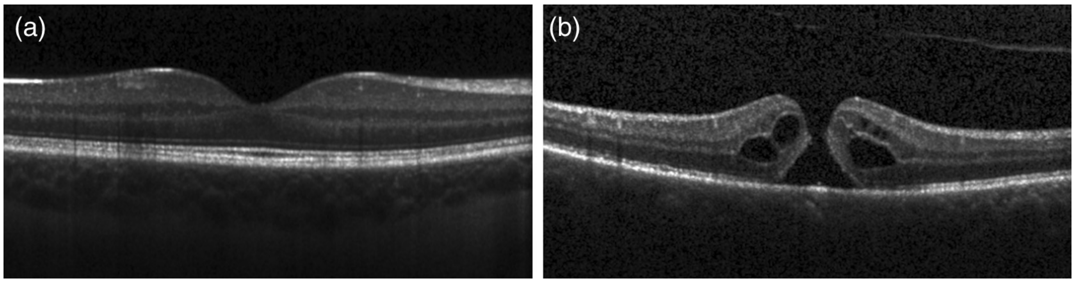

Fig. 1 (a) Comparison of a macular OCT image centered at the fovea in a healthy retina; (b) to a retina with a full-thickness macular hole.

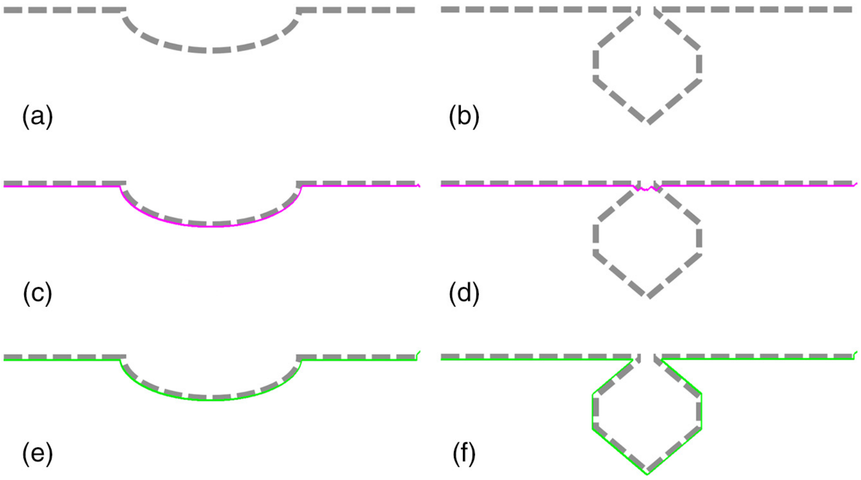

Fig. 2 Comparison of the shortest path and length-adaptive graph search. The function in (a), which may represent a normal retinal layer boundary, crosses each A-scan only once. Thus, the shortest path perfectly matches the contour of the target object, as this path goes through the minimum number of vertices, which have the lowest weights (b). However, the nonfunction in (c), which may represent retina vitreous boundary in a full-thickness macular hole case, crosses some A-scans more than once. In this case, the shortest path metric does not match the contour of the object (d), because the alternative path going through a fewer number of vertices, even though they may have higher weights, has a lower overall path length. On the other hand, due to path length normalization, the length-adaptive search is less sensitive to the geometric path length, and correctly segments both features (e and f).

2.1 *Length-Adaptive Graph Search*

In the shortest path problem, given a graph with a set of vertices (nodes) and positive weighted edges, the goal is to find a path that connects a start vertex to an end vertex by minimizing the total path length

$$\text{Path Length} = \sum_{i=1}^{N-1} w_i, \tag{1}$$

where N is the total number of nodes and w_i is the i'th edge weight in the path. Following the principle of mean arc length[29] and normalized cuts,[30] in our proposed length-adaptive method the goal is to find a path that minimizes the following metric, which we call the AMAL,

$$\text{AMAL} = \frac{1}{N-1}\left(\sum_{i=1}^{N-1} w_i\right)^x, \tag{2}$$

where x is a constant. In the case where x is one, Eq. (2) simplifies to the mean arc length of the path.

To demonstrate the effect of varying the value of exponent in Eq. (2), we have segmented the same image with four different values of x, shown in Fig. 3. Utilization of a graph search metric which is completely independent of path length [i.e., $x = 1$, Fig. 3(a)] may result in erroneously long paths. Increasing the value of x causes the length-adaptive algorithm to become more sensitive to the total distance of the path. This results in a shorter overall path length of the segmentation, which, when segmenting the retina/vitreous boundary of a full-thickness macular hole, reduces the accuracy. Searching for a path with an AMAL calculated with an exponent greater than 1 in Eq. (2) provides a balance between the total distance of the path and the mean arc length of the path.

2.2 *Implementation of the Adjusted Mean Arc Length Metric*

Unfortunately, unlike the classic shortest path-based segmentation, which can be implemented efficiently using Dijkstra's algorithm,[2] implementation of length-adaptive graph search using the AMAL metric is more challenging. Wimer et al.[29] developed two methods for finding the path with the lowest mean arc length in weighted graphs: an iterative approach

