Fall 2013

Cuda K-Nn: application to the segmentation of the retinal vasculature within SD-OCT volumes of mice

Wenxiang Deng
University of Iowa

CUDA K-NN: APPLICATION TO THE SEGMENTATION OF THE RETINAL VASCULATURE WITHIN SD-OCT VOLUMES OF MICE

by

Wenxiang Deng

A thesis submitted in partial fulfillment of the
requirements for the Master of Science degree
in Biomedical Engineering
in the Graduate College of
The University of Iowa

December 2013

Thesis Supervisor: Assistant Professor Mona K. Garvin

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

MASTER'S THESIS

This is to certify that the Master's thesis of

Wenxiang Deng

has been approved by the Examining Committee for the thesis requirement for the Master of Science degree in Biomedical Engineering at the December 2013 graduation.

Thesis Committee: _____
Mona K. Garvin, Thesis Supervisor

_____
David G. Wilder

_____
Michael D. Abràmoff

_____
Joseph M. Reinhardt

_____
Todd E. Scheetz

ACKNOWLEDGEMENTS

First, I would first like to thank my thesis advisor, Dr. Mona Garvin for her guidance and support for me in this work. I would also like to thank Dr. Michael D. Abràmoff, Dr. Joseph Reinhardt, Dr. Todd E. Scheetz, and Dr. David Wilder for being on my committee and for their helpful feedback.

This work was supported by a grant from the Veterans Administration.

TABLE OF CONTENTS

LIST OF TABLES

Table

LIST OF FIGURES

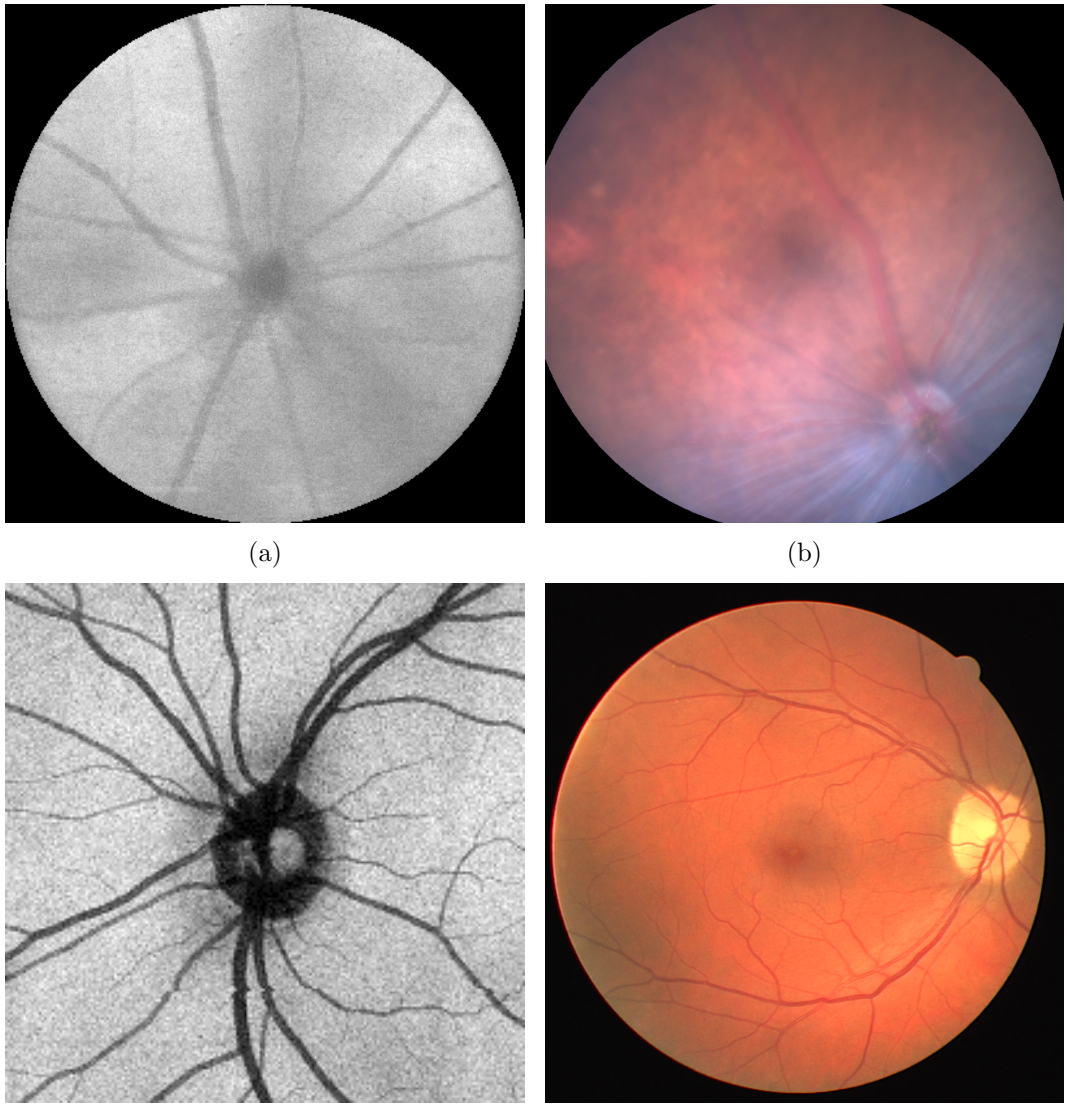
Figure

CHAPTER 1
INTRODUCTION

Blood vessels in the retina supply retinal tissues with nutrition and oxygen and can change their appearance in diseases such as diabetic retinopathy. The automated segmentation of retinal vessels using ophthalmic imaging modalities, such as color fundus photography and optical coherence tomography (OCT), is important not only for the measurement of vasculature properties (such as vessel widths and tortuosity), but also to enable image intrasubject image registration (e.g., fundus-to-fundus, OCT-to-OCT, fundus-to-OCT, etc.). With human data, the automated segmentation of retinal blood vessels in fundus and OCT images is a well understood problem and has many published papers [1–12]. Amongst these, some approaches adopted optical coherence tomography (OCT) images for vessel segmentations [9–12]. OCT is a widely adopted noninvasive modality for retinal imaging introduced in 1991 [13]. Newly available spectral-domain (SD-OCT) scanners can further provide 3D structure of retina and present detailed 3D images of the eye. In SD-OCT images, blood vessels are not distinguishable, since the vessels absorb the wavelengths used in the scanner. There are silhouettes below vessels caused by the absorption. As proposed by Webhe *et al.* [14], such shadows can be used for detection of the vessels in 2D projection images of OCT volumes.

For further understanding of the eye diseases, retinal vasculature studies of mice can be useful because of their anatomical similarity to humans and relatively inexpensive experimental cost. Also, rapid reproduction rate and short lifespan of mice could make longitudinal studies easier. Fig. 1.1(b) and Fig. 1.1(d) are examples of 2D fundus images of a mouse and human.

SD-OCT imaging is also used in mouse retinal studies [15–17] apart from human. Fig. 1.2 shows an example of mouse SD-OCT volumes. However, for mouse SD-OCT images, there has been no approach for vasculature segmentation.



(a) (b)

(c) (d)

Figure 1.1: Comparison of mouse and human retinal images. (a) and (c) are example projection images in a mouse (a) and human (c) retinal SD-OCT volumes. (b) and (d) are examples of fundus image of mouse (b) and human (d).

