Research Article Vol. 30, No. 2 / 17 Jan 2022 / Optics Express 2453
Optics EXPRESS

Globally optimal OCT surface segmentation using a constrained IPM optimization

HUI XIE,[1] ZHE PAN,[2,3] LEIXIN ZHOU,[1] FAHIM A. ZAMAN,[1] DANNY Z. CHEN,[4] JOST B. JONAS,[2,5] WEIYU XU,[1] YA XING WANG,[2,7] AND XIAODONG WU[1,6,8]

[1]*Department of Electrical and Computer Engineering, University of Iowa, Iowa City, IA 52242, USA*
[2]*Beijing Institute of Ophthalmology, Beijing Tongren Hospital, Capital University of Medical Science, Beijing Ophthalmology and Visual Sciences Key Laboratory, Beijing, China*
[3]*Department of Ophthalmology, Peking University Third Hospital, Beijing 100191, China*
[4]*Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556, USA*
[5]*Department of Ophthalmology, Medical Faculty Mannheim, Heidelberg University, 68167 Mannheim, Germany*
[6]*Department of Radiation Oncology, University of Iowa, Iowa City, IA 52242, USA*
[7]*yaxingw@gmail.com*
[8]*xiaodong-wu@uiowa.edu*

Abstract: Segmentation of multiple surfaces in optical coherence tomography (OCT) images is a challenging problem, further complicated by the frequent presence of weak boundaries, varying layer thicknesses, and mutual influence between adjacent surfaces. The traditional graph-based optimal surface segmentation method has proven its effectiveness with its ability to capture various surface priors in a uniform graph model. However, its efficacy heavily relies on handcrafted features that are used to define the surface cost for the "goodness" of a surface. Recently, deep learning (DL) is emerging as a powerful tool for medical image segmentation thanks to its superior feature learning capability. Unfortunately, due to the scarcity of training data in medical imaging, it is nontrivial for DL networks to *implicitly* learn the global structure of the target surfaces, including surface interactions. This study proposes to parameterize the surface cost functions in the graph model and leverage DL to learn those parameters. The multiple optimal surfaces are then simultaneously detected by minimizing the total surface cost while *explicitly* enforcing the mutual surface interaction constraints. The optimization problem is solved by the primal-dual interior-point method (IPM), which can be implemented by a layer of neural networks, enabling efficient end-to-end training of the whole network. Experiments on spectral-domain optical coherence tomography (SD-OCT) retinal layer segmentation demonstrated promising segmentation results with sub-pixel accuracy.

1. Introduction

Automated surface segmentation with high accuracy for retina optical coherence tomography (OCT) is a clinical necessity in many diagnostic and treatment tasks of ophthalmic diseases. Examples include quantification of retinal layer morphology and thickness in glaucoma, age-related macular degeneration, diabetic macular edema, and using it for treatment decisions with retinal OCT [1,2]. In retina OCT imaging, the layered surfaces that need to be identified appear in mutual interactions. These surfaces are "coupled" in a way that their topology and relative positions are usually known already (at least in a general sense), and the distances between them are within some specific ranges. Incorporating these surface-interrelations into the segmentation can further improve its accuracy and robustness, especially when insufficient image-derived information is available for defining some object boundaries or surfaces. Such insufficiency can be remedied by using clues from other related boundaries or surfaces. Simultaneous optimal

#444369 https://doi.org/10.1364/OE.444369
Journal © 2022 Received 27 Sep 2021; revised 12 Dec 2021; accepted 1 Jan 2022; published 11 Jan 2022

Optics EXPRESS

detection of multiple coupled surfaces thus yields superior results compared to the single-surface detection approaches [3].

Many retina OCT segmentation methods have been proposed in past years. Garvin *et al.* first introduced the graph-based optimal surface segmentation method [3] to surface segmentation in retinal OCT [4,5], which was further developed by incorporating various *a priori* knowledge reflecting anatomic and imaging information [6–9]. The methods can model the interacting retinal surfaces in a complex multi-layered graph, enabling simultaneous segmentation of all desired 3D surfaces in a single optimization process with guaranteed global optimality. Chiu *et al.* modeled the retinal layer segmentation in OCT as computing shortest paths in a directed acyclic graph for each B-scan [10]. Due to its computational efficiency, the shortest-path-based method has attracted much attention for retinal OCT segmentation [11–13]. All these methods work best in 2D while lacking the capability of making use of the contextual information between B-scans. Other known OCT surface segmentation approaches include level set [14–18], probabilistic global shape model [19], and random forest classifier [20,21]. Each of these traditional methods has its own strength. They all share a common drawback that is their dependence on handcrafted features.

Armed with superior data representation learning capacity, deep learning (DL) methods are emerging as powerful alternatives to traditional segmentation algorithms for many medical image segmentation tasks [22,23]. Both fully convolutional networks (FCN) [24,25] and U-Net [26–28] have been utilized for retinal layer segmentation in OCT images. All these methods model the segmentation problems as a pixel-wise classification problem, in which each pixel is labeled as one of the retinal layers or background. As such, the retinal layering topology cannot be guaranteed with those methods, neither the continuity and smoothness of the retinal surfaces can be ensured. Especially, due to the scarcity of training data in medical imaging, it is nontrivial for DL networks to *implicitly* learn those global structures of the target surfaces. To address these limitations, the graph-based method [10] was used as post-processing for the deep learning models to enforce surface monotonicity and smoothness [29,30]. Pekala *et al.* utilized a dense-blocked FCN with Gaussian regression as postprocessing for multiple OCT surface segmentation [31]. In this scheme, feature learning is, in fact, disconnected from the graph model; the learned features thus may not be truly appropriated for the model. Shah *et al*. [32] first formulated the retinal OCT surface segmentation as a regression problem using an FCN followed by fully connected layers to directly extract the layered multiple surfaces simultaneously from OCT. The direct surface segmentation encodes the monotonicity prior of each surface (i.e., each surface intersects every A-scan of the OCT image exactly once) within the deep network by column-wise regression. He *et al.* further extended the deep regression idea with fully differentiable soft-argmax operations to generate surface positions followed by ReLU operations to guarantee the surface order in their fully convolutional regression network (FCRN) [33]. Their method has been demonstrated to achieve state-of-the-art results for multiple surface segmentation in retinal OCT. Though being able to achieve highly accurate structured surfaces, unlike model-based approaches, the FCRN method is more prone to be affected by outlier images with bad quality or artifacts with limited training data [33]. Furthermore, the ReLU topology guarantee module in the FCRN method just updates the surfaces with ReLU operations from top to bottom to correct the surface order, which may propagate the errors in the upper surfaces to the lower surfaces.

This study proposes to unify the powerful feature learning capability of DL with the successful graph-based surface segmentation model in a single deep neural network for end-to-end training to achieve globally optimal segmentation of multiple interacting surfaces; the surface order is explicitly enforced with linear constraints to the objective function of the segmentation problem. In the proposed segmentation framework, the surface costs are parameterized, and the DL network is leveraged to learn the model from the training data to determine the parameters for the

Optics EXPRESS

input image. The multi-surface inference by minimizing the total surface cost while satisfying the surface order constraints is realized by the primal-dual Interior-Point Method (IPM) for constrained convex optimization. It can be implemented by a layer of neural networks with efficient backward propagation of gradients with virtually no additional cost [34]. Thus, the DL network for surface cost parameterization can be seamlessly integrated with the multi-surface inference to achieve the end-to-end training with a global optimization guarantee.

2. Methods

2.1. Problem formulation

Let $\mathcal{I}(X, Y, Z)$ of size $X \times Y \times Z$ be a given 3-D volumetric image, as oriented in Fig. 1. For each (x, y) pair ($x = 0, 1, \ldots, X-1$ and $y = 0, 1, \ldots, Y-1$), the voxel subset $\{\mathcal{I}(x, y, z) | 0 \leq z < Z\}$ forms a column parallel to the **z**-axis, denoted by $q(x, y)$. The optimizing target is to find $N > 1$ most possible terrain-like surfaces $S = \{S_0, S_1, \ldots, S_{N-1}\}$, where N depends different application contexts and is decided by domain experts. Each surface intersects every column $q(x, y)$ at exactly one location point z_i, where $i = 0, 1, 2, \ldots, N-1$.

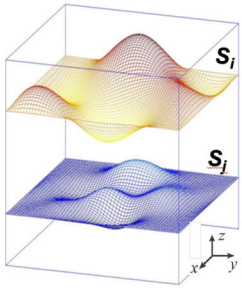

Fig. 1. Terrain-like surfaces S_i and S_j intersect each $q(x, y)$-column exactly one time, and adjacent surfaces do not interfere with each other.

In the graph-based surface segmentation model [3,6,8], each voxel $\mathcal{I}(x, y, z)$ is associated with an on-surface cost $c_i(x, y, z)$ for each sought surface location z_i at column $q(x, y)$ crossing with surface S_i, which is inversely related to the likelihood that the desired surface location z_i contains the voxel, and is computed based on handcrafted image features. The *surface cost* of S_i is the total on-surface costs of all voxels on S_i. The on-surface cost function $c_i(x, y, z)$ for the column $q(x, y)$ can be an arbitrary function in the graph model. However, an ideal cost function $c_i(x, y, z)$ should express a certain type of convexity: as the aim is to minimize the surface cost, $c_i(x, y, z)$ should be low at the surface location; while the distance increases from the surface location along the column $q(x, y)$, the cost should increase accordingly. This study proposes to leverage DL networks to learn a Gaussian distribution $\mathcal{G}(\mu_i(q), \sigma_i(q))$ to model the on-surface cost function $c_i(x, y, z_i)$ for each column $q(x, y)$, that is, $c_i(x, y, z_i) = \frac{(z_i - \mu_i)^2}{2\sigma_i^2}$. Thus, the surface cost of z_i is parameterized with (μ_i, σ_i).