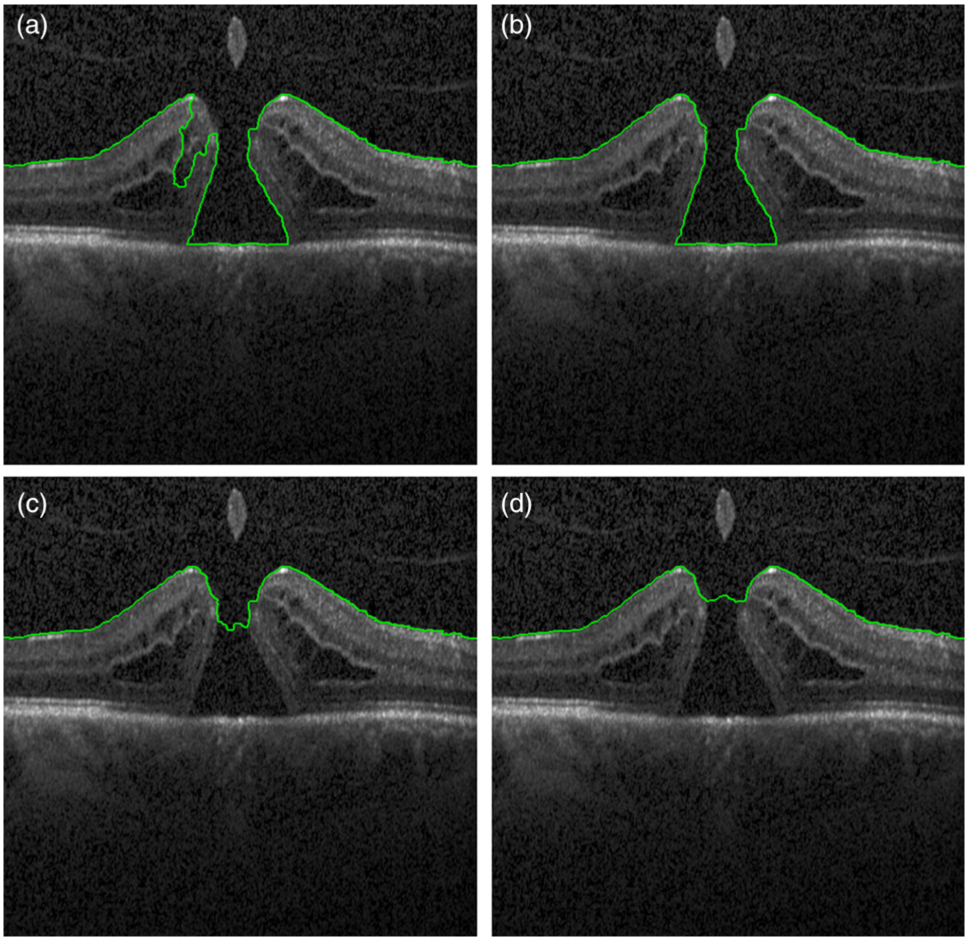

Fig. 3 Illustration of the effect varying x in Eq. (2) has on retina/vitreous boundary segmentation in a patient with a macular hole. (a) Segmentation with $x = 1.0$ (i.e., the mean arc length of the path).[29] (b) Segmentation with $x = 1.02$ (value used in validation). (c) Segmentation with $x = 1.25$. (d) Segmentation with $x = 1.75$.

that finds an approximation to the solution and an exact solution. These two methods could potentially be modified to find the path with the lowest AMAL, but they both assume the graph is acyclic, where the search space prevents visiting a vertex more than once. Because we want to segment arbitrary shapes (Fig. 4), edges between neighboring nodes must be bidirectional, which creates cycles in the search space. Without cycles, the segmentation would only be able to search in one direction along the horizontal axis of the image, either left-to-right or right-to-left. Thus, in order to segment arbitrary shapes, our overall graph search space, but not the final segmentation, must contain cycles.

In this paper, to find an estimate of the path with the lowest AMAL in a graph, we created a priority queue sorted by the AMAL value of visited nodes. For each node in the graph, we recorded the number of neighbor nodes, the edge weights to the neighbor nodes, the mean arc length of the path up to that node, the previous node, the number of previous nodes in the path, and the AMAL of the path up to that node. The priority queue was initialized with the start node, which had an AMAL of zero, and we set the AMAL value of all other nodes to infinity. We then modified the iterative Dijkstra's method outlined by Chen et al.[31] to greedily search for the path with the lowest AMAL. At every iteration, we removed