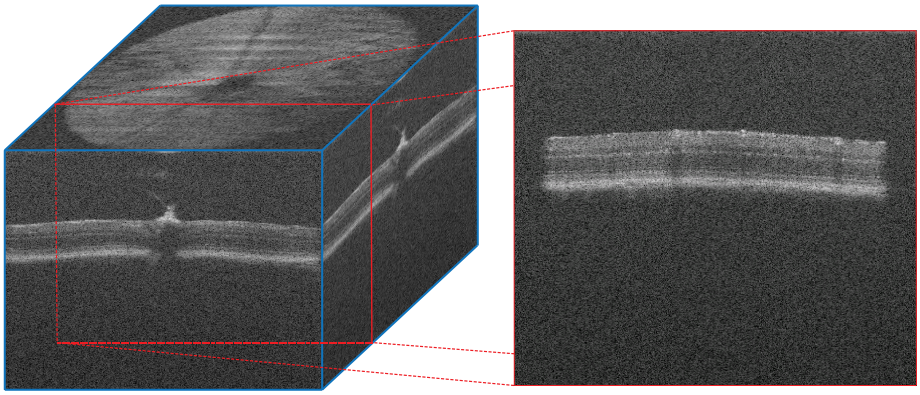


Figure 1.2: A typical mouse OCT volume and an example b-scan slice of the image.

Segmenting vessels in mouse SD-OCT data need different approaches than in the human case, because of the differences between mice and human retinal images. More specifically, differences can be seen in example projection images created from near the retinal pigment epithelium (RPE) layers in SD-OCT volumes, shown in Fig. 1.1(c) and Fig. 1.1(a). Fig. 1.1(c) is an example projection image obtained from human retinal OCT volume. It has a high contrast between vessels and background, which makes segmentation of vessels easier. Fig. 1.1(a) shows a projection of mouse OCT image. This is a different scenario. The neural canal opening (NCO) boundary is small and there is less interaction between the NCO and vessels. Also, vessels in the image are vaguely identifiable. In order to segment vessels under highly successful pixel classification methods [18], a better set of features images are needed. In this thesis, we apply a widely used k-nearest-neighbor (k-NN) approach for segmentation of mouse retinal vessels. In order to process more features using k-NN approach with less computation time, we first explore two fast k-NN implementations for ophthalmic image data.

k-NN is a simple and widely used machine learning method [19]. It uses a labeled reference set and assigns each query point the most common label of its k closest neighbors. Fig. 1.3 shows an example of k-NN search, with the number of neighbors

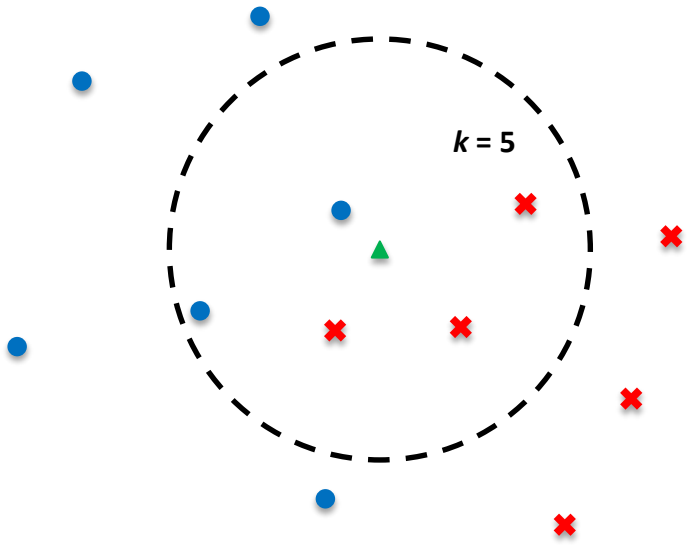


Figure 1.3: Example of k-NN search where $k = 5$. Blue circles and red crosses are reference points of two classes. The green triangle is a query point. In this case, it should be assigned to class red.

$k = 5$. Although the approach itself is sensitive to noise, with better selection of features, k-NN would generate satisfying results. Therefore, k-NN has been widely used in medical image processing [9, 20–27].

Behind its simplicity, however, using brute force (BF) search for k-NN approach is time consuming. As an instance-based learning approach, each test of finding k nearest neighbor would take $O(kdn)$, where n is the number of points in training set, d is the dimension of the dataset, and k is the number of neighbors to search. Approaches including preprocessing the training data into data structures such as tree structure [28, 29] or using a graphic processing unit (GPU) to parallelize the algorithm [30] have been proposed to reduce computation time.

There are two purposes of this thesis. We first explore the speed performance of a GPU-based parallel k-NN approach [30] against the commonly used ANN library [29] for ophthalmic image data. Three datasets are used in the test, including data from 2D and 3D images. We test the running time of each data under different settings

(i.e, number of reference data points, number of features used, different ks). For the second aim, we establish three approaches built on k-NN to solve vessel segmentation problem in mouse retinal data. Our work adopts ideas from the pixel classification method for human SD-OCT data by Niemeijer *et al.* [9], partially because of its high reported accuracy of 0.97 area under Receiver Operating Characteristic (ROC) curve (AUC), but with modifications to specifically deal with the challenges of SD-OCT data of mice.

The rest of the thesis is organized as follows. In Chapter 2, we describe prior work done to enhance the running speed of k-NN approach. We also review methods proposed to segment human retinal vessels in fundus and OCT images. Chapter 3 includes our work to test GPU-based k-NN [30] in ophthalmic images against the ANN implementation [29]. In Chapter 4, we describe our novel approaches for mouse vessel segmentation in SD-OCT images. And finally conclusions and future work are provided in Chapter 5.

CHAPTER 2
BACKGROUND AND PRIOR WORK

2.1 k-NN for medical image analysis

Machine learning is an important part in many medical image analysis approaches for segmentation and computer-aid diagnosis. As a non-parametric method in machine learning, k-NN requires no prior knowledge of data characteristic structure and is easy to configure. It's been widely applied to various types of images for image analysis. For example, de Bruijne *et al.* [21] presented a general approach in 3D image segmentation using k-NN to improve grey level appearance models in Active Shape Models (ASM) for segmentation of tubular structure in medical data. In their work, k-NN gave a probability profile for aneurysm boundaries. However, it was also mentioned in the work that a k-NN based approach would take much more time than the baseline approach. For MR images, k-NN algorithm can be used for segmenting lesion [22] or brain tumors [23]. It has also been used in CT images for detection of pathological tissue [31, 32]. In terms of microscopic images, k-NN is often used for automatic segmentation of cells and tissues. For instance, [20] used a k-NN classifier with wavelets and densitometric features to distinguish different classes of breast tissue nuclei. And in [24], it was used for identifying cell phase.

Of particular relevance to this thesis, k-NN is widely used to process fundus and OCT images in ophthalmology [6, 9, 10, 12]. Use of k-NN for vessel segmentation is discussed further as part of Section 2.4 (which serves as a foundation for Chapter 4). Three additional ophthalmic applications that use k-NN are highlighted in the experiments of Chapter 3. In particular, in the work by Miri *et al.* [26], after abstracting features from fundus and 2D projection of SD-OCT images, k-NN was used for classifying neuroretinal cup and rim structure. Antony *et al.* used k-NN for classifying texture features from SD-OCT data [25], in order to generate cost functions for layer segmentations. In the work by Qiao *et al.* [27], 2D fundus images were used to

identify bifurcation points in blood vessel trees in the images. Principle component analysis (PCA) based features were extracted in fundus images. A k-NN classifier was then used to determine vessel bifurcations.

2.2 k-NN optimization approaches

Slow execution time is a major drawback of instance-based methods such as k-NN. One main reason for time consumption is that it compares a query point with all training data points. There have been efforts to minimize the running time. Here we review three types of approaches to reduce the running time.