For multiple surfaces segmentation without adjacent surfaces interferences in OCT application context, a surface interacting constraint is added to every column $q(x, y)$ for each pair of the sought adjacent surface locations z_i and z_{i+1}. For each $q(x, y)$, $z_{i+1}(q) - z_i(q) \leq 0$ indicates that z_i is always on the top of z_{i+1}. The multi-surface segmentation is formulated as an optimization

Optics EXPRESS

problem, where the parameterized surface costs are derived using deep CNNs:

$$S^* = \underset{S}{\operatorname{argmin}} \sum_{i=0}^{N-1} \sum_{\mathcal{I}(x,y,z_i)\in S_i} c_i(x,y,z_i)_{|(\mu_i,\sigma_i)} \quad \text{s.t.} \quad z_{i+1}(q) - z_i(q) \le 0, \quad i = 0,1,\ldots,N-2, \forall q(x,y). \tag{1}$$

2.2. Surface segmentation network

As shown in Fig. 2, The proposed segmentation network consists of two integrative components: One aims to learn the surface cost parameterization (μ, σ) in Eq. (1); the other strikes to solve the optimal surface interference by optimizing Eq. (1) with the IPM optimization module. Thus, the whole network can then be trained in an end-to-end fashion and outputs globally optimal solutions for the multiple surface segmentation.

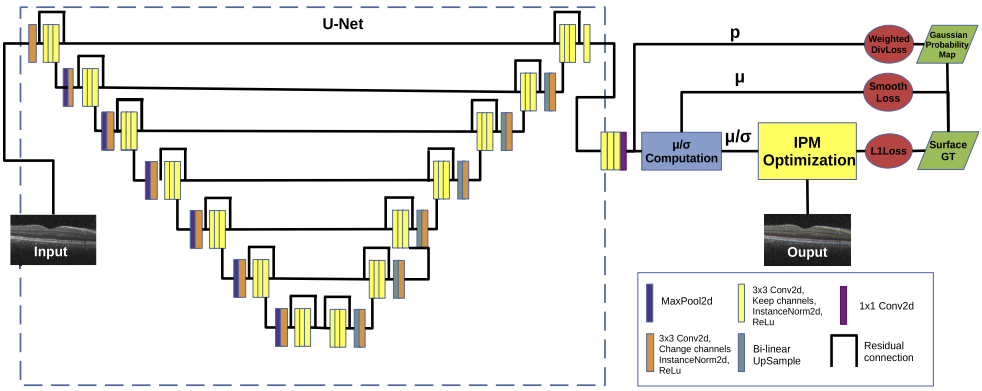

Fig. 2. Illustration of the network architecture of the proposed multiple surface segmentation. The surface cost is parameterized with $(\vec{\mu}, \vec{\sigma})$, which models the Gaussian distribution of the surface locations along each image column. IPM Optimization indicates primal-dual Interior-Point Method for constrained convex optimization. Weighed DivLoss is an image-gradient weighted divergence loss. GT denotes ground truth.

Surface Cost Parameterization. This study utilizes U-net [35] as the backbone for surface feature extraction. The input of the U-net consists of five channels – the original image, the directional gradients of the original image along the **x**- and **z**- directions, and the magnitude and direction of the image gradient. The implemented U-net has seven layers with a filter size of 24 at the first layer and with long skip connections between the corresponding blocks of its encoder and decoder. Each block has three convolution layers with a residual connection [36]. The output feature maps of the U-net module are then fed into a segmentation head (Fig. 2) which is implemented with three-layer convolutions followed by a 1×1 convolution and softmax along every image column for each surface to obtain the probability maps for the N surfaces. Note that each sought surface S_i intersects every image column exactly once at location z_i.

As in He *et al.*'s methods [33,37], for each surface location z_i, based on the surface location probability $p_i(z)$ on every image column $q(x,y)$ from the segmentation head, the expected surface location $\mu_i = \sum_{z=0}^{Z-1} z p_i(z)$. The surface location distribution of z_i on column q is modeled with a Gaussian $\mathcal{G}_i(\mu_i, \sigma_i)$, with

$$\sigma_i^2 = \sum_{z=0}^{Z-1} p_i(z)(z-\mu_i)^2, \tag{2}$$

where $\sum_{z=0}^{Z-1} p_i(z) = 1.0$ by the softmax operation on each image column.

Optics EXPRESS

The surface cost $\sum_{\mathcal{I}(x,y,z)\in S_i} c_i(x,y,z_i)_{|(\mu_i,\sigma_i)}$ of each surface location z_i is parameterized with (μ_i, σ_i), such that

$$\sum_{\mathcal{I}(x,y,z_i)\in S_i} c_i(x,y,z_i)_{|(\mu_i,\sigma_i)} = \sum_{\mathcal{I}(x,y,z_i)\in S_i} \frac{(z_i-\mu_i)^2}{2\sigma_i^2}. \tag{3}$$

Globally optimal multiple surface inference. Given the surface cost parameterization $(\vec{\mu}, \vec{\sigma})$, the inference of optimal multiple surfaces can be solved by optimizing Eq. (1), which is a constrained convex optimization problem. To achieve end-to-end training, the optimization inference needs to be able to provide gradient backward propagation, which impedes the use of traditional convex optimization techniques in deep learning. This study exploits the OptNet technique [34] to integrate a primal-dual interior-point method (IPM) for solving Eq. (1) as an individual layer in the surface segmentation network (Fig. 2). Based on Amos and Kolter's theorem [34], with a full differential technology, the residual equation $\vec{r}(\vec{z^*}, \vec{\lambda}) = \vec{0}$ derived from the Karush-Kuhn-Tucker conditions, with the hidden parameters $(\vec{\mu}, \vec{\sigma})$ in the residual equation, where $\vec{\lambda}$ is the Lagrangian variable and $\vec{z^*}$ is optimized N-surface locations, can be converted into a full differential equation. Then deduce the gradients of $\frac{dL}{d\vec{\mu}}$ and $\frac{dL}{d\boldsymbol{Q}}$ for backward propagation, where L is the training loss of the whole deep learning network and $\boldsymbol{Q}$ is a diagonal matrix with $\boldsymbol{Q} = Diag[\frac{1}{\sigma_0^2}, \frac{1}{\sigma_1^2}, \ldots, \frac{1}{\sigma_{N-1}^2}]$. The deduction details of the matrix formulas for forwarding optimization and backward gradients are in Appendix A. The IPM method is a 2nd order Newton method with matrix inversions. In the OCT surface segmentation, it typically needs less than 10 iterations to converge, which supports high-epoch training for the proposed deep model.

2.3. Network training strategy

Multiple loss functions are introduced to focus on the training of different modules in the proposed multiple-surface segmentation network (Fig. 2). In the segmentation head, the softmax operations are performed on each image *column* to obtain the probability of each voxel that would be on a specific surface.

To encourage the segmentation head to output reasonable probability maps, a novel image-gradient weighted divergence loss L_{Div} is utilized for training. It inherits from the Kullback–Leibler divergence loss KLDloss, which measures how one probability distribution is different from a reference one. Observing that layered surfaces appear on locations with high image gradients, this study thus proposes to make use of image gradients as weighted coefficients for KLDloss to highlight the probability differences on those surface locations, which provides attention mechanism [38] for the segmentation network. The proposed L_{Div} for any image column q is calculated, as follows.

$$L_{Div} = \sum_{i\in q} w_i g_i \| \log \frac{g_i}{p_i} \|,$$

where g_i is the Gaussian probability of pixel i on the column q, which is computed by using the ground truth surface location as the mean and a fixed standard deviation σ for each pixel i on the column q; the standard deviation σ is $4-6\%$ of the B-scan height in pixel to allow possible oscillation of surface locations; p_i is the network-output probability of pixel i, and $w_i \in W$ is the magnitude of the gradient of the original image I at pixel i. The standard deviation of ground truth distribution g_i is related with Z, the height of A-scan. Bigger Z, bigger standard deviation of g_i. The magnitude W of the image gradient is computed with $W = 1 + \alpha\|\nabla(I)\|$, where α is an experimental constant to balance the effect of gradient to the weight of L_{Div}. The L_{Div} loss is used to ensure that the network-output probability map p be well aligned with the ground truth Gaussian probability map g, especially on those pixels with a large image gradient.

In addition to using an L_1 loss, denoted as L_1, to measure the difference between the predicted surface and ground truth, a smoothness loss, L_{smooth}, is introduced to regularize the smoothness

Optics EXPRESS

and mutual interaction of the sought surfaces. More precisely, L_{smooth} in Eq. (5) below is the total sum of the mean-squared-errors (MSEs) of the surface location changes between any two adjacent image columns while compared to the corresponding ground truth, plus the total sum of the MSEs of predicted layer thickness on all columns while compared to the ground truth of the layer thickness.

$$\begin{aligned} L_{smooth} = & \frac{1}{N \cdot (X-1)} \sum_{n=0}^{N-1} \sum_{x=0}^{X-2} ((\mu_{n,x} - \mu_{n,x+1}) - (s_{n,x} - s_{n,x+1}))^2 \\ & + \frac{1}{(N-1) \cdot X} \sum_{n=0}^{N-2} \sum_{x=0}^{X-1} ((\mu_{n+1,x} - \mu_{n,x}) - (s_{n+1,x} - s_{n,x}))^2, \end{aligned} \tag{4}$$

where N is the number of surfaces, X is the number of A-scans in one B-scan, μ is predicted surface locations, and s is ground truth surface locations.