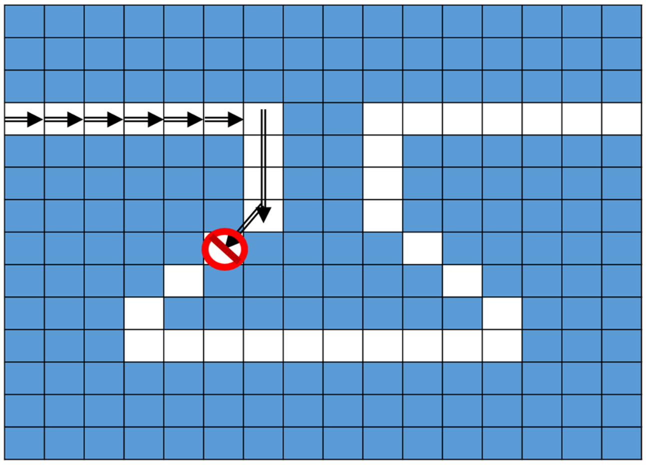

Fig. 4 Illustration of a graph that prevents cycles in the search space. Black arrows represent attempted segmentation and the white squares represent the image to be segmented. Because the graph search space does not contain cycles, there cannot be bidirectional edges between neighboring nodes. Therefore, correct segmentation of nonfunctional shapes becomes impossible (red circle). To prevent this problem, we allow our graph search space to contain cycles.

```
Algorithm Length-Adaptive Graph Search
 1: start.AMAL ← 0
 2: queue.push(start)
 3: while $queue$ is not empty do
 4:     curNode ← queue.pop()
 5:     for all neighbors n of $curNode$ do
 6:         newAMAL ← calc_AMAL()                        ▷ calc_AMAL() uses Equation 3
 7:         if ($newAMAL < n.AMAL$) then
 8:             update_neighbor()                        ▷ Update MAL, AMAL, etc. and push on queue
 9:         else if ($newAMAL = n.AMAL$ and
10:             $n.prevVertex = curNode$ and
11:             $n.prevNodes \neq curNode.prevNodes + 1$) then
12:             update_neighbor()                        ▷ Update MAL, AMAL, etc. and push on queue
13:         else if ($newAMAL > n.AMAL$ and
14:             $n.prevVertex == curNode$ and
15:             $n \neq EndNode$) then                   ▷ Node used for automatic endpoint initialization
16:             update_neighbor()                        ▷ Update MAL, AMAL, etc. and push on queue
17:         else if ($newAMAL > n.AMAL$ and
18:             $n.prevVertex == curNode$ and
19:             $n == EndNode$) then
20:             find_low_AMAL()                          ▷ Find node N which gives $EndNode$ lowest AMAL
21:             update_neighbor()                        ▷ Update MAL, AMAL, previous node etc.
22:         end if
23:     end for
24: end while
```

```
Function update_neighbor
1: n.AMAL ← newAMAL
2: n.MAL ← newMAL
3: n.prevVertex ← curNode
4: n.prevNodes ← curNode.prevNodes + 1
5: queue.push(n)
```

```
Function find_low_AMAL
 1: lowestAMAL ← ∞
 2: lowNode ← 0
 3: for all neighbors n of EndNode do
 4:     newAMAL ← calc_AMAL()
 5:     if ($newAMAL < lowestAMAL$) then
 6:         lowestAMAL ← newAMAL
 7:         lowNode ← n
 8:     end if
 9: end for
10: return lowNode
```

Fig. 5 Pseudocode for the length-adaptive graph search algorithm (left). Functions which are called in the algorithm (update_neighbor and find_low_AMAL) are described in pseudocode on the right.

the node with the lowest adjusted mean arc length from the top of the queue and made it the current node (Fig. 5 line 4). We generated a new AMAL for each neighbor of the current node (Fig. 5 line 6) using the equation

$$\text{AMAL}_{\text{new}} = \frac{(N_i \bar{w}_i + w_{ij})^x}{N_i + 1}, \tag{3}$$

where $\bar{w}_i$ is the mean arc length of the path up to the current node, w_{ij} is the weight between the current node and the neighbor node, N_i is the current node's number of previous nodes, and x is the exponential factor. We then checked each neighbor node to see if it satisfied one of four conditions. If the new AMAL was lower than the old AMAL, the mean arc length, AMAL, previous node, and number of previous nodes were all updated and the neighbor node was placed in the queue (Fig. 5 lines 7 to 8). If the new path to the neighbor superseded its stored path, we updated the mean arc length, AMAL, previous node, number of previous nodes and placed the neighbor node in the queue (Fig. 5 lines 9 to 16). If the new path to the end node (used for automatic endpoint initialization, see below) superseded its stored path, we looked at all the nodes connected to the end node, found the one that produced the lowest AMAL for the end node, and updated the end node accordingly (Fig. 5 lines 17 to 21). The algorithm terminated when there were no more nodes left in the queue. To determine the path with the lowest AMAL, we started at the end node and traced backward through previous nodes until the start node was reached.

2.3 *Representative Application in Retinal Optical Coherence Tomography Layer Segmentation*

The length-adaptive graph search algorithm described is general and applicable for segmenting virtually any layered feature. Indeed, optimal utilization for specific applications requires customization. One such application of this new graph search algorithm is segmentation of the retina/vitreous boundary in OCT images of patients with full-thickness macular holes, which is detailed in the following.

We began the segmentation process by denoising the 8-bit grayscale OCT images using a previously described sparsity-based denoising technique.[32] Next, we calculated three gradient images, two horizontal gradient images, and one vertical gradient image by convolving the image with $[1,1,1,1,1,0,-1,-1,-1,-1,-1]$, $[-1,-1,-1,-1,-1,0,1,1,1,1,1]$, and $[1;1;1;1;1;0;-1;-1;-1;-1;-1]$ (MATLAB notation) filters. We then linearly normalized the pixel values in these three gradient images to be between 0 and 1. We performed nonmaximum gradient suppression along each column for the vertical gradient image and each row for the horizontal gradients images using the following equation:

$$G_i = \begin{cases} G_i & \text{if } \operatorname{sgn}(G_{i+1} - G_i) - \operatorname{sgn}(G_i - G_{i-1}) < 0, \\ -1 & \text{otherwise}, \end{cases} \tag{4}$$

where G_i is the gradient at index i in the column/row. Finally, we computed the edge weights between adjacent nodes in the graph as[2]

$$w_{ij} = 2 - (G_i + G_j) + 1e^{-5}, \tag{5}$$

where w_{ij} is the edge weight between nodes i and j, G_i is the gradient at node i, and G_j is the gradient at node j. If node j was above node i we used the $[1,1,1,1,1,0,-1,-1,-1,-1,-1]$ gradient, if node j had the same y-coordinate as node i we used the $[1;1;1;1;1;0;-1;-1;-1;-1;-1]$ gradient, and if node j was below node i we used the $[-1,-1,-1,-1,-1,0,1,1,1,1,1]$ gradient.

In order to not erroneously segment the bright hyperreflective retinal pigment epithelium (RPE) layer in place of our target retina/vitreous boundary, we performed an additional prepossessing step before running the length-adaptive graph search. We began by creating a binary image to isolate the nerve fiber layer (NFL) and RPE using the technique described by

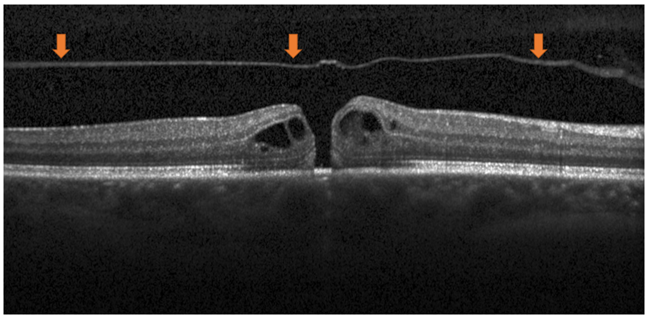

Fig. 6 Image of full-thickness macular hole with posterior hyaloid, denoted by orange arrows, in the vitreous.