2.2.1 Instance reduction

Since the nearest-neighbor algorithm stores all reference instances, this not only uses large memory, but also causes speed issues. One category of methods is to remove some reference instances in the algorithm. These approaches mostly use a subset of the original training set, and aim to increase noise tolerance of k-NN algorithm, as well as reduce running time. Most of them primarily focused on $k = 1$, and could be modified for $k > 1$. A collection of such approaches are reviewed by Wilson *et al.* [33].

2.2.2 Preprocessing data structure and approximation

Other techniques like k-d trees [28, 34], quadtrees [35], or box-decomposition trees [29] use data structures to preprocess and store the training set. This would decrease the number of instances that would need to be compared with each query point. For instance, k-d tree approach by Friedman *et al.* [34] preprocesses data into a data structure where queries can run nearest neighbor searches within sublinear (logarithmic) time. In practice, such approaches have very good performance in low dimensions, but running time would grow rapidly with increasing dimension d.

On the other hand, approximate nearest neighbor (ANN) approach by Arya *et al.*

[29] managed to achieve efficient performance query time with some loss of precision. For constant $\epsilon > 0$, each of the nearest neighbors from ANN approach would have a smaller than $(1+\epsilon)$ difference to the precise distance. A balanced box-decomposition (BBD) tree was also introduced as a variation of k-d tree to store the reference instances. The algorithm first preprocesses training data and builds them into a tree data structure. For the neighbor search, given a query point q, the leaf cell that contains the point is located. Then a priority search is conducted by computing the distance between the q and nearby cells and reference points in the cells. The process terminates when the distance $dist(q, nextcell) > dist(q,p)/(1+\epsilon)$, where p is the current kth nearest point and *nextcell* is the next nearest cell to be examined.

As one of the fastest k-NN approach of its kind, ANN also has an implemented library [29]. Therefore, we use it as one approach for time comparison in Chapter 3.

2.2.3 CUDA-based approaches

However, for the high-dimensional nearest neighbor problem, approaches involving preprocessed data structures may still require too much processing time [29]. Approaches using GPU framework can be used for the problem. As a relatively new platform for computation, general purpose computation of GPU (GPGPU) can parallelize BF k-NN and greatly increase the speed performance of k-NN approach. Under CUDA (Compute Unified Device Architecture) framework by NVIDIA, there are implementations of BF k-NN approaches developed on GPU [30, 36, 37].

Fig. 2.1 shows a simple CUDA programming model. This is a heterogeneous process. Within the model, a CPU-based program accesses the GPU as a device via kernel functions. When execution finishes, the device callbacks to CPU part of the program. Also, in the device, there are dual level of parallelization, block and thread. With a highly parallelized hardware and dual level of parallelization in CUDA, up to two orders of magnitude of speed up compared with ANN implementation [29] for synthesized data was reported by Garcia *et al.* [30].

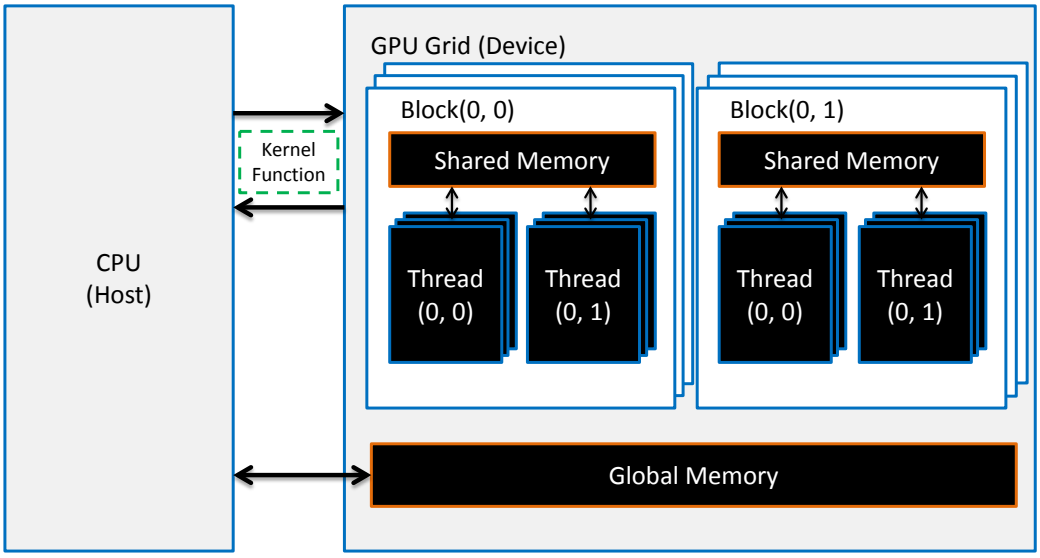


Figure 2.1: A simple programming model of CUDA framework. CPU host dispatches kernel task to GPU device, and dual level structure of blocks and threads can be used for better performance.

In this work, we adopt this CUDA-based approach by Garcia *et al.* [30] as the other candidate for speed comparison.

2.3 Retinal layers

The retina inside the eye is a structure of multiple layers. It converts light into electrical neural impulse for generating vision. Photoreceptor cells, including rods and cones, respond to light and initiate series of chemical reactions to trigger neural impulse. Generated impulse are then transmitted to brain to create vision. There are two main regions in retina, macula and optic nerve head (ONH). The macular region is responsible for sharp central vision. The ONH is not light sensitive and it is where optic nerve and blood vessels leave the eye.

Fig. 2.2 shows a typical mouse ONH SD-OCT b-scan slice and the corresponding layers in a mouse retina. Blue circles represents shadows projected by retinal vessels. The distinguishable surfaces and layers are briefly illustrated as follows [38, 39]:

- Inner limiting membrane (ILM) - is the basement membrane as boundary of vitreous and retina.
- Nerve fiber layer (NFL) - contains axons of ganglion cells.
- Ganglion cell layer (GCL) - contains nuclei of ganglion cells.
- Inner plexiform layer (IPL) - contains synapse between bipolar cell axons and dendrites of ganglion cells.
- Inner nuclear layer (INL) - contains nuclei and cells bodies of bipolar, horizontal, Müller, and interplexiform cells.
- Outer plexiform layer (OPL) - contains synapse between rod and cone cells and dendrites of horizontal and bipolar cells.
- Outer nuclear layer (ONL) - contains cells bodies of rod and cone cells (photoreceptors).
- External limiting membrane (ELM) - is a layer that separates between photoreceptor cells and also photoreceptor with Müller cells.
- The outer segments (OS) and inner segments (IS) of photoreceptors - contains rod and cone cells.
- Retinal pigment epithelium (RPE) - is a single layer of cuboidal cells between choroid and retina.
- Bruch's membrane - is the anterior surface of the choroid.

As for mouse retinal image shown in Fig. 2.2, unlike human SD-OCT images, nerve fiber layer (NFL) is not distinguishable from the ganglion cell layer (GCL) and Inner Plexiform Layer (IPL). Therefore, we refer to these layers as the NF+GC+IPL.