The whole network loss $L = L_{Div} + L_{smooth} + wL_1$, where w is a coefficient used to alleviate the effect of weak gradients while the prediction is close to the ground truth. Thus, it enables the network to put more weight on the L_1 loss among all three loss functions as our goal is to minimize the error between the predicted surfaces and the ground truth.

3. Experiments

The proposed method was validated on two SD-OCT datasets for segmenting $N = 9$ retinal surfaces. All experiments of this study experiencedly chose the weight of L_1 loss $w = 10$ for all different application data sets. Ablation experiments were also performed to examine the contribution of each component in the proposed segmentation network.

3.1. Beijing Eye Study OCT dataset

47 participants were randomly selected from all OD eyes of the population-based Beijing Eye Study 2011 (BES) [39,40], in which 3468 participants aged 50+ years were enrolled. All participants have scans on the macula and optic nerve head by SD-OCT (Heidelberg Engineering, Inc., Germany) with a pixel resolution of 3.87 μm in the height ($\mathbf{z}$-axis) direction. Each volume has scans consisting of 31 single lines on the $30^\circ \times 30^\circ$ field centered on the macula. The horizontal area of the scan was reduced to 20° centered on the macula to remove the optic disc region. Nine boundary surfaces in Table 1 of all 47 OCT volumes (Fig. 3) were first delineated by OCT-Explorer 3.8 [41], a graph-search based ophthalmic image analysis software [5,42], and then were manually corrected by an experienced ophthalmologist.

Table 1. The definition of nine surfaces of BES OCT dataset.

Abbreviation	Full Name
ILM	internal limiting membrane
RNFL-GCL	the surface between retinal nerve fiber layer and ganglion cell layer
IPL-INL	the surface between inner plexiform layer and inner nuclear layer
INL-OPL	the surface between inner nuclear layer and outer plexiform layer
OPL-ONL	the surface between outer plexiform layer and outer nuclear layer
MZ-EZ	the surface between myoid zone and ellipsoid zone of inner segments of photoreceptors
IS-OS	the inner/outer photoreceptor segments junction
IB_RPE	the inner boundary of the retinal pigment epithelium
OB_RPE	the outer boundary of the retinal pigment epithelium

This experiment used a fixed $\sigma = 20$ in pixels to generate the ground truth for the Gaussian distribution of surface location on each image column. The Gaussian and salt & pepper noise were

Optics EXPRESS

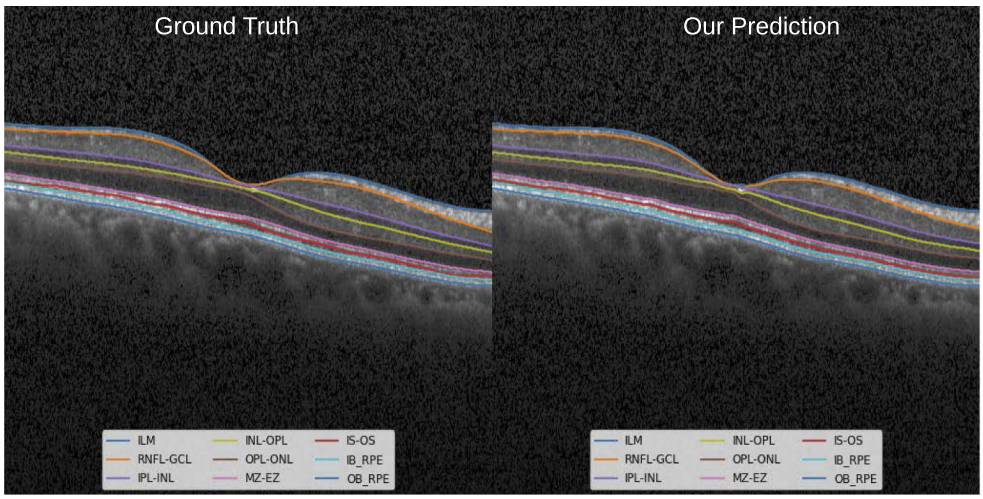

Fig. 3. Simultaneous segmentation of 9 intraretinal surfaces in an SD-OCT image of BES OCT dataset.

added for data augmentation. We expect that adding speckle-noise would be further improving data augmentation. 10-fold cross-validation was performed to evaluate our method: 8 folds for training, 1 fold for validation, and 1 fold for testing. The mean absolute surface distance errors (MASD) for each sought surface over the testing results on all 47 scans are shown in Table 2. Sample segmentation results are illustrated in Fig. 3. Compared to OCT-Explorer [41], Table 2 shows that our method significantly improves the segmentation accuracy indicated by the MASD errors and the robustness indicated by the standard deviations.

Table 2. Mean Absolute Surface Distance error ± standard deviation (in μm) evaluated on the BES dataset for segmenting 9 retinal surfaces with mean of 10-fold cross validation over all test folds. The OCT-Explorer [41] is based on the graph search method. Penetration resolution is 3.87μm/pixel.

	ILM	RNFL-GCL	IPL-INL	INL-OPL	OPL-ONL	MZ-EZ	IS-OS	IB_RPE	OB_RPE	Average
OCT-Explorer	1.79±4.34	3.58±4.75	2.92±4.77	2.54±4.77	2.73±4.72	1.79±4.74	8.61±5.35	1.82±4.72	1.78±4.72	3.06±5.15
Our Method	0.98±0.09	2.98±0.41	2.59±0.47	2.38±0.43	2.70±0.65	1.43±0.49	2.82±0.70	1.53±0.28	1.21±0.19	2.07±0.91

The experiments were performed on an Nvidia Titan X GP102 GPU server with 12GB GPU memory. For the network training, An Adam optimizer was used, with an initial learning rate of 0.01 and without weight decay. We reduced the learning rate on plateaus with patience 20 and used the batch size of 4 to train the proposed network. It took 17.5 hours and 302 epochs to get convergence. In the inference phase, the proposed model took 3.8 seconds on average to segment 9 surfaces for a 3D OCT volume consisting of 31 B-scans, each with 496×512 pixels.

3.2. Public JHU OCT MS dataset

The public JHU retinal OCT MS dataset [43] includes 35 human retina scans acquired on a Heidelberg Spectralis SD-OCT system, of which 14 are healthy controls (HC) and 21 have a diagnosis of multiple sclerosis (MS). Each patient has 49 B-scans with pixel size 496×1024, and nine ground truth surfaces on each B-scan. The z-axial resolution in each A-scan is 3.87 μm/pixel. The original images were manually delineated with 21 control points on each surface, and then a cubic interpolation was performed on each B-scan to obtain the ground truth by a Matlab script [33,37]. Each B-scan was cropped to keep the center 128 rows to form a 128×1024 image.

Optics EXPRESS

In this experiment, the proposed segmentation model was trained on the last 6 HC and last 9 MS subjects according to the order of their IDs and tested on the other 20 subjects, which is the same experimental configuration as in He *et al.*'s [33,37]. A fixed $\sigma = 8$ in pixels was used to generate a Gaussian probability map. The Gaussian and salt & pepper noise were used for data augmentation. The MASD errors for the proposed and He *et al.*'s methods are shown in Table 3. Our method improved the segmentation accuracy for 6 out of 9 surfaces as well as the average segmentation error over all nine surfaces. Compared to He *et al.*'s method, the proposed method also achieved improved standard deviations, which demonstrated the robustness of our method. Sample segmentation results are illustrated in Fig. 4.

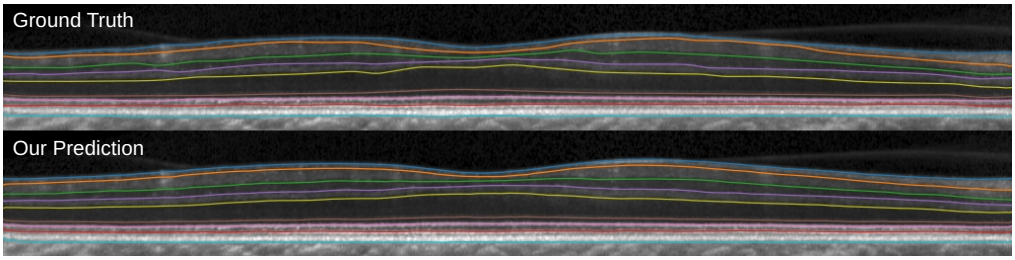

Fig. 4. Simultaneous segmentation of 9 intraretinal surfaces in an SD-OCT image of JHU OCT dataset.

Table 3. Mean Absolute Surface Distance error ± standard deviation (in μm) evaluated on JHU OCT MS Data test set for the delineation of 9 surfaces, comparing AURA toolkit [20], R-Net [44], ReLayNet [26], SP (shortest path) [33], FCRN [33] and our proposed method. The first 5 experimental results are directly from Table 1 of He *et al.*'s work [33]. Numbers in bold are the best in that row. Penetration resolution is 3.87μm/pixel.