Chiu et al.[7] Next, we divided each binary image horizontally into two equal halves and processed them separately. We obtained pilot estimates of the RPE and NFL layers by using Dijkstra's algorithm to segment each halved section twice. We performed a third segmentation if over 80% of the columns in the binary image contained more than two black to white transitions. This third segmentation was put in place to account for posterior hyaloids (Fig. 6), which in a binary image can appear like another retinal layer. We removed segmented nodes from the graph after finding each layer, and the remaining graph was segmented again. The NFL was chosen to be highest layer in the image if two layers were detected, or the middle layer if three layers were detected.

After determining the estimate for the NFL, we increased edge weights in the graph that were above and below the NFL estimate proportional to their vertical distance from the NFL estimate using

$$w_{x,y} = w_{x,y} + \mathrm{abs}(y - \mathrm{NFL}_x) \times 0.05, \tag{6}$$

where $w_{x,y}$ represents the non-zero edge weights for all vertices connected to the node (x, y) and NFL_x represents the y-location of the NFL estimate in column x. We only increased the edge weights of nodes that were in the left and right third of the image, as we assumed the macular hole was in the center third of the image.

We added two more nodes to the graph for automatic endpoint initialization, a start node and an end node. The start node was defined to be neighbors with every pixel in the first column of the image, and every pixel in the last column of the image was neighbors with the end node. The weights between the start/end node and its neighbors were set to the minimum weight value of $1e^{-5}$. A flow chart of the segmentation process is shown in Fig. 7.

Based on testing performed with our training data set, we set the exponential parameter x of our AMAL metric in Eq. (2) to 1.02. For the specific application of retinal layer segmentation, we added two extra conditions in the implementation of the length-adaptive graph search algorithm to decide whether or not to add a neighbor node to the priority queue. If the neighbor node's new AMAL was lower than its stored AMAL, we also checked to see if one of the following constraints was violated before updating corresponding parameters and placing the neighbor node in the queue:

- the neighbor node was already in the path, or
- the neighbor node was closer than the lateral or axial pixel resolution to a set of nodes already in the path.

We evoked the first constraint to prevent cycles in the final result [Fig. 8(a)]. As noted, it is important that the graph search space is able to contain cycles for representing arbitrary shapes, but no cycles are desired in the final segmentation. The second constraint was evoked to prevent adding unnecessary nodes to the segmentation, e.g., to prevent the segmentation from folding back on itself [Fig. 8(b)]. If neither of these constraints were violated, the appropriate properties of the node were updated and this node was placed in the queue. Otherwise, the node was not added to the queue.

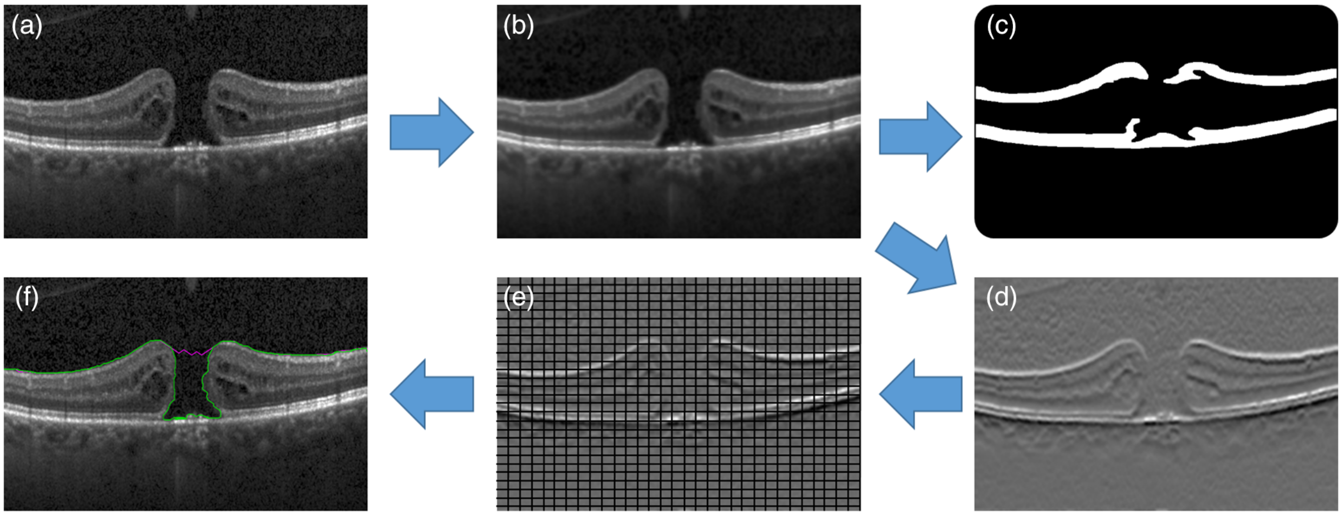

Fig. 7 Graphical representation of the segmentation process. The original image is shown in (a), which is denoised to create image (b). We then obtain a rough estimate of the location of the nerve fiber layer (top white band) and retinal pigment epithelium (bottom white band) using the binary image (c) as described in Sec. 2.3. From (b) we also create the gradient images (d). Next, we create the graph (e) from the gradient images and modify the edge weights based on the pilot estimation of the nerve fiber layer location from (c). Finally, we perform the length-adaptive graph search (green) and shortest path search (magenta) to segment the retina/vitreous boundary (f).