2.4 Retinal vessel segmentation

In animal retinal studies, no current approach for mouse vessel segmentation has been proposed. For human case, on the other hand, there are a lot of approaches available for segmenting human retinal vessels. Most of them are for fundus images,

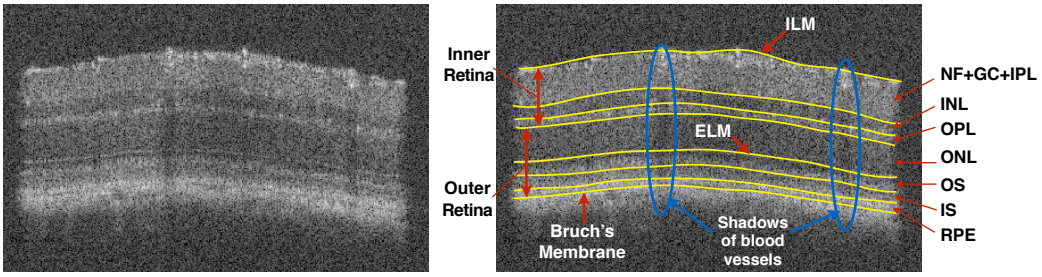


Figure 2.2: Example of a b-scan slice and its corresponding layers in mouse SD-OCT volume.

and many of them can extend naturally to OCT projection images in human retinal studies. Within current approaches, a most successful category is by using pixel classification. Pixel classification is an important machine learning category which uses features for each pixel to decide classes of the pixel in image.

For fundus images, Niemeijer *et al.* [6] compared a number of approaches on a publicly available DRIVE (Digital Retinal Images for Vessel Extraction) datasets. A supervised pixel classification method using k-NN classifier was also proposed. In the method, 31-dimensional multifeature vectors were generated using Gaussian-based filter banks of up to 2^{nd} order derivative and 5 scales of σ, as well as a green channel intensity image. Soares *et al.* [7] developed a classification method using Gabor features from different scales. Bayesian classifier with Gaussian mixture model (GMM) was applied then as pixel classifier. Another supervised classification method was proposed by Staal *et al.* [5]. By assuming vessels as elongated structures, ridge-based features were obtained and the most distinguishing features were selected using feature selection. A k-NN classifier was then applied for the segmentation. Moreover, supervised classifier such as neural network [40, 41], support vector machine [42], or AdaBoost-based methods [43] are applied as well. Other than supervised classifiers listed above, unsupervised classifiers are also used for the problem. In [44] and [45], unsupervised classifier c-Means clustering were applied. [44] used texture features

extracted by Gabor features for fuzzy c-means clustering. In [45], spatially weighted fuzzy c-means clustering (SWFCM) was applied after matched filters for the images.

For SD-OCT volumes, there is less published work. Here we focus on approaches to segment vessels in OCT projection images. A k-NN based approach was brought up in [9,10], which used segmented retinal layers in SD-OCT for 2D projection image for vessel classification. This work used a projection image near the RPE layer. Similar step as [6] was then applied for pixel classifications. This method was modified by Hu *et al.* [12] for a better vessel segmentation near the neural canal opening (NCO). This was done by incorporating presegmented NCO information in the pixel classification. Another approach using SD-OCT image was by Xu *et al.* [11]. LogitBoost boosting algorithm was applied on 2D Gaussian filtered features in OCT projection images and Haar features on A-scans of 3D images.

Other than pixel classification, methods such as region growing [1], mathematical morphology [2], or local thresholding [4] are also used for vessel segmentation. Martínez-Pérez *et al.* [1] used scale-space and region growing for segmentation of vasculature. In the approach, two features were generated using gradient magnitude and ridge strength in the image. Histograms of them were then used for region growing. In the approach by Zana *et al.* [2], general mathematical morphology was applied to detect vessel like objects in fundus images. Another morphological method by Lam and Yang [3] applies divergence of vector fields for segmentation of retinal images with leisures. Centerlines are obtained by morphological operations, and vessels and non-vessels are decided. Adaptive thresholding was used by X. Jiang *et al.* [4]. A series of individual thresholds were applied and skelotonization was used for pruning the results.

CHAPTER 3
COMPARISON OF CUDA K-NN WITH ANN LIBRARY FOR MEDICAL IMAGING

3.1 Motivation

CUDA framework is a heterogeneous programming model. Host CPU side of the program calls GPU as a device, and gets a call back when finished. Although CUDA-based k-NN implementation [30] has been tested with the ANN library [29], it was mainly on synthetic data. There is need to explore them for more medical data, in order to better understand the speed performance of CUDA k-NN. Here we test these two methods on three sets of ophthalmic image data. The first one is fundus image data for bifurcation identification. The second dataset is from SD-OCT projection image and fundus image for cup and rim features. And the last dataset is SD-OCT data for segmenting retinal layers.

3.2 CUDA k-NN approach

In this section, we briefly review the CUDA-based k-NN approach by Garcia *et al.* [30].

For brute force k-NN algorithm, let $P = \{p_1, p_2, ..., p_m\}$ be the training set and $Q = \{q_1, q_2, ..., q_n\}$ be the testing set. Within the sets, elements are $p_i = (p_{i1}, p_{i2}, ..., p_{id})$ and $q_j = (q_{j1}, q_{j2}, ..., q_{jd})$, where d is the dimension of features and p_{ix} or q_{jy} denotes a single feature. Each query point $q_i \in Q$ goes through the following steps to find k nearest neighbors:

- For each $q_i \in Q$, compute its distance to every $p_j \in P$. The distances can be measured in Euclidean metric, Manhattan metric, or other metrics.
- Sort distances computed above.
- Output reference points of k lowest distances.

Although brute force k-NN in CPU-based program is not time efficient, it can be easily parallelizable in CUDA framework. The implementation of BF k-NN in CUDA

framework consists of two kernel functions. The first is to compute the distances of all query points and reference points. This is fully parallelizable. Every thread computes the distance of one query point q_j and one reference point p_i. A distance matrix of size $m \times n$ is generated containing distances between m reference points and n query points.

For the second part, partial sorting is applied to sort the distance matrix. In their work, a parallel version of insert sort and comb sort were compared. Insert sort was used in the work because of its faster performance for smaller ks. In more details, the insert sort are parallelized such that each thread sorts all distances for one query point. After sorting is finished, the top k elements and the corresponding distances are selected as k nearest neighbors.

3.3 Experimental methods

To test the speed performance of CUDA k-NN implementation [30], we compare it to a widely used fast k-NN implementation: ANN library in C++ [29]. Random data was well tested in [30], so we only test them on ophthalmic image data. Also, since speed performance under different data dimensions is a most important aspect, our tests lean on the speed comparison of the two implementations with different dimensions of data.

There are three datasets with feature vectors and their corresponding truths used for validating the CUDA-based k-NN method against ANN library. The datasets we use are illustrated as follows:

- Principle component analysis (PCA) based features abstracted from green and red channel in fundus images to determine vessel bifurcations [27].
- 2D features in retinal projection images for classifying neuroretinal cup and rim using fundus and SD-OCT images [26].
- Most descriptive 2D and 3D Gabor-based features for classifying the cost func-

tion for surface segmentations in SD-OCT volumes [25].

All data are normalized with zero mean and unit variance in each feature they contain. A fixed size of 10,000 testing data points is used for all tests since both implementations have a linear increase of time for increasing number of testing data points. Training data and testing data are all randomized then with regard to arrangement of different features as well as data points.

For each dataset, we first test CUDA [30] and ANN implementation [29] in number of training data points, dimension of data (i.e., number of features), and measure the running speed. For the dimensionality of data, ranking of features are first randomized to make sure adjacent features are not similar. Subset of features are tested from $5dim$ to the maximum dimension of given dataset, with $10dim$ interval. The number k is fixed as 21. A subset of training data from 1,000 data points up to 250,000 are used in the test.

Then a comparison of ks is also tested in one of the datasets to find correlation between choice of k, dimension of data, and running time. For convenience, the number of training data points is fixed to 200,000 and $k = 21, 41, 61, 81$ are compared.