Methods	AURA	R-Net	ReLayNet	SP	FCRN	OurMethod
ILM	2.37±0.36	2.38±0.36	3.17±0.61	2.70±0.39	2.41±0.40	**2.32**±0.27
RNFL-GCL	3.09±0.64	3.10±0.55	3.75±0.84	3.38±0.68	**2.96**±0.71	3.07±0.68
IPL-INL	3.43±0.53	2.89±0.42	3.42±0.45	3.11±0.34	2.87±0.46	**2.86**±0.33
INL-OPL	3.25±0.48	3.15±0.56	3.65±0.34	3.58±0.32	**3.19**±0.53	3.24±0.60
OPL-ONL	2.96±0.55	2.76±0.59	3.28±0.63	3.07±0.53	**2.72**±0.61	2.73±0.57
ELM	2.69±0.44	2.65±0.66	3.04±0.43	2.86±0.41	2.65±0.73	**2.63**±0.51
IS-OS	2.07±0.81	2.10±0.75	2.73±0.45	2.45±0.31	2.01±0.57	**1.97**±0.57
OS-RPE	3.77±0.94	3.81±1.17	4.22±1.48	4.10±1.42	3.55±1.02	**3.35**±0.83
BM	2.89±2.18	3.71±2.27	3.09±1.35	3.23±1.36	3.10±2.02	**2.88**±1.63
Overall	2.95±1.04	2.95±1.10	3.37±0.92	3.16±0.88	2.83±0.99	**2.78**±0.85

3.3. Ablation experiments

To examine the contribution of each major component in the proposed segmentation network, an ablation study was conducted to compare the performances of its major components, denoted as follows – (1) using a surface smoothness loss L_{smooth}; (2) using additional four image gradient channels as input; (3) using weighted KLD loss L_{div}; (4) using the IPM module.

The ablation experiments were conducted on the JHU OCT MS data. Each ablation experiment only removed one of those four components from the proposed network and assessed its segmentation performance. All the ablation experiments used the same sets of training, validation,

Optics EXPRESS

and test data. The more the MASD error increases, the more important the corresponding component is.

The ablation experimental results are shown in Table 4, and the visualization samples are illustrated in Fig. 5. This study observed that the use of weighted KLD Loss, as well as the use of image gradients as additional input channels, leads to a big boost to the segmentation performance of the proposed model. The use of the surface smoothness loss and the IPM module also yield reasonable improvement on segmentation accuracy. From the visualization examples in Fig. 5, the use of the smoothness loss especially helps to obtain smoother surfaces. Without the IPM module, the network cannot guarantee the correct surface order, as demonstrated in Fig. 5. Our method achieved higher accuracy in 7 out of 9 surfaces than the method without using the IPM module (NoIPM). For the two surfaces RNFL-GCL and INL-OPL, NoIPM obtained slightly higher accuracy. It is possible due to the matrix inverse computation in the IPM optimization as this study used pseudo-inverse, which may yield inaccurate gradient optimization directions in singular matrix cases.

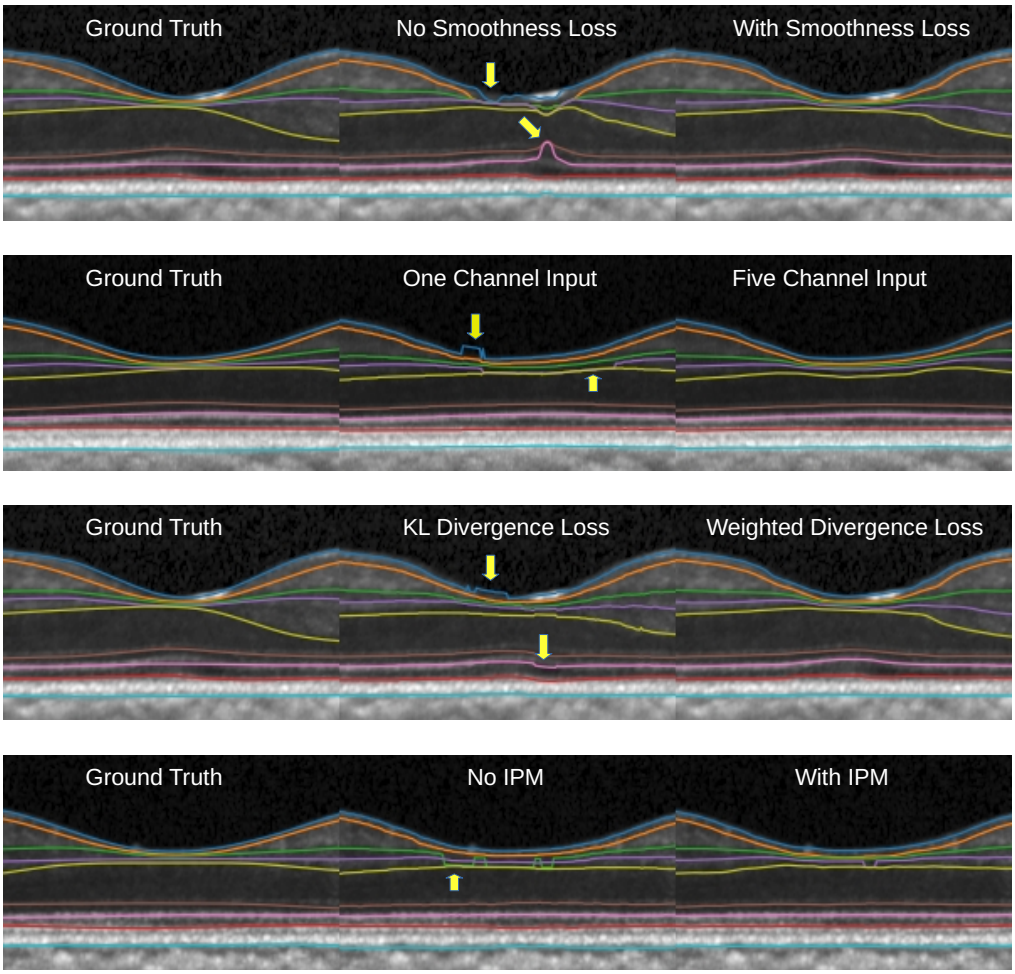

Fig. 5. Sample segmentations on JHU OCT MS data for 4 ablation experiments. All images are cropped from A-scan 400 to 600 in JHU OCT MS test set. They show that smoothness loss, input gradient channels, and weighted divergence loss improved the smoothness of surfaces, and IPM module guarantees the surface order.

Optics EXPRESS

Table 4. Mean Absolute Surface Distance error ± StdDev (μm) of ablation experiments on JHU OCT MS Data test set.

Methods	NoSmoothnessLoss(1)	OneChannelInput(2)	KLDivLoss(3)	NoIPM(4)	OurMethod
ILM	2.44±0.38	2.42±0.36	2.50±0.48	2.36±0.38	2.32±0.27
RNFL-GCL	3.12±0.64	3.25±0.84	3.24±0.89	2.89±0.66	3.07±0.68
IPL-INL	3.00±0.46	3.03±0.43	3.02±0.43	2.86±0.44	2.86±0.33
INL-OPL	3.25±0.58	3.27±0.43	3.25±0.49	3.07±0.45	3.24±0.60
OPL-ONL	2.82±0.53	2.88±0.71	3.26±0.80	2.78±0.61	2.73±0.57
ELM	2.64±0.81	2.67±0.58	2.62±0.74	2.67±0.61	2.63±0.51
IS-OS	2.01±0.81	1.95±0.71	2.02±0.61	2.08±1.00	1.97±0.57
OS-RPE	3.40±0.71	3.45±0.83	3.64±0.82	3.62±0.98	3.35±0.83
BM	2.83±1.80	2.90±1.88	2.96±1.32	3.12±2.28	2.88±1.63
Overall	2.83±0.92	2.87±0.96	2.94±0.89	2.83±1.06	2.78±0.85

4. Discussion and conclusion

This study proposes a novel method of using deep neural networks for multiple interacting surface segmentation from retina OCT. The proposed method seamlessly integrates the graph-based surface segmentation model into a deep network for end-to-end learning to achieve globally optimal surface segmentation. The graph-based surface segmentation model is parameterized, which enables to make use of the powerful feature learning capability of DL to automatically learn the model parameters. A constrained IPM convex optimization module, which can be implemented by layers of neural networks with efficient backward propagation of gradients, is then used to solve the inference optimization problem of graph-based surface segmentation. Thus, the global optimality of multiple interacting surface segmentation can be achieved while subject to the surface topology constraints. This is in contrast to He *et al.*'s FCRN method [33], in which, though surface topology can be guaranteed, the surface updates by ReLU operations for that purpose may compromise segmentation accuracy. In addition, The proposed method with the integration of the graph-based surface segmentation model into the deep network is more robust to the outlier images with bad quality or artifacts, as demonstrated by smaller error standard deviations in the experiments. The newly introduced weighted KLD loss based on the image gradients provides a certain attention mechanism for the network to focus on specific feature learning for retina surfaces, further improving segmentation accuracy.

There are also limitations to the proposed work. The proposed segmentation method was implemented in our experiments for segmenting each 2D B-scan individually, which does not make use of the contextual information between B-scans. This may cause the breaking of the surface smoothness constraints across B-scans, though the order of those segmented surfaces is still maintained. This problem can be solved by extending our network using 3D convolutional filters or leveraging adjacent B-scan information within the network. The current constraint $\boldsymbol{A}\vec{z} \leq \vec{0}$ in the proposed model is merely a surface ordering constraint. In practice, the thickness range of each retinal layer may be well known. These thickness priors can be effectively incorporated into our proposed model by introducing additional constraints. For example, for each column $q(x, y)$, we can enforce $\delta_{ij}(q) \leq S_i(q) - S_j(q) \leq \Delta_{ij}(q)$, where $\delta_{ij}(q)$ and $\Delta_{ij}(q)$ are two specified minimum and maximum distances between surfaces S_i and S_j, respectively, with S_i on top of S_j, and $S(q)$ denotes the surface location of S on column $q(x, y)$. In this way, the parameters $\delta_{ij}(q)$ and $\Delta_{ij}(q)$ specify the thickness range of the layer bounded by the surfaces S_i and S_j. The IPM optimization is ready to handle those layer thickness constraints. We can also go a step further to develop a deep CNN network to learn those thickness parameters from data.