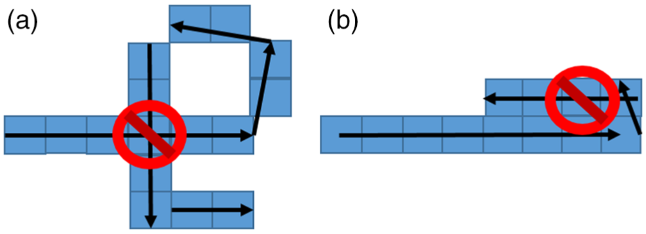

Fig. 8 Illustration of when neighbor nodes were not added to the priority queue in the length-adaptive graph search algorithm. (a) Node circled in red that would not be added to the priority queue because it causes a cycle. (b) Node that would not be added to the priority queue because it is too close to other nodes in the segmentation.

2.4 *Dataset*

This study was approved by the Duke University Health System Institutional Review Board in accordance with Health Insurance Portability and Accountability Act regulations and the standards of the 1964 Declaration of Helsinki. Ten different patients with full-thickness macular holes were imaged with a Heidelberg Spectralis (Heidelberg Engineering, Heidelberg, Germany) OCT system using a radial scan pattern centered on the fovea. Twenty-four images were collected from each patient. Nine of the 10 sets of images were 512 × 496 (width and height) pixels, with pixel resolution 11.06 to 11.65 μm laterally and 3.87 μm axially. The other set of images was 1024 × 496 with a lateral pixel resolution of 5.73 μm and an axial pixel resolution of 3.87 μm. Five images from each patient were randomly selected to segment. These images were manually segmented by a clinical retina specialist and compared with the previous shortest path segmentation and our length-adaptive segmentation. We derived all algorithm parameters from a separate training set of images from subjects with full-thickness macular holes. None of the training images were used in validation. For a fair comparison, all preprocessing steps were performed for both shortest path and length-adaptive techniques.

3 Results

A qualitative comparison of the two segmentation methods is shown in Fig. 9. The top row shows the original images, from four separate patients with full-thickness macular holes, and the bottom row shows the segmentation for the length-adaptive (green) and shortest path (magenta) segmentations. These results show that the length-adaptive method more closely follows the contours of the macular holes, while the shortest path algorithm misses them entirely.

For each B-Scan, we calculated the mean distance between each pixel location on the manual segmentation and the closest point on the automatic segmentations. Then, we computed the average, standard deviation, and median of these mean pixel distances for all 50 B-Scans as a measure of error. The mean, standard deviation, and median of the error for the length-adaptive and shortest path algorithms are shown in Table 1.

The mean, standard deviation, and median of the time needed to segment the macular hole images with the shortest path method were 0.043, 0.013, and 0.039 s, respectively. The mean, standard deviation, and median of the time needed to segment the images with the length-adaptive method were 4.53, 10.47, and 2.65 s, respectively. Timing results were obtained on a 64-bit desktop computer with an Intel Core i7-4930K 3.4 GHz CPU. Our implementation of the length-adaptive algorithm was not optimized to run quickly and we expect the execution time could be reduced by applying parallel processing

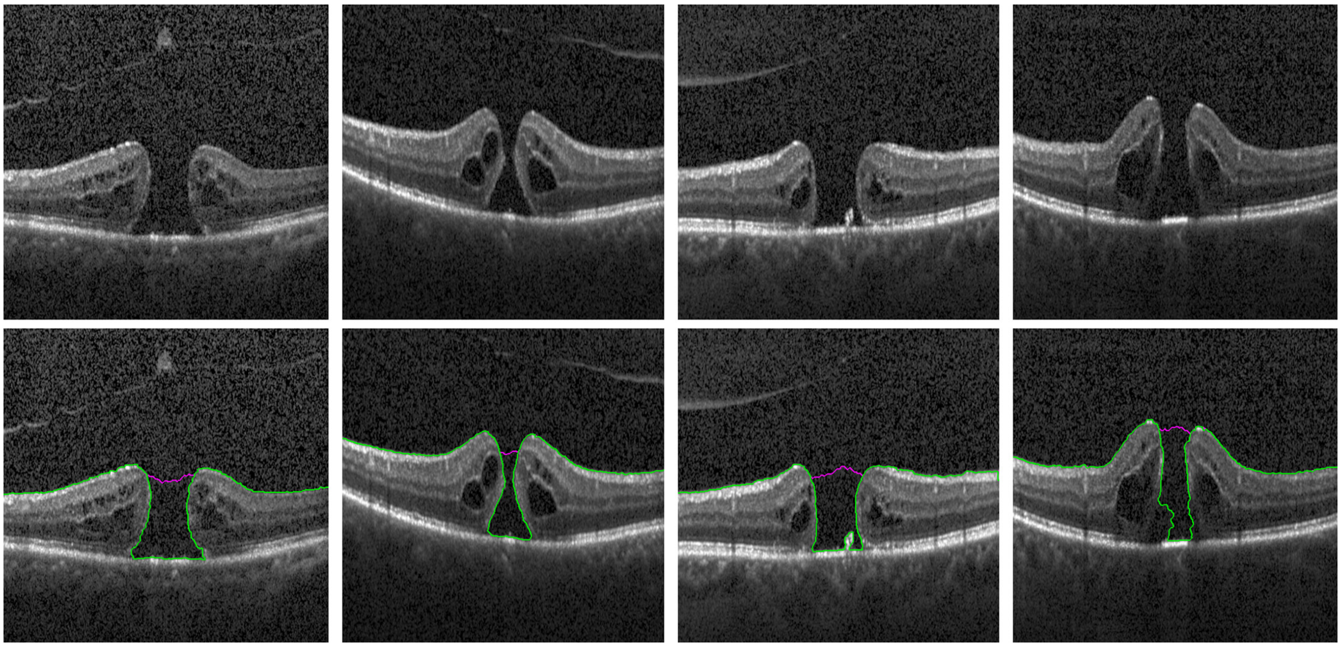

Fig. 9 Segmentation of the retina/vitreous boundary of four patients with full-thickness macular holes using shortest path length and length-adaptive algorithms. The top row are the original images and the bottom row are the segmented images. Magenta denotes the classic shortest path segmentation and green denotes the proposed length-adaptive segmentation. Where the magenta is not visible the two methods segmented the same line. The far right image shows an example where the length-adaptive algorithm failed to successfully segment the retina/vitreous boundary in the lower-left part of the macular hole.