To perform the tests, we use a desktop computer with Intel i7-2600k processor, 16GB memory, and NVIDIA GeForce GTX 570 GPU. Both implementations are compiled in Visual Studio 2010 in 64bit Microsoft Windows 7.

3.4 Results

Fig. 3.1 and Table 3.1 shows the speed performance of each approach. In Fig. 3.1, it shows the performance of CUDA and ANN implementation for all three datasets. Fig. 3.1(a), Fig. 3.1(d), and Fig. 3.1(g) are illustrations of ANN library's speed performance. Fig. 3.1(b), Fig. 3.1(e), and Fig. 3.1(h) show the performance of CUDA k-NN implementation. Then, a speed up comparison of the two are shown in Fig. 3.1(c), Fig. 3.1(f), and Fig. 3.1(i). In particular, speed up is defined as the running time

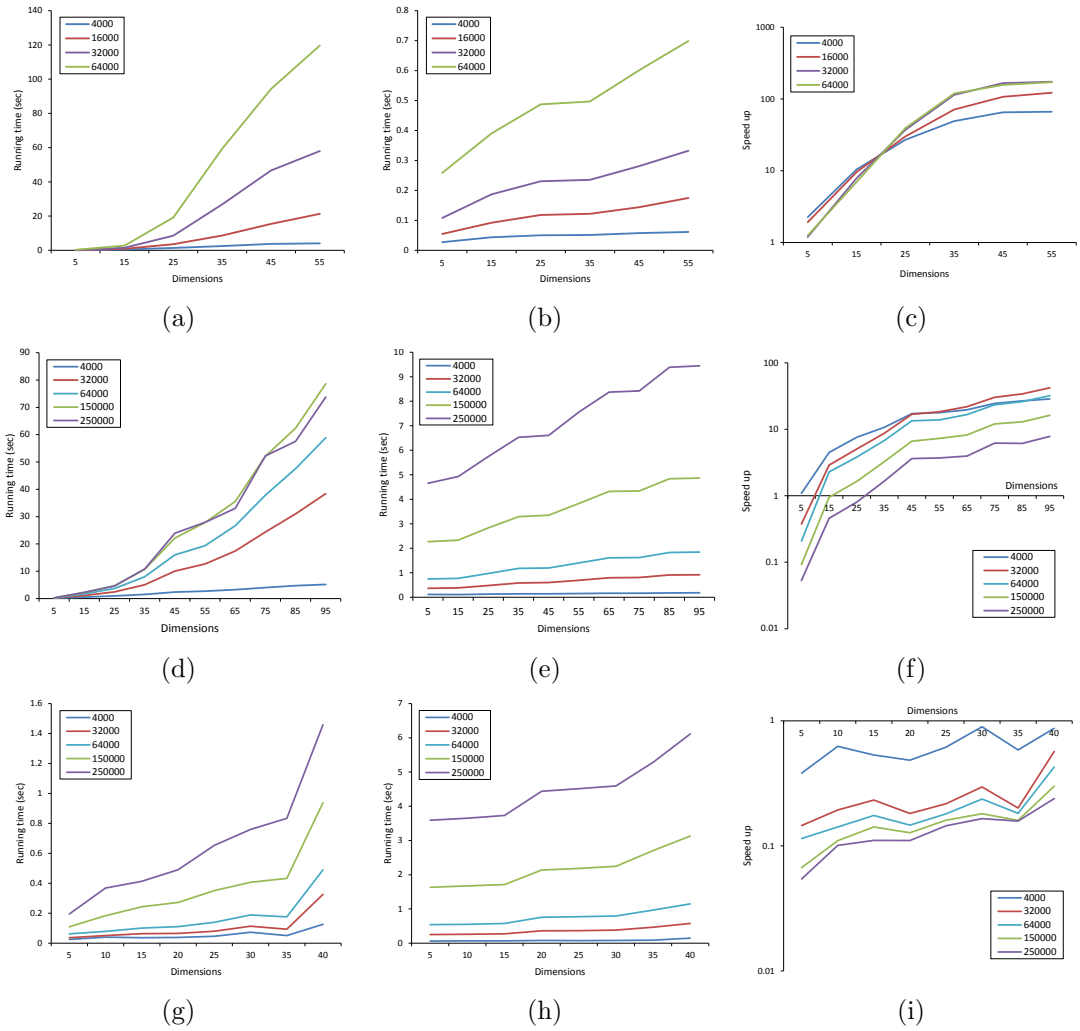


Figure 3.1: Speed performance of ANN and CUDA k-NN implementation tested on different data size and dimensions. In the figures, horizontal coordinates are different dimensional features used for test, and vertical coordinates are running time or speed different ratio in (c), (f), and (i). Different colors of plot lines stand for different size of sampled training data points. (a), (b), and (c) are ANN, CUDA k-NN and speed comparison using bifurcation data. (d), (e), and (f) are the corresponding for cup & rim data. (g), (h), and (i) are for surface segmentation data.

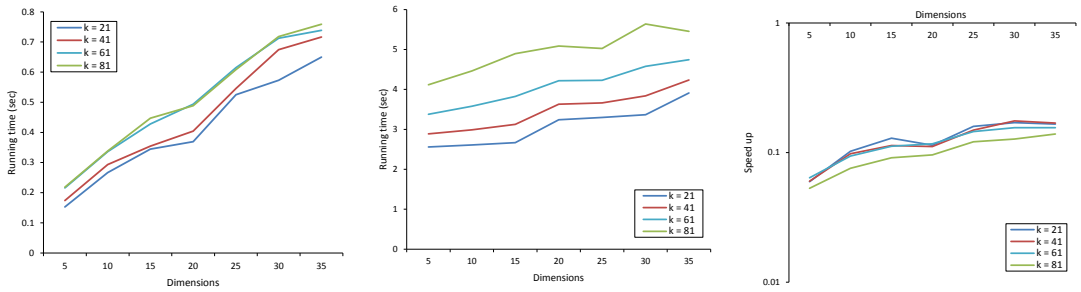
Table 3.1: Speed up from ANN to CUDA k-NN implement where $k = 21$. †

Dimension	Tested data	$n = 4000$	$n = 8000$	$n = 16000$	$n = 32000$	$n = 64000$
$dim = 5$	Bifurcation	2.54	1.94	2.14	1.4	1.13
	Cup & rim	1.08	0.79	0.62	0.38	0.21
	OCT Surface	0.38	0.27	0.20	0.15	0.11
$dim = 15$	Bifurcation	10.46	8.24	9.57	7.87	6.97
	Cup & rim	4.48	4.42	3.79	2.88	2.28
	OCT Surface	0.53	0.41	0.30	0.23	0.17
$dim = 25$	Bifurcation	26.94	33.29	30.10	37.16	39.26
	Cup & rim	7.58	7.17	6.26	5.03	3.81
	OCT Surface	0.61	0.40	0.33	0.21	0.18
$dim = 35$	Bifurcation	49.07	55.25	71.17	113.98	119.27
	Cup & rim	10.70	10.41	9.32	8.69	6.77
	OCT Surface	0.59	0.38	0.33	0.20	0.18

† Speed up is the comparison of running time between CUDA and ANN implementations. It is defined as ratio of $\frac{Time_{ANN}}{Time_{CUDA}}$.

of the ANN approach over that of the CUDA-based approach (i.e., $\frac{Time_{ANN}}{Time_{CUDA}}$). One noticeable result from Table 3.1 is that the three datasets perform differently. Under similar settings ($k = 21$, number of query points fixed to 10000, and number of reference points from 1000 to a maximum of 250,000), CUDA k-NN implementation performs poorly in OCT surface segmentation data, and performs best in bifurcation data. It is easy to see that with increasing dimensions, running time of CUDA approach tends to increase slower than that of ANN. Overall, Fig. 3.1 also shows that the speed up increases with the increasing of dimension, but decreases with more data points.