Optics EXPRESS

In summary, a novel DL framework for segmenting multiple interacting surfaces in retina OCT is proposed with end-to-end training. The globally optimal solutions are achieved by seamlessly integrating two DL components: one for surface cost parameterization with a Gaussian model and the other for solving the inference optimization of the graph-based surface segmentation model while explicitly enforcing the surface mutual interaction constraints. The method has the potential to be adapted for other structured terrain-like surface segmentation problems in medical imaging.

Appendix. IPM optimization

In this appendix, we describe the derivations of the forward and backward propagations of the IPM optimization. The forward propagation algorithm used the standard IPM (interior-point method) algorithm [45], and the backward propagation algorithm used the full differential idea of OptNet [34].

Because the optimization model in Eq. (1) only considers the surface interaction along each A-scan, in experiments, we implemented the model parallel for all columns in batched 2D images. Multi-surfaces in a 2-D image can be viewed as a series of smooth curves. A feasible surface intersects with each image column (along **z**-axis) exactly once. We implemented the math optimization model for each column $q(x, y)$ along **z**-axis, and parallelized these models on GPU for all the columns $q(x, y)$ in batched 2D input images.

Notations: We use bold characters to represent matrices, and use hat arrows to represent vectors. We use $Diag[a_0, a_1, a_2, \ldots, a_{N-1}]$ to represent a diagonal matrix of size $N \times N$ with a_0, a_1, a_2,. . . , and a_{N-1} as its diagonal elements. We use vector $\vec{z} = [z_0, z_1, \ldots, z_{N-1}]^T$ to denote N surfaces along a column $q(x, y)$, and use similar conventions for vectors $\vec{\mu}$ and $\vec{\sigma}$.

Before we proceed, we use $\boldsymbol{A}\vec{z} \leq \vec{0}$ to compactly represent the constraints $z_{i+1}(q) - z_i(q) \leq 0$ expressing all 2-adjacent-surface relations in the context of **z**-axis pointing downward, where the matrix $A \in R^{(N-1)\times N}$ is given as follows:

$$\boldsymbol{A} = \begin{bmatrix} 1 & -1 & 0 & \cdots & 0 \\ 0 & 1 & -1 & \cdots & 0 \\ & & \ddots & \ddots & \\ 0 & \cdots & 0 & 1 & -1 \end{bmatrix}. \tag{5}$$

The implemented column model makes sure the locations of the final surfaces $\vec{z}$ satisfy surface order constraints, while trying to achieve minimal deviations of $\vec{z}$ from initial prediction $\vec{\mu}$ under confidence index $\vec{\sigma}$. Therefore, a constrained optimization column model in matrix form can be formulated as follows:

$$\vec{z}^* = \underset{\vec{z}}{\operatorname{argmin}} \frac{1}{2}(\vec{z} - \vec{\mu})^T \boldsymbol{Q}(\vec{z} - \vec{\mu}), \tag{6a}$$

$$s.t. \quad \boldsymbol{A}\vec{z} \leq \vec{0}, \tag{6b}$$

where $\vec{z} = [z_0, z_1, \ldots, z_{N-1}]^T$, $\vec{\mu} = [\mu_0, \mu_1, \ldots, \mu_{N-1}]^T$, $\boldsymbol{Q} = Diag[\frac{1}{\sigma_0^2}, \frac{1}{\sigma_1^2}, \ldots, \frac{1}{\sigma_{N-1}^2}]$, and $\vec{z}^*$ expresses the final optimized solution in each image column.

A.1. IPM forward optimization

In order to solve the above constrained convex optimization problem Eq. (6a) with the constraint $\boldsymbol{A}\vec{z} \leq \vec{0}$, we first convert it to its Lagrangian form:

$$L(\vec{z}, \vec{\lambda}) = \frac{1}{2}(\vec{z} - \vec{\mu})^T \boldsymbol{Q}(\vec{z} - \vec{\mu}) + \vec{\lambda}^T \boldsymbol{A}\vec{z}, \tag{7}$$

Optics EXPRESS

where $\vec{\lambda} \in R^{N-1}$.

Because the primal problem in Eq. (6a) is convex, the KKT conditions are sufficient for the points to be primal and dual optimal [45]. Its corresponding perturbed KKT conditions are given as follows:

$$\text{Stationarity:} \quad \boldsymbol{Q}(\vec{z}^* - \vec{\mu}) + \boldsymbol{A}^T\vec{\lambda}^* = \vec{0} \tag{8}$$

$$\text{Perturbed complementary slackness condition:} \quad -Diag(\vec{\lambda}^*)\boldsymbol{A}\vec{z}^* = \frac{\vec{1}}{t} \tag{9}$$

$$\text{Primal feasibility:} \quad \boldsymbol{A}\vec{z}^* \le \vec{0} \tag{10}$$

$$\text{Dual feasibility:} \quad \vec{\lambda}^* \ge \vec{0}, \tag{11}$$

where t is a perturbed slackness scalar and $t>0$, and $\vec{1} \in R^{N-1}$ in Eq. (9). Bigger t means smaller dual gap between original model Eq. (6) and the Lagrangian formula (7) $L(\vec{z}, \vec{\lambda})$ [45]; and vectors $\vec{z}^*$ and $\vec{\lambda}^*$ indicate the optimal solution for the Lagrangian $L(\vec{z}, \vec{\lambda})$. Equations (8) and (9) give very important relations between $\vec{z}^*$ and $\boldsymbol{Q}$, and between $\vec{z}^*$ and $\vec{\mu}$, which can be utilized in the backward propagation of the big deep learning optimization when $\vec{z}^*$ and $\vec{\lambda}^*$ are the final optimal solutions.

A residual equation of Eq. (8) and (9) is further constructed as follows:

$$\vec{r}_t(\vec{z}, \vec{\lambda}) = \begin{bmatrix} \boldsymbol{Q}(\vec{z} - \vec{\mu}) + \boldsymbol{A}^T\vec{\lambda} \\ -Diag(\vec{\lambda})\boldsymbol{A}\vec{z} - \frac{\vec{1}}{t} \end{bmatrix} \tag{12}$$

where $\vec{r}_t(\vec{z}, \vec{\lambda}) \in R^{2N-1}$. When $\vec{r}_t(\vec{z}, \vec{\lambda}) = \vec{0}$, $\vec{z}$ and $\vec{\lambda}$ get their optimal solutions $\vec{z} = \vec{z}^*$, $\vec{\lambda} = \vec{\lambda}^*$.

Using the Newton iteration method to find the root of $\vec{r}_t(\vec{z}, \vec{\lambda}) = \vec{0}$, we can get the iterative optimization formula in the small IPM forward optimization as follows:

$$\begin{bmatrix} \vec{z} \\ \vec{\lambda} \end{bmatrix} \leftarrow \begin{bmatrix} \vec{z} \\ \vec{\lambda} \end{bmatrix} - \alpha \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{A}^T \\ -Diag(\vec{\lambda})\boldsymbol{A} & -Diag(\boldsymbol{A}\vec{z}) \end{bmatrix}^{-1} \vec{r}_t(\vec{z}, \vec{\lambda}), \tag{13}$$

where $\alpha>0$ is an iterative step length. And let

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{A}^T \\ -Diag(\vec{\lambda})\boldsymbol{A} & -Diag(\boldsymbol{A}\vec{z}) \end{bmatrix}, \tag{14}$$

where $\boldsymbol{J} \in R^{(2N-1)\times(2N-1)}$ is the Jacobian matrix of $\vec{r}_t(\vec{z}, \vec{\lambda})$ with respect to $[\vec{z}, \vec{\lambda}]$. After the IPM forward iteration ends, $\boldsymbol{J}^{-1}$ will be saved for reuse in backward propagation of big deep learning optimization, which saves expensive inverse computation of matrix of a size of $R^{(2N-1)\times(2N-1)}$.

Therefore, the IPM iterative formula (13) can be further expressed as:

$$\begin{bmatrix} \vec{z} \\ \vec{\lambda} \end{bmatrix} \leftarrow \begin{bmatrix} \vec{z} \\ \vec{\lambda} \end{bmatrix} - \alpha\boldsymbol{J}^{-1}\vec{r}_t(\vec{z}, \vec{\lambda}) = \begin{bmatrix} \vec{z} \\ \vec{\lambda} \end{bmatrix} + \alpha \begin{bmatrix} \triangle\vec{z} \\ \triangle\vec{\lambda} \end{bmatrix}, \tag{15}$$

where $\triangle\vec{z}$ and $\triangle\vec{z}$ represent the improving directions of $\vec{z}$ and $\vec{\lambda}$, and $\begin{bmatrix} \triangle\vec{z} \\ \triangle\vec{\lambda} \end{bmatrix} = -\boldsymbol{J}^{-1}\vec{r}_t(\vec{z}, \vec{\lambda})$.