Table 1 Error comparison of shortest path and length-adaptive segmentation to expert manual grader for 50 images.

Method	Mean (pixels)	Standard Deviation (pixels)	Median (pixels)
Shortest path	39.98	10.85	42.31
Length-adaptive	3.40	5.74	1.31

techniques. We could also use segmentation results from images in the same volume to limit the search region to reduce the processing time and improve segmentation accuracy. A similar method was used by Xu et al.[33]

Further, as an illustrative example, to qualitatively demonstrate that the length-adaptive method can also be used on non-pathological images, we segmented the retina/vitreous boundary with $x = 1.02$ in a normal subject from the dataset distributed by Srinivasan et al.[26] This image was not part of our training dataset. Results of this segmentation are shown in Fig. 10 with the original image on the left and the segmented image on the right. As expected, the results of these two techniques for segmenting such simple features are virtually identical.

As a demonstrative example of how the length-adaptive graph search can be used to segment other layers in retinal OCT images, we have segmented the inner aspect of the RPE in two separate patients with age-related macular degeneration (AMD) (Fig. 11). In order to segment these images, we obtained an estimate of the RPE using the same method described earlier. We then increased the weights between edges above the RPE estimate to prevent the segmentation of the NFL. When creating the gradient images, we used $[1,1,1,0,-1,-1,-1]$, $[-1,-1,-1,0,1,1,1]$, and $[1;1;1;0;-1;-1;-1]$ filters. Finally, we changed the exponent from Eq. (2) to be 1.4. Optimization of an AMAL-based drusen segmentation algorithm and careful validation is outside the scope of this preliminary paper and will be addressed in our future publications.

4 Discussion

We have demonstrated the utility of a new metric in graph search called AMAL. Applications of this metric are general and it can be potentially applied to virtually any layer (or closed contour feature that can be converted to a layer[34]) segmentation problem including segmentation of intravascular,[35] esophageal wall,[36] epithelium,[37] or epidermal[38] OCT images. In this first report, we demonstrate its application for segmentation of the retina/

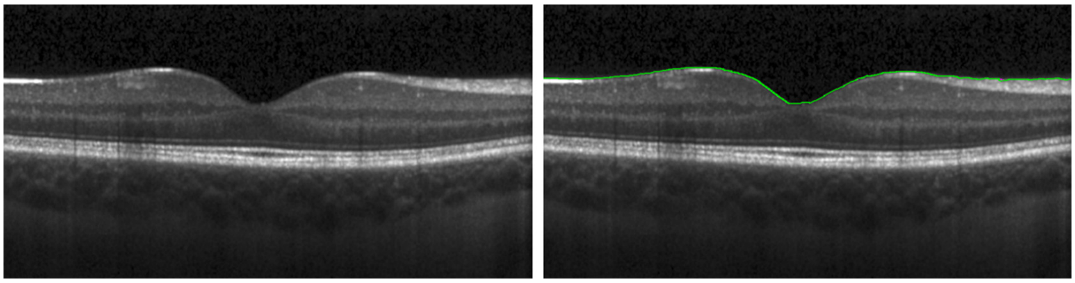

Fig. 10 Segmentation of the retina/vitreous boundary of a normal subject without a full-thickness macular hole. The left image is the original and the right image is the segmented image. Magenta denotes the shortest path segmentation and green denotes the length-adaptive segmentation. Where the magenta is not visible (most of the pixels in this case) the two methods segmented the same line.

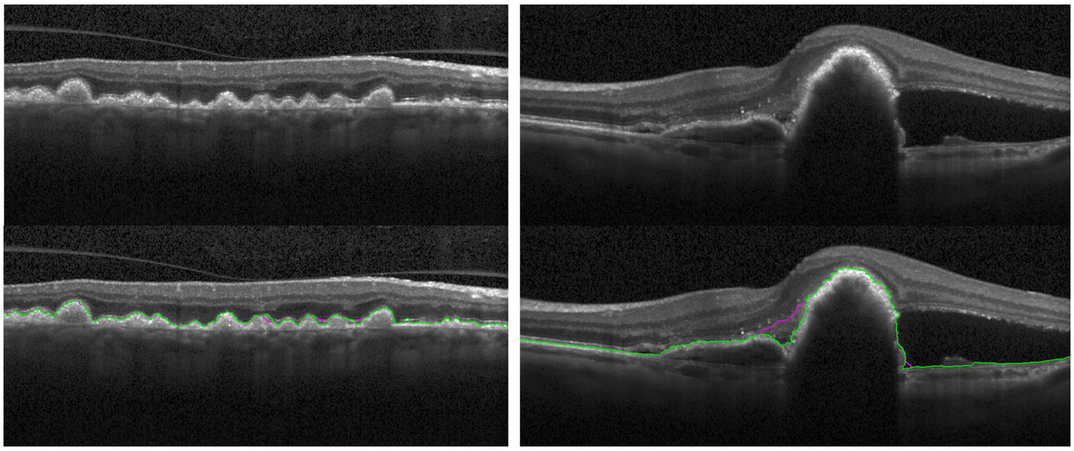

Fig. 11 Segmentation of the inner aspect of the RPE in patients with age-related macular degeneration using shortest path length and length-adaptive algorithms. The top row are the original images and the bottom row are the segmented images. In the segmented images, magenta denotes the classic shortest path segmentation and green denotes the proposed length-adaptive segmentation. Where the magenta is not visible the two methods segmented the same line.

vitreous boundary in patients with full-thickness macular holes. The data obtained from the length-adaptive segmentation could be used to estimate parameters such as macular hole height and base width. Our new method outperformed our popular GTDP method,[2] which is based on the shortest path metric, when compared to expert manual grading.