In the bifurcation data, we can see that CUDA k-NN has the most speed up. It has up to two orders of magnitude of speed up compared to ANN implementation. For cup & rim data, CUDA implementation has a worse speed performance when the dimension of data is low, but the speed up increases with higher dimensions. Meanwhile, it also shows that speed up of CUDA implementation gets smaller when the number of training data points gets bigger.



(a) (b) (c)

Figure 3.2: Speed of ANN and CUDA k-NN implementation on surface segmentation data. (a), (b), and (c) are ANN, CUDA k-NN and speed up.

In OCT surface segmentation data, however, CUDA k-NN doesn't show any speed up compared to ANN. Actually, ANN implementation is up to 10 times faster than CUDA implementation. On the other hand, there is a clear pattern that with the increase of dimensions CUDA-based approach tends to get relatively faster. We notice that the main reason for this is that while the running time in CUDA approach remains similar, ANN implementation runs much faster in surface segmentation data than it in the other two datasets. More exploration of this is proposed in discussion section.

For comparison under different ks, Fig. 3.2 shows that the line patterns for different ks are similar. Also, the overall running time for CUDA k-NN increases faster than ANN implementation for increase k. Therefore, CUDA k-NN tends to have lower speed up ratio against ANN for larger ks.

3.5 Discussion

Tests on different ophthalmic image data show that CUDA-based k-NN approach as up to 2 orders of magnitude speed increase comparing with ANN implementation. This is similar to random data tested in [30]. We also notice that for different image data, the speed increase may vary from 2 to -1 orders of magnitude.

Additionally, running time of CUDA k-NN approach increases rapidly with in-

Table 3.2: Number of dimensions to explain 99% of variance in each testing dataset

	Surface data	Cup & rim data	Bifurcation data
Dimensions	1 out of 39	41 out of 96	55 out of 60

creasing size of reference set. Therefore, a resampling process is needed for a larger reference set. For same reference point size, increasing dimensions only has a small impact on running time. This gives CUDA k-NN more advantage over ANN implementation for high dimension data.

As can be seen in Fig. 3.1(i), in some cases ANN could have an up to 10 times better performance. We assume the cause for this is that after query points are located in leaf nodes, the distribution of reference points could cause less cells to be visited in the search process. To explore this further, we do a principle component analysis (PCA), and find the number of dimensions needed to explain 99% of variance. The result is shown in Table 3.2. This confirms our assumption. It seems the performance of ANN library is highly dependent on the distribution of data, even if the given data is in relatively high dimensions. More future explorations can be done to verify the assumption.

One main drawback of CUDA-based k-NN technique is that it has large memory requirements. In our test, a number of approximately 300,000 reference points would exhaust GPU on board memory. With limited GPU memory, instances in the training set need to be resampled before copying to GPU memory.

In this work, comparison of training time (i.e. time for preprocessing reference set) and testing time (i.e. time for finding k closest neighbors) in ANN implementation are not tested specifically. Also, comparison tests under different ks only involve one dataset. For future work, more tests on more datasets can be performed.

Potential improvements can also be made in CUDA k-NN approach for better

performance. For instance, processes such as pruning the training set [33, 46] can be adopted. This could not only improve the running speed, but also make k-NN more robust to noise. Other than this, since the partial sorting used for finding the kth smallest element is not necessarily the fastest way, there are other selection algorithm can be used to optimize the speed. We have tried quickselect algorithm, but since it doesn't work very well in parallel computation, other faster select algorithm such as radix select can be used to improve the speed.

CHAPTER 4
MOUSE RETINAL VESSEL SEGMENTATION

4.1 Motivation

As a widely used subject for longitudinal studies, mouse models can help with the understanding and analysis of retinal diseases. Since there is no existing work proposed on segmenting blood vessels in SD-OCT data of mice, we propose three approaches for segmenting the mouse vasculature using a CUDA-based k-NN classifier. The basic idea is to extract features from projection images of SD-OCT volumes and use a classifier to classify each pixel in the projection images as "vessel" or "non-vessel". The first approach uses similar approach by Niemeijer *et al.* [9] that was proposed for human SD-OCT images. On the basis of the first approach, the second one uses more features and a feature selection process for classification. The last one extracts features from projection images of all segmented retinal layers, instead of single projection image as in the first two. For convenience, we call these methods by baseline, single projection, and all layers approaches.

4.2 Methods

An overview of the approaches we propose is illustrated in Fig. 4.1. It consists of four steps. First, a total of 8 retinal surfaces are segmented using the graph-theoretic approach proposed by Antony *et al.* [17] (Section 4.2.1). Second, 2D projection images are created using intraretinal layers from the segmented surfaces. This is needed because vessels cannot be readily seen in SD-OCT images, but absorption of light in blood vessels creates silhouettes in the retinal layers below vessels. Such silhouettes would represent vessel locations in a projection image. Each of three approaches adopts a different combination of retinal layers (Section 4.2.2). Third, a collection of features are generated from projection images using Gaussian, Hessian, and Gabor-based filters. Each approach uses a different selection of features (Section 4.2.3). Finally, a greedy feature forward selection is performed to get better feature sets for

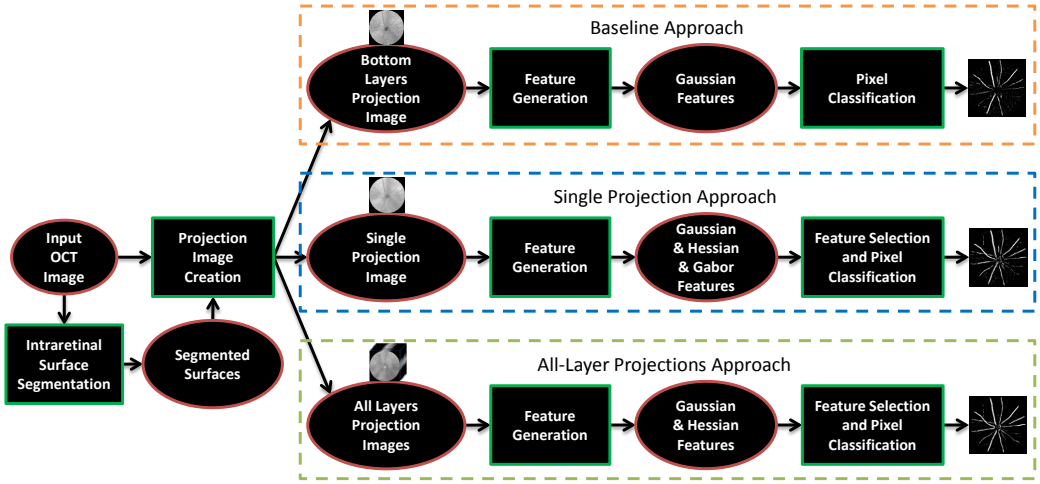


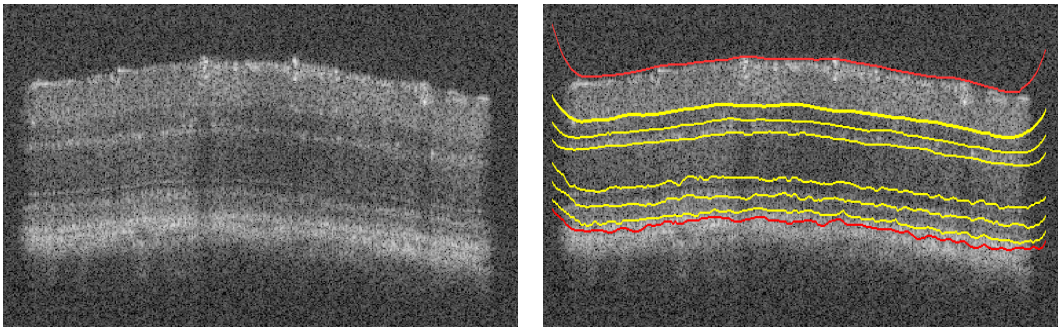
Figure 4.1: Schematic overview of the methods proposed. It includes layer segmentation using the graph-theoretic method and three approaches for pixel classification to find blood vessels.

single projection and all layers approaches. Pixel classification of CUDA-based k-NN is used to create a blood vessel probability map (Section 4.2.4).