Newton's iterative method guarantees a stationary solution, but it does not guarantee the feasibility of the solution [45]. Another core idea of IPM in finding an optimal solution is to start from an interior feasible point, use the Newton method to find iterative improving direction,

Optics EXPRESS

and then uses linear search to find a proper step to make sure a new iterative point is still in the feasible domain and at the same time reduces the norm of residual $\vec{r}_t(\vec{z}, \vec{\lambda})$. Therefore, in each step of the linear search process, the IPM forward algorithm needs to make sure Eqs. (10) and (11) hold. The detailed algorithm of the forward IPM iteration is illustrated in Algorithm

Algorithm 1: IPM Forward Propagation

Input : $\vec{\mu}$, $\vec{\lambda}>\vec{0}$, $\boldsymbol{Q}$, $\boldsymbol{A}$, $1>\beta_1>0$, $1>\beta_2>0$, $\beta_3>1$, $\epsilon>0$
Output : $\vec{z}^*$, $\boldsymbol{J}^{-1}$
$\vec{z} = LIS(\vec{\mu})$;
$N = length(\vec{z})$;
while *True* **do**
  $\vec{z}_0 = \vec{z}$;
  $\vec{\lambda}_0 = \vec{\lambda}$;
  $t = -\frac{\beta_3(N-1)}{(\boldsymbol{A}\vec{z})^T\vec{\lambda}}$;
  $\vec{r}_{0t}(\vec{z}, \vec{\lambda}) = \begin{bmatrix} \boldsymbol{Q}(\vec{z}-\vec{\mu})+\boldsymbol{A}^T\vec{\lambda} \\ -Diag(\vec{\lambda})\boldsymbol{A}\vec{z}-\frac{\vec{1}}{t} \end{bmatrix}$;
  $\boldsymbol{J} = \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{A}^T \\ -Diag(\vec{\lambda})\boldsymbol{A} & -Diag(\boldsymbol{A}\vec{z}) \end{bmatrix}$;
1. $\begin{bmatrix} \triangle\vec{z} \\ \triangle\vec{\lambda} \end{bmatrix} = -\boldsymbol{J}^{-1}\vec{r}_{0t}$;
  $\alpha = min(1, min(-\lambda_i/\triangle\lambda_i | \triangle\lambda_i < 0))$;
  $\vec{z} = \vec{z}_0 + \alpha\triangle\vec{z}$;
  while $\boldsymbol{A}\vec{z}>\vec{0}$ **do**
    $\alpha = \alpha\beta_1$;
    $\vec{z} = \vec{z}_0 + \alpha\triangle\vec{z}$;
  end
  while $\|\vec{r}_t(\vec{z}, \vec{\lambda})\|>(1-\beta_2\alpha)\|\vec{r}_{0t}\|$ **do**
    $\alpha = \alpha\beta_1$;
    $\vec{z} = \vec{z}_0 + \alpha\triangle\vec{z}$;
    $\vec{\lambda} = \vec{\lambda}_0 + \alpha\triangle\vec{\lambda}$;
  end
  if $\|\vec{r}_t\|<\epsilon$, **then** break;
end
return $\vec{z}$ and $\boldsymbol{J}^{-1}$

As the Newton iterative method requires an initial point near its final goal root, in Algorithm 1 a parallel *LIS* (Largest Increasing Sub-sequence algorithm) is used to find most matching initial surface locations from the initial prediction $\vec{\mu}$, and then fills the non-largest increasing sub-sequence points with its neighbor value to make the initial $\vec{z}$ feasible. As the dual gap between original cost function and Lagrangian is less than $\frac{N-1}{t}$, and $t = -(\boldsymbol{A}\vec{z})^T\vec{\lambda}$ deduced from the perturbed complementary slackness Eq. (9), this algorithm gradually enlarges t by using $t = -\frac{\beta_3(N-1)}{(\boldsymbol{A}\vec{z})^T\vec{\lambda}}, \beta_3>1$ to reduce dual gap, in order to get more accurate optimal solution to original cost function. In order to avoid $\vec{\lambda} = \vec{\lambda} + \alpha\triangle\vec{\lambda}<0$ when $\triangle\lambda_i<0$, we choose $\alpha = min(1, min(-\lambda_i/\triangle\lambda_i | \triangle\lambda_i<0))$ to make sure $\vec{1} \geq \vec{\lambda} \geq \vec{0}$.

Optics EXPRESS

A.2. IPM backward propagation

When the IPM forward optimization converges, we have $\vec{r}_t(\vec{z}^*, \vec{\lambda}^*) = \vec{0}$. Its matrix form is given as follows:

$$\vec{r}_t(\vec{z}^*, \vec{\lambda}^*) = \begin{bmatrix} \boldsymbol{Q}(\vec{z}^* - \vec{\mu}) + \boldsymbol{A}^T\vec{\lambda}^* \\ -Diag(\vec{\lambda}^*)\boldsymbol{A}\vec{z}^* - \frac{\vec{1}}{t} \end{bmatrix} = \vec{0}. \tag{16}$$

Using total differential on the above equation with respect to variables $\boldsymbol{Q}$, $\vec{z}^*$, $\vec{\lambda}^*$, and $\vec{\mu}$ gives:

$$\begin{bmatrix} \boldsymbol{Q} & \boldsymbol{A}^T \\ -Diag(\vec{\lambda}^*)\boldsymbol{A} & -Diag(\boldsymbol{A}\vec{z}^*) \end{bmatrix} \begin{bmatrix} d\vec{z}^* \\ d\vec{\lambda}^* \end{bmatrix} = \begin{bmatrix} -d\boldsymbol{Q}(\vec{z}^* - \vec{\mu}) + \boldsymbol{Q}d\vec{\mu} \\ \vec{0} \end{bmatrix}. \tag{17}$$

Using formula (14) to replace the left-most matrix above with $\boldsymbol{J}$, we have

$$\boldsymbol{J} \begin{bmatrix} d\vec{z}^* \\ d\vec{\lambda}^* \end{bmatrix} = \begin{bmatrix} -d\boldsymbol{Q}(\vec{z}^* - \vec{\mu}) + \boldsymbol{Q}d\vec{\mu} \\ \vec{0} \end{bmatrix}. \tag{18}$$

In the backward propagation of the big deep learning optimization, the backward input to the IPM optimization module is $\frac{dL}{d\vec{z}^*} \in R^N$, where L represents the loss in the deep learning network. Let us define two vectors $\vec{d}_z \in R^N$ and $\vec{d}_\lambda \in R^{N-1}$ as follows (Note: $\vec{d}_z \neq d\vec{z}^*$, and $\vec{d}_\lambda \neq d\vec{\lambda}^*$):

$$\begin{bmatrix} \vec{d}_z \\ \vec{d}_\lambda \end{bmatrix} = -\boldsymbol{J}^{-T} \begin{bmatrix} \frac{dL}{d\vec{z}^*} \\ \vec{0} \end{bmatrix}. \tag{19}$$

Transposing the above equation gives

$$\left[(\tfrac{dL}{d\vec{z}^*})^T, \vec{0}^T \right] \boldsymbol{J}^{-1} = -\left[\vec{d}_z{}^T, \vec{d}_\lambda{}^T \right]. \tag{20}$$

This Eq. (20) will be used in the following backward gradient computation to cancel $\boldsymbol{J}$ and introduce $\frac{dL}{d\vec{z}^*}$ at the same time. Now we are ready to compute $\frac{dL}{d\vec{\mu}}$ and $\frac{dL}{d\boldsymbol{Q}}$.

Compute $\frac{dL}{d\vec{\mu}}$: On the right side of full differential Eq. (18), we set $d\boldsymbol{Q} = \boldsymbol{0}$, try to derive a Jacobian matrix $\frac{d\vec{z}^*}{d\vec{\mu}}$, and then apply chain rule to compute $\frac{dL}{d\vec{\mu}}$. The detailed derivations are given as follows, where superscripts like $M \times N$ etc. represent dimensions of matrices, subscripts like i express the index of a component, I is the identity matrix, and $M = N - 1$.

Because $d\boldsymbol{Q} = \boldsymbol{0}$, we have

$$\boldsymbol{J} \begin{bmatrix} d\vec{z}^* \\ d\vec{\lambda}^* \end{bmatrix} = \begin{bmatrix} \boldsymbol{Q}d\vec{\mu} \\ \vec{0}^{M\times 1} \end{bmatrix}, \tag{21}$$

which leads to

$$\boldsymbol{J} \begin{bmatrix} \frac{d\vec{z}^*}{d\vec{\mu}} \\ \frac{d\vec{\lambda}^*}{d\vec{\mu}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{Q}^{N\times N} \\ \boldsymbol{0}^{M\times N} \end{bmatrix}. \tag{22}$$

Multiplying both sides of the equation above with Eq. (20), we get

$$\left[(\tfrac{dL}{d\vec{z}^*})^T, \vec{0}^T \right] \boldsymbol{J}^{-1}\boldsymbol{J} \begin{bmatrix} \frac{d\vec{z}^*}{d\vec{\mu}} \\ \frac{d\vec{\lambda}^*}{d\vec{\mu}} \end{bmatrix} = -\left[\vec{d}_z{}^T, \vec{d}_\lambda{}^T \right] \begin{bmatrix} \boldsymbol{Q}^{N\times N} \\ \boldsymbol{0}^{M\times N} \end{bmatrix}. \tag{23}$$

Optics EXPRESS

After matrix multiplications, we have

$$(\frac{dL}{d\vec{z}^*})^T \frac{d\vec{z}^*}{d\vec{\mu}} = -\vec{d}_z{}^T \boldsymbol{Q}, \tag{24}$$

which is equivalent to

$$(\frac{dL}{d\vec{\mu}})^T = -\vec{d}_z{}^T \boldsymbol{Q}. \tag{25}$$

Namely,

$$\frac{dL}{d\vec{\mu}} = -\boldsymbol{Q}^T \vec{d}_z. \tag{26}$$

Compute $\frac{dL}{d\boldsymbol{Q}}$: On the right side of full differential Eq. (18), we set $d\vec{\mu} = \vec{0}$, try to derive $\frac{d\vec{z}^*}{d\boldsymbol{Q}}$, and then apply chain rule to compute $\frac{dL}{d\boldsymbol{Q}}$. We first decompose $\boldsymbol{Q} \in R^{N\times N}$ into N rows that $\boldsymbol{Q} = [\vec{q}_0, \vec{q}_1, \vec{q}_2, \ldots, \vec{q}_{N-1}]^T$, where each row is a row vector $\vec{q}_i^T \in R^{1\times N}$, $i = 0, 1, 2, \ldots, N-1$. The detailed derivations are given as follows.