The length-adaptive algorithm we have presented is similar in theory to other graph search algorithms like Dijkstra's shortest path and Wimer's lowest mean arc length path in that we are searching through a graph for a path that minimizes a cost metric. As we explained in Sec. 2, what separates our algorithm from these two algorithms is the utilization of AMAL that is a compromise between total path length and mean arc length metrics. Moreover, since neither Dijkstra's or Wimer's techniques can utilize this new metric, we proposed a new graph search algorithm. This algorithm is an extension of the algorithm used by Chen et al.,[31] which is only applicable for the shortest path metric, modified to find the path with the lowest AMAL (Fig. 5) and account for cycles in the search space (Fig. 8). Because our algorithm performs a greedy search, it is not guaranteed that the solution produced is globally optimal. However, as demonstrated in our experiments, this shortcoming does not limit utilization in practical applications.

We note that although the length-adaptive method performed better than the shortest path algorithm in our experiments on pathologic eyes, it is not able to successfully segment every feature or image. For example, in the far right image in Fig. 9 the segmentation line does not perfectly adapt to the contour of the macular hole in the region with very weak gradient, suggesting that there is still room for improvement. The length-adaptive algorithm spans a significantly larger search space as compared with the shortest path algorithm because of its insensitivity to path length. In many cases, such as segmenting smooth structures like the ganglion cell layer in normal eyes, it is not reasonable to replace the shortest path method, which can be implemented efficiently via Dijkstra's algorithm, with an AMAL-based method. Thus, we expect that comprehensive software for segmentation of different anatomic and pathologic features in the retina will utilize both of the shortest path and AMAL metrics in its algorithm. In this framework, AMAL is only used for segmenting pathologic structures representing nonfunction features [e.g., Fig. 2(b)], while other structures are segmented using the shortest path method. This is an attainable goal because algorithms for detecting diseased eyes[26] or deformed or missing layers[39] from retinal OCT image have already been developed.

We also note on the reliance of our algorithm on a few empirically selected algorithmic parameters. We should emphasis that since selection of the algorithm's parameters is based on a dataset completely separate from the dataset in which the performance of the algorithm is evaluated upon, the results of this study are generalizable. We should also note that such empirical selection of segmentation parameters is explicitly or implicitly utilized in most of the other OCT layer segmentation algorithms.[2,13,14,40] Alternatively, automated algorithms presented in previous work[15] can be utilized to pick the parameters that reduce the error between manual and automatic segmentation boundaries in the training dataset.

In conclusion, we expect the AMAL metric to become a standard component of future graph-based automated segmentation methods. While efficient algorithms for implementing graph search using the shortest path metric are available online, there is a significant need for an algorithm to perform graph search using the AMAL metric. Thus, we have distributed freely online a general application open source software package coded in C++ for segmenting graphs using the length-adaptive algorithm. This software can be found in Ref. 41. Extending the test dataset and application to other types of pathology in ophthalmic images is part of our ongoing work.

Acknowledgments

We thank Prof. Robyn Guymer of the Centre for Eye Research Australia, University of Melbourne and Dr. Frederick L. Ferris III of the National Eye Institute for generously sharing the optical coherence tomography images in Fig. 11. This work was supported in part by grants from NIH R01, EY022691, and U.S. Army Medical Research Acquisition Activity Contract W81XWH-12-1-0397.

References

1. D. Huang et al., "Optical coherence tomography," *Science* **254**(5035), 1178–1181 (1991).
2. S. J. Chiu et al., "Automatic segmentation of seven retinal layers in SDOCT images congruent with expert manual segmentation," *Opt. Express* **18**(18), 19413–19428 (2010).
3. A. Mishra et al., "Intra-retinal layer segmentation in optical coherence tomography images," *Opt. Express* **17**(26), 23719–23728 (2009).
4. M. K. Garvin et al., "Automated 3-D intraretinal layer segmentation of macular spectral-domain optical coherence tomography images," *IEEE Trans. Med. Imaging* **28**(9), 1436–1447 (2009).
5. D. Cabrera Fernández, H. M. Salinas, and C. A. Puliafito, "Automated detection of retinal layer structures on optical coherence tomography images," *Opt. Express* **13**(25), 10200–10216 (2005).
6. A. Lang et al., "Retinal layer segmentation of macular OCT images using boundary classification," *Biomed. Opt. Express* **4**(7), 1133–1152 (2013).
7. S. J. Chiu et al., "Validated automatic segmentation of AMD pathology including drusen and geographic atrophy in SD-OCT images," *Invest Ophthalmol. Visual Sci.* **53**(1), 53–61 (2012).
8. H. Ishikawa et al., "Macular segmentation with optical coherence tomography," *Invest. Ophthalmol. Visual Sci.* **46**(6), 2012 (2005).
9. K. Vermeer et al., "Automated segmentation by pixel classification of retinal layers in ophthalmic OCT images," *Biomed. Opt. Express* **2**(6), 1743–1756 (2011).
10. Q. Yang et al., "Automated segmentation of outer retinal layers in macular OCT images of patients with retinitis pigmentosa," *Biomed. Opt. Express* **2**(9), 2493–2503 (2011).
11. S. J. Chiu et al., "Kernel regression based segmentation of optical coherence tomography images with diabetic macular edema," *Biomed. Opt. Express* **6**(4), 1172–1194 (2015).
12. B. J. Antony et al., "A combined machine-learning and graph-based framework for the segmentation of retinal surfaces in SD-OCT volumes," *Biomed. Opt. Express* **4**(12), 2712–2728 (2013).
13. J. Tian et al., "Real-time automatic segmentation of optical coherence tomography volume data of the macular region," *PLoS ONE* **10**(8), e0133908 (2015).
14. S. Huang et al., "*In vivo* microvascular network imaging of the human retina combined with an automatic three-dimensional segmentation method," *J. Biomed. Optics* **20**(7), 076003 (2015).
15. P. A. Dufour et al., "Graph-based multi-surface segmentation of OCT data using trained hard and soft constraints," *IEEE Trans. Med. Imaging* **32**(3), 531–543 (2013).
16. D. Kaba et al., "Retina layer segmentation using kernel graph cuts and continuous max-flow," *Opt. Express* **23**(6), 7366–7384 (2015).
17. J. Novosel et al., "Loosely coupled level sets for simultaneous 3D retinal layer segmentation in optical coherence tomography," *Med. Image Anal.* **26**(1), 146–158 (2015).
18. T. Zhang et al., "Fast retinal layer segmentation of spectral domain optical coherence tomography images," *J. Biomed. Opt.* **20**(9), 096014 (2015).