4.2.1 Intraretinal layer segmentation

The layer segmentation is performed first using an established approach with a modified graph-theoretic method for mouse SD-OCT images [17]. Within the approach, multiple surfaces in SD-OCT scans are segmented using the graph-theoretic approach [47, 48] for global optimal solution. Similar to human approaches [48–50], the outer surfaces namely ILM, ELM, the junction of the inner and outer segments, RPE were segmented first, followed by the segmentation of the inner surfaces, NF+GC+IPL, INL, and OPL. A multi-resolution approach [49, 50] was also used in each these steps in order to reduce computation time. Meanwhile, some changes were made in order to accommodate the differences between human and mouse retinae. The cost functions used here consisted of on-surface cost term [49, 50] derived from gradient images computed using Gaussian derivative filters [17].

Fig. 4.2 shows an example result of the layer segmentation.



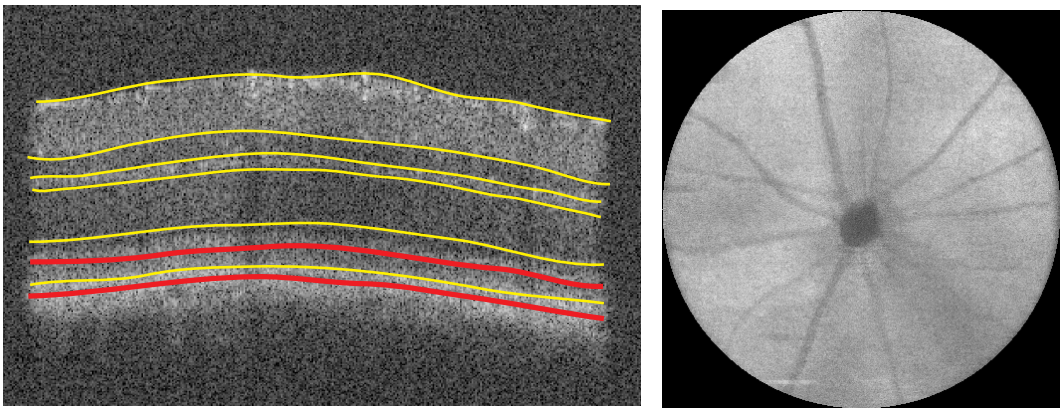
(a) (b)

Figure 4.2: Simple illustration of intraretinal layer segmentation. (a) A sample b-scan from mouse SD-OCT volume. (b) Same scan with segmented surfaces marked.

4.2.2 Projection image creation

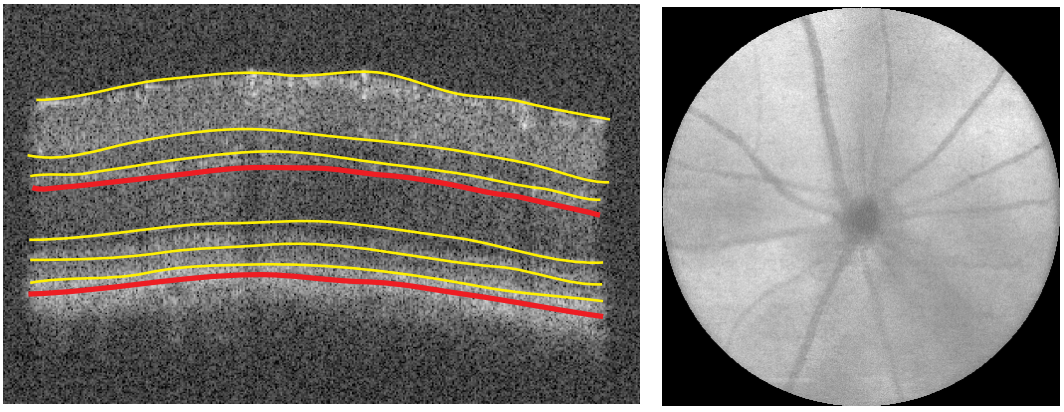
Since we are segmenting vessels using shadows of them projected in intraretinal layers, creating projection image using retinal layers below blood vessels would give a clearer visualization of vessels in the image. In baseline approach, a projection image from the volume is created combining IS and RPE layers in accordance to Niemeijer's work [9]. Fig. 4.3 shows an example of layers used for generating the projection image and the corresponding projection image generated from the layers. On the other hand, for single projection image approach, we create one overall 2D projection image using all of layers from ONL to RPE in SD-OCT volumes. This combination of layers was experimentally chosen, as it would create a projection image with best visually separable vessels. Fig. 4.4 is an example of layers used in single projection approach and the corresponding projection image.

However, selection of layers for projecting them is fairly subjective. Each voxel in the retinal part of the SD-OCT image potentially contains information of blood vessels, no matter if it is a part of the vessel structure or a part of the silhouette of a vessel. Using 3D features from all related pixels in the retina would ideally solve the problem. However, 3D feature data generated from the process is too



(a) (b)

Figure 4.3: Projection image creation for baseline approach. (a) Layers used for creating projection image are between red marked surfaces. (b) Projection image generated from layers marked in (a).



(a) (b)

Figure 4.4: Projection image creation for single projection approach. (a) Layers used for creating projection image are between red marked surfaces. (b) Projection image generated from layers marked in (a).

large for classification. In order to classify the SD-OCT images without losing more 3D information, we get distinguishable layers in the 3D volumes. From SD-OCT scans of the image, more than 7 intraretinal layers can be detected using graph-based surface segmentation. Since the intensity of voxels of the same column (z direction) in the same layer is likely to be similar, averaging each intraretinal layer into a single projection image will reduce amount of voxels we need to deal with. As a result, for all layers approach, we adopt all layers and create 7 projection images for each location in the 2D projection image. Fig. 4.5 shows the layer projection images we can get from the mouse SD-OCT images. Fig. 4.5(h) is an alignment of the images. Conceptually we generate a 400×400×7 volume instead of 400×400×1024 to find blood vessels.

4.2.3 Feature generation

Given single projection image or set of projection images, different filters are applied to get a feature vector for each pixel in each projection image. The features include Gaussian-based features, Hessian analysis based features, and Gabor features from projection images. For each of three approaches, we use some different features. How different sets of features are generated are shown below.

Gaussian filter banks including Gaussian derivatives can be used as detecting vesselness. It is also adopted in the human approach [9]. The Gaussian-based features would apply Gaussian filters from 0 to 2$^{\text{nd}}$ order derivatives (L, L_x, L_y, L_{xy}, L_{xx}, L_{yy}) with scale σ of 1, 2, 4, 8, 16 to get 30 dimension feature vectors for each projection image.