Because $d\vec{\mu} = \vec{0}$, we have

$$\boldsymbol{J}\begin{bmatrix} d\vec{z}^* \\ d\vec{\lambda}^* \end{bmatrix} = \begin{bmatrix} -d\boldsymbol{Q}(\vec{z}^* - \vec{\mu}) \\ \vec{0}^{M\times 1} \end{bmatrix}. \tag{27}$$

Replacing $d\boldsymbol{Q}$ with its decomposing form above gives

$$\boldsymbol{J}\begin{bmatrix} d\vec{z}^* \\ d\vec{\lambda}^* \end{bmatrix} = \begin{bmatrix} -\begin{bmatrix} d\vec{q}_0^T \\ d\vec{q}_1^T \\ d\vec{q}_2^T \\ \vdots \\ d\vec{q}_{N-1}^T \end{bmatrix}(\vec{z}^* - \vec{\mu}) \\ \vec{0}^{M\times 1} \end{bmatrix}. \tag{28}$$

In order to get partial differential with respect to $\vec{q}_i \in R^N$, we let $d\vec{q}_j = \vec{0}$ where $j \neq i$, getting

$$\boldsymbol{J}\begin{bmatrix} d\vec{z}^* \\ d\vec{\lambda}^* \end{bmatrix} = \begin{bmatrix} -\begin{bmatrix} \vec{0}^T \\ \vec{0}^T \\ \vdots \\ d\vec{q}_i^T \\ \vdots \\ \vec{0}^T \end{bmatrix}^{N\times N}(\vec{z}^* - \vec{\mu}) \\ \vec{0}^{M\times 1} \end{bmatrix}, \tag{29}$$

Optics EXPRESS

Computing the multiplication of the vector $(\vec{z}^* - \vec{\mu})$ gives

$$\boldsymbol{J}\begin{bmatrix} d\vec{z}^* \\ d\vec{\lambda}^* \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ -(\vec{z}^* - \vec{\mu})^T d\vec{q}_i \\ \vdots \\ 0 \end{bmatrix}^{N\times 1} \\ \\ \vec{0}^{M\times 1} \end{bmatrix}. \tag{30}$$

Moving the $d\vec{q}_i$ to the left side of the equation above gives

$$\boldsymbol{J}\begin{bmatrix} \frac{d\vec{z}^*}{d\vec{q}_i} \\ \frac{d\vec{\lambda}^*}{d\vec{q}_i} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} \vec{0}^T \\ \vec{0}^T \\ \vdots \\ -(\vec{z}^* - \vec{\mu})^T \\ \vdots \\ \vec{0}^T \end{bmatrix}^{N\times N} \\ \\ \boldsymbol{0}^{M\times N} \end{bmatrix}. \tag{31}$$

Multiplying both sides of the equation above with Eq. (20), we get

$$\left[(\frac{dL}{d\vec{z}^*})^T, \vec{0}^T\right]\boldsymbol{J}^{-1}\boldsymbol{J}\begin{bmatrix} \frac{d\vec{z}^*}{d\vec{q}_i} \\ \frac{d\vec{\lambda}^*}{d\vec{q}_i} \end{bmatrix} = -\left[\vec{d}_z{}^T, \vec{d}_\lambda{}^T\right]\begin{bmatrix} \begin{bmatrix} \vec{0}^T \\ \vec{0}^T \\ \vdots \\ -(\vec{z}^* - \vec{\mu})^T \\ \vdots \\ \vec{0}^T \end{bmatrix}^{N\times N} \\ \\ \boldsymbol{0}^{M\times N} \end{bmatrix}. \tag{32}$$

After matrix multiplications in both sides of the above equation, we have

$$(\frac{dL}{d\vec{z}^*})^T \frac{d\vec{z}^*}{d\vec{q}_i} = (\vec{d}_z)_i(\vec{z}^* - \vec{\mu})^T. \tag{33}$$

With chain rule of backward propagation, we have

$$\frac{dL}{d\vec{q}_i^T} = (\vec{d}_z)_i(\vec{z}^* - \vec{\mu})^T. \tag{34}$$

Optics EXPRESS

Now, combining all row vectors $\frac{dL}{d\vec{q}_i^T} \in R^{1\times N}$ of $d\boldsymbol{Q}$ gives

$$\frac{dL}{d\boldsymbol{Q}} = Diag(\vec{d}_z)\begin{bmatrix}(\vec{z}^* - \vec{\mu})^T\\(\vec{z}^* - \vec{\mu})^T\\\vdots\\(\vec{z}^* - \vec{\mu})^T\end{bmatrix}^{N\times N}. \tag{35}$$

Equations (26) and (35) are the desired backward propagation loss gradient with respect to $\boldsymbol{Q}$ and $\vec{\mu}$ in the big deep learning optimization. Therefore, the backward propagation algorithm for the IPM module in the big deep learning optimization is illustrated in Algorithm

Algorithm 2: IPM Backward Propagation Algorithm

Input : $\frac{dL}{d\vec{z}^*}, \boldsymbol{J}^{-1}, \vec{z}^*, \boldsymbol{Q}, \vec{\mu}$

Output : $\frac{dL}{d\boldsymbol{Q}}, \frac{dL}{d\vec{\mu}}$

$\begin{bmatrix}\vec{d}_z\\\vec{d}_\lambda\end{bmatrix} = -\boldsymbol{J}^{-T}\begin{bmatrix}\frac{dL}{d\vec{z}^*}\\\vec{0}\end{bmatrix}$;

2. $\frac{dL}{d\boldsymbol{Q}} = Diag(\vec{d}_z)\begin{bmatrix}(\vec{z}^* - \vec{\mu})^T\\(\vec{z}^* - \vec{\mu})^T\\\vdots\\(\vec{z}^* - \vec{\mu})^T\end{bmatrix}^{N\times N}$;

$\frac{dL}{d\vec{\mu}} = -\boldsymbol{Q}^T\vec{d}_z$

This study used PyTorch [46] 1.8 in implementing this IPM forward and backward propagation algorithm exploiting batch-parallel GPU computations on Ubuntu Linux. In our experiments, seven IPM iterations can achieve enough accuracy such that the L2 norm of residual $\|\vec{r}\|_2 \le 0.01$, with $\beta_1 = 0.5$, $\beta_2 = 0.055$, $\beta_3 = 10.0$ and $\epsilon = 0.01$ for Algorithm 1. This study used pseudo-inverse to replace normal inverse in computing $\boldsymbol{J}^{-1}$ when $\boldsymbol{J}$ is a singular matrix.

Funding. National Science Foundation (CCF-1733742, ECCS-2000425).

Disclosures. The authors declare no conflicts of interest.

Data availability. The BES OCT dataset [39,40] underlying the results presented in this paper is not publicly available at this time but may be obtained from the authors upon reasonable request. The JHU OCT MS dataset is a public dataset [43].

References

1. J. W. Shin, K. R. Sung, G. C. Lee, M. K. Durbin, and D. Cheng, "Ganglion cell–inner plexiform layer change detected by optical coherence tomography indicates progression in advanced glaucoma," Ophthalmology **124**(10), 1466–1474 (2017).
2. D. Gupta and S. Asrani, "Macular thickness analysis for glaucoma diagnosis and management," Taiwan J. Ophthalmol. **6**(1), 3–7 (2016).
3. K. Li, X. Wu, D. Z. Chen, and M. Sonka, "Optimal surface segmentation in volumetric images-a graph-theoretic approach," IEEE Trans. Pattern Anal. Mach. Intell. **28**(1), 119–134 (2006).
4. M. K. Garvin, M. D. Abràmoff, R. Kardon, S. R. Russell, X. Wu, and M. Sonka, "Intraretinal layer segmentation of macular optical coherence tomography images using optimal 3-D graph search," IEEE Trans. Med. Imaging **27**(10), 1495–1505 (2008).
5. M. K. Garvin, M. D. Abramoff, X. Wu, S. R. Russell, T. L. Burns, and M. Sonka, "Automated 3-D intraretinal layer segmentation of macular spectral-domain optical coherence tomography images," IEEE Trans. Med. Imaging **28**(9), 1436–1447 (2009).
6. Q. Song, J. Bai, M. K. Garvin, M. Sonka, J. M. Buatti, and X. Wu, "Optimal multiple surface segmentation with shape and context priors," IEEE Trans. Med. Imaging **32**(2), 376–386 (2013).