19. Q. Chen et al., "Automated choroid segmentation based on gradual intensity distance in HD-OCT images," *Optics Express* **23**(7), 8974–8994 (2015).
20. M. Zhang et al., "Advanced image processing for optical coherence tomographic angiography of macular diseases," *Biomed. Opt. Express* **6**(12), 4661–4675 (2015).
21. M. A. Mayer et al., "Retinal nerve fiber layer segmentation on FD-OCT scans of normal subjects and glaucoma patients," *Biomed. Opt. Express* **1**(5), 1358–1383 (2010).
22. V. Kajić et al., "Automated choroidal segmentation of 1060 nm OCT in healthy and pathologic eyes using a statistical model," *Biomed. Opt. Express* **3**(1), 86–103 (2012).
23. M. Hidalgo-Aguirre et al., "Automatic segmentation of the optic nerve head for deformation measurements in video rate optical coherence tomography," *J. Biomed. Opt.* **20**(11), 116008 (2015).
24. X. Yin, J. R. Chao, and R. K. Wang, "User-guided segmentation for volumetric retinal optical coherence tomography images," *J. Biomed. Opt.* **19**(8), 086020 (2014).
25. E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik* **1**(1), 269–271 (1959).
26. P. P. Srinivasan et al., "Fully automated detection of diabetic macular edema and dry age-related macular degeneration from optical coherence tomography images," *Biomed. Opt. Express* **5**(10), 3568–3577 (2014).
27. J. S. Duker et al., "The International Vitreomacular Traction Study Group classification of vitreomacular adhesion, traction, and macular hole," *Ophthalmology* **120**(12), 2611–2619 (2013).
28. A. X. Falcão et al., "User-steered image segmentation paradigms: live wire and live lane," *Graph. Models Image Process.* **60**(4), 233–260 (1998).
29. S. Wimer, I. Koren, and I. Cederbaum, "On paths with the shortest average arc length in weighted graphs," *Discrete Appl. Math.* **45**(2), 169–179 (1993).
30. J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(8), 888–905 (2000).
31. M. Chen et al., "Priority queues and Dijkstra's algorithm," Computer Science Department, University of Texas, Austin (2007).
32. L. Fang et al., "Fast acquisition and reconstruction of optical coherence tomography images via sparse representation," *IEEE Trans. Med. Imaging* **32**(11), 2034–2049 (2013).
33. D. Xu et al., "A novel segmentation algorithm for volumetric analysis of macular hole boundaries identified with optical coherence tomography-volumetric macular hole segmentation," *Invest. Ophthalmol. Visual Sci.* **54**(1), 163–169 (2013).
34. S. J. Chiu et al., "Automatic segmentation of closed-contour features in ophthalmic images using graph theory and dynamic programming," *Biomed. Opt. Express* **3**(5), 1127–1140 (2012).
35. Z. Wang et al., "Volumetric quantification of fibrous caps using intravascular optical coherence tomography," *Biomed. Opt. Express* **3**(6), 1413–1426 (2012).
36. G. J. Ughi et al., "Automated segmentation and characterization of esophageal wall in vivo by tethered capsule optical coherence tomography endomicroscopy," *Biomed. Opt. Express* **7**(2), 409–419 (2016).
37. S. Chitchian et al., "Automated segmentation algorithm for detection of changes in vaginal epithelial morphology using optical coherence tomography," *J. Biomed. Opt.* **17**(11), 116004 (2012).
38. A. Li et al., "Epidermal segmentation in high-definition optical coherence tomography," in *37th Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society (EMBC 2015)*, pp. 3045–3048 (2015).
39. P. P. Srinivasan et al., "Automatic segmentation of up to ten layer boundaries in SD-OCT images of the mouse retina with and without missing layers due to pathology," *Biomed. Opt. Express* **5**(2), 348–365 (2014).
40. J. Tian et al., "Automatic segmentation of the choroid in enhanced depth imaging optical coherence tomography images," *Biomed. Opt. Express* **4**(3), 397–411 (2013).
41. B. Keller et al., "Length adaptive graph search source code," www.duke.edu/~sf59/Keller_JBO_2016.htm (18 July 2016).

Biographies for the authors are not available.