Optics EXPRESS

7. M. D. Abràmoff, X. Wu, K. Lee, and L. Tang, "Subvoxel accurate graph search using non-Euclidean graph space," PLoS One **9**(10), e107763 (2014).
8. A. Shah, M. D. Abámoff, and X. Wu, "Optimal surface segmentation with convex priors in irregularly sampled space," Med. Image Anal. **54**, 63–75 (2019).
9. P. A. Dufour, L. Ceklic, H. Abdillahi, S. Schroder, S. De Dzanet, U. Wolf-Schnurrbusch, and J. Kowal, "Graph-based multi-surface segmentation of OCT data using trained hard and soft constraints," IEEE Trans. Med. Imaging **32**(3), 531–543 (2013).
10. S. J. Chiu, X. T. Li, P. Nicholas, C. A. Toth, J. A. Izatt, and S. Farsiu, "Automatic segmentation of seven retinal layers in SDOCT images congruent with expert manual segmentation," Opt. Express **18**(18), 19413–19428 (2010).
11. Q. Yang, C. A. Reisman, Z. Wang, Y. Fukuma, M. Hangai, N. Yoshimura, A. Tomidokoro, M. Araie, A. S. Raza, D. C. Hood, and K. Chan, "Automated layer segmentation of macular OCT images using dual-scale gradient information," Opt. Express **18**(20), 21293–21307 (2010).
12. B. Keller, D. Cunefare, D. S. Grewal, T. H. Mahmoud, J. A. Izatt, and S. Farsiu, "Length-adaptive graph search for automatic segmentation of pathological features in optical coherence tomography images," J. Biomed. Opt. **21**(7), 076015 (2016).
13. J. Tian, B. Varga, G. M. Somfai, W.-H. Lee, W. E. Smiddy, and D. Cabrera DeBuc, "Real-time automatic segmentation of optical coherence tomography volume data of the macular region," PLoS One **10**(8), e0133908 (2015).
14. A. Carass, A. Lang, M. Hauser, P. A. Calabresi, H. S. Ying, and J. L. Prince, "Multiple-object geometric deformable model for segmentation of macular OCT," Biomed. Opt. Express **5**(4), 1062–1074 (2014).
15. K. Gawlik, F. Hausser, F. Paul, A. U. Brandt, and E. M. Kadas, "Active contour method for ILM segmentation in ONH volume scans in retinal OCT," Biomed. Opt. Express **9**(12), 6497–6518 (2018).
16. Y. Liu, A. Carass, Y. He, B. J. Antony, A. Filippatou, S. Saidha, S. D. Solomon, P. A. Calabresi, and J. L. Prince, "Layer boundary evolution method for macular OCT layer segmentation," Biomed. Opt. Express **10**(3), 1064–1080 (2019).
17. J. Novosel, G. Thepass, H. G. Lemij, J. F. de Boer, K. A. Vermeer, and L. J. van Vliet, "Loosely coupled level sets for simultaneous 3D retinal layer segmentation in optical coherence tomography," Med. Image Anal. **26**(1), 146–158 (2015).
18. J. Novosel, K. A. Vermeer, J. H. De Jong, Z. Wang, and L. J. Van Vliet, "Joint segmentation of retinal layers and focal lesions in 3-D OCT data of topologically disrupted retinas," IEEE Trans. Med. Imaging **36**(6), 1276–1286 (2017).
19. F. Rathke, S. Schmidt, and C. Schnörr, "Probabilistic intra-retinal layer segmentation in 3-D OCT images using global shape regularization," Med. Image Anal. **18**(5), 781–794 (2014).
20. A. Lang, A. Carass, M. Hauser, E. S. Sotirchos, P. A. Calabresi, H. S. Ying, and J. L. Prince, "Retinal layer segmentation of macular OCT images using boundary classification," Biomed. Opt. Express **4**(7), 1133–1152 (2013).
21. D. Xiang, G. Chen, F. Shi, W. Zhu, Q. Liu, S. Yuan, and X. Chen, "Automatic retinal layer segmentation of OCT images with central serous retinopathy," IEEE J. Biomed. Health Inform. **23**(1), 283–295 (2019).
22. G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. Van Der Laak, B. Van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," Med. Image Anal. **42**, 60–88 (2017).
23. D. Shen, G. Wu, and H.-I. Suk, "Deep learning in medical image analysis," Annu. Rev. Biomed. Eng. **19**(1), 221–248 (2017).
24. T. Schlegl, S. M. Waldstein, H. Bogunovic, F. Endstraßer, A. Sadeghipour, A.-M. Philip, D. Podkowinski, B. S. Gerendas, G. Langs, and U. Schmidt-Erfurth, "Fully automated detection and quantification of macular fluid in OCT using deep learning," Ophthalmology **125**(4), 549–558 (2018).
25. S. Masood, R. Fang, P. Li, H. Li, B. Sheng, A. Mathavan, X. Wang, P. Yang, Q. Wu, J. Qin, and W. Jia, "Automatic choroid layer segmentation from optical coherence tomography images using deep learning," Sci. Rep. **9**(1), 3058 (2019).
26. A. G. Roy, S. Conjeti, S. P. K. Karri, D. Sheet, A. Katouzian, C. Wachinger, and N. Navab, "Relaynet: retinal layer and fluid segmentation of macular optical coherence tomography using fully convolutional networks," Biomed. Opt. Express **8**(8), 3627–3642 (2017).
27. C. S. Lee, A. J. Tyring, N. P. Deruyter, Y. Wu, A. Rokem, and A. Y. Lee, "Deep-learning based, automated segmentation of macular edema in optical coherence tomography," Biomed. Opt. Express **8**(7), 3440–3448 (2017).
28. V. A. Dos Santos, L. Schmetterer, H. Stegmann, M. Pfister, A. Messner, G. Schmidinger, G. Garhofer, and R. M. Werkmeister, "CorneaNet: fast segmentation of cornea OCT scans of healthy and keratoconic eyes using deep learning," Biomed. Opt. Express **10**(2), 622–641 (2019).
29. L. Fang, D. Cunefare, C. Wang, R. H. Guymer, S. Li, and S. Farsiu, "Automatic segmentation of nine retinal layer boundaries in OCT images of non-exudative AMD patients using deep learning and graph search," Biomed. Opt. Express **8**(5), 2732–2744 (2017).
30. J. Kugelman, D. Alonso-Caneiro, S. A. Read, S. J. Vincent, and M. J. Collins, "Automatic segmentation of OCT retinal boundaries using recurrent neural networks and graph search," Biomed. Opt. Express **9**(11), 5759–5777 (2018).
31. M. Pekala, N. Joshi, T. A. Liu, N. M. Bressler, D. C. DeBuc, and P. Burlina, "Deep learning based retinal OCT segmentation," Comput. Biol. Med. **114**, 103445 (2019).
32. A. Shah, L. Zhou, M. D. Abrámoff, and X. Wu, "Multiple surface segmentation using convolution neural nets: application to retinal layer segmentation in OCT images," Biomed. Opt. Express **9**(9), 4509–4526 (2018).

Optics EXPRESS

33. Y. He, A. Carass, Y. Liu, B. M. Jedynak, S. D. Solomon, S. Saidha, P. A. Calabresi, and J. L. Prince, "Structured layer surface segmentation for retina OCT using fully convolutional regression networks," Med. Image Anal. **68**, 101856 (2021).
34. B. Amos and J. Z. Kolter, "Optnet: Differentiable optimization as a layer in neural networks," in *International Conference on Machine Learning*, (PMLR, 2017), pp. 136–145.
35. O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, (Springer, 2015), pp. 234–241.
36. K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*, (Springer, 2016), pp. 630–645.
37. Y. He, A. Carass, Y. Liu, B. M. Jedynak, S. D. Solomon, S. Saidha, P. A. Calabresi, and J. L. Prince, "Fully convolutional boundary regression for retina OCT segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, (Springer, 2019), pp. 120–128.
38. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, (2017), pp. 5998–6008.
39. J. Zhao, Y. X. Wang, Q. Zhang, W. B. Wei, L. Xu, and J. B. Jonas, "Macular choroidal small-vessel layer, sattler's layer and haller's layer thicknesses: The beijing eye study," Sci. Rep. **8**(1), 4411 (2018).
40. Y. X. Wang, H. Yang, H. Luo, S. W. Hong, S. K. Gardiner, J. W. Jeoung, C. Hardin, G. P. Sharpe, K. Nouri-Mahdavi, J. Caprioli, S. Demirel, C. A. Girkin, J. M. Liebmann, C. Y. Mardin, H. A. Quigley, A. F. Scheuerle, B. Fortune, B. C. Chauhan, and C. F. Burgoyne, "Peripapillary scleral bowing increases with age and is inversely associated with peripapillary choroidal thickness in healthy eyes," Am. J. Ophthalmol. **217**, 91–103 (2020).
41. K. Lee, M. D. Abramoff, M. Garvin, M. Sonka, and X. Wu, "Iowa reference algorithm – OCTExplorer," (2020), https://www.iibi.uiowa.edu/oct-reference.
42. M. D. Abràmoff, M. K. Garvin, and M. Sonka, "Retinal imaging and image analysis," IEEE Rev. Biomed. Eng. **3**, 169–208 (2010).
43. Y. He, A. Carass, S. D. Solomon, S. Saidha, P. A. Calabresi, and J. L. Prince, "Retinal layer parcellation of optical coherence tomography images: Data resource for multiple sclerosis and healthy controls," Data brief **22**, 601–604 (2019).
44. Y. He, A. Carass, B. M. Jedynak, S. D. Solomon, S. Saidha, P. A. Calabresi, and J. L. Prince, "Topology guaranteed segmentation of the human retina from OCT using convolutional neural networks," arXiv preprint arXiv:1803.05120 (2018).
45. S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization* (Cambridge University, 2004).
46. A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems* 32, 8026–8037 (2019).