For Hessian analysis based features, method based on point detection in [8] is used for creating filters. 2$^{\text{nd}}$ order derivative Gaussian filters are used as L_{xy}, L_{xx}, and L_{yy} kernels of Hessian matrices. The Hessian-based filters are generated by combining all three kernels with rotations. A collection of σ of 2, 3, 4, 5 and orientations θ from 0$^{\text{o}}$ to 360$^{\text{o}}$ with 15$^{\text{o}}$ are used to generate a total of 48 features.

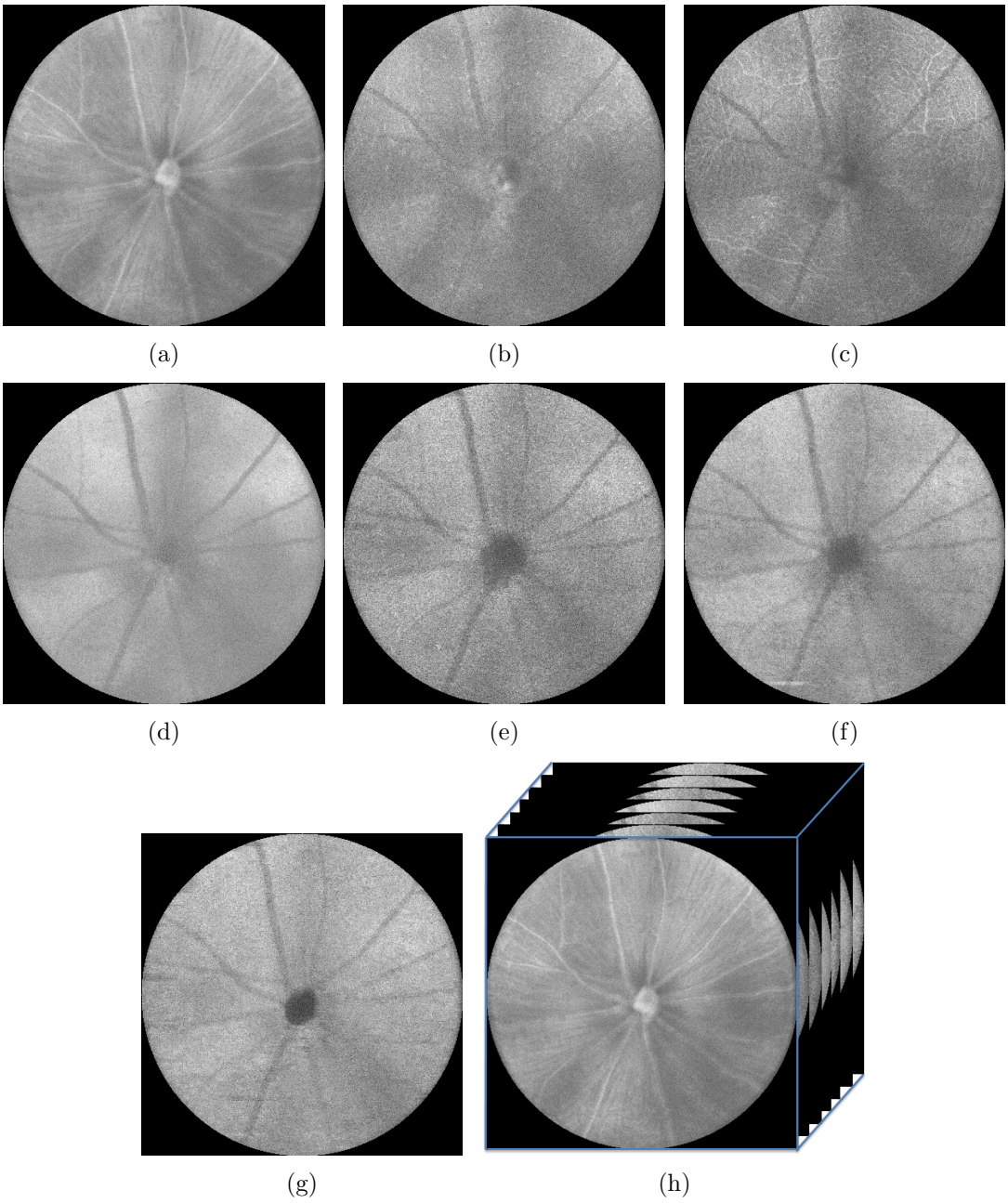


Figure 4.5: Example of all the layer projection images obtained for all layers approach. (h) shows the overall layout of them.

Texture-based features such as Gabor features can also be used for identifying vessels [44]. For Gabor features, filters with standard deviation in the filter σ of 1, 2, 4, 6 and orientations θ from 0^o to 180^o with 15^o interval will generate 64 Gabor-energy features [51]. The mean and variance from symmetric Gabor filters are also used as features. In total, there will be 192 Gabor features for one projection image.

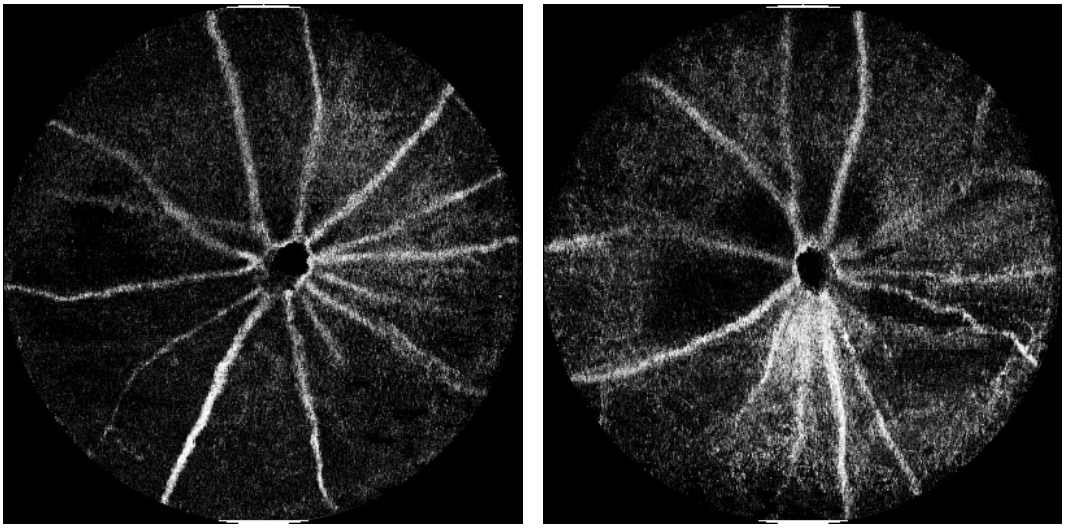
In the baseline method, only features generated using Gaussian filter banks are used for the process according to [9]. This gives some very efficient and minimal number of features. Single projection method, on the other hand, uses all features from Gaussian, Hessian, and Gabor-based features. A total of 270 features is used for vessel segmentations. For all layers method, 7 projection images are used for generating features. Since there are too many features generated from Gabor set, nearly 2,000 features need to be applied if using them. Hence, we would only apply Gaussian and Hessian-based filters to generate $(30 + 48) \times 7 = 546$ features.

Each feature obtained is then normalized with zero mean and unit variance before the next step.

4.2.4 Feature selection and pixel classification

For feature selection and pixel classification, CUDA-based k-NN classifier as described in Chapter 3 with soft labels is used for identifying pixels as a part of blood vessel or background. In the baseline approach, we don't perform feature selection because it already has a small enough feature set. For single projection and all layers set, feature selections are used.

A feature selection approach is used because some features obtained are helpful for identifying the vessels, while many of them are not or even detrimental. Meanwhile, k-NN is not able to weigh the features and automatically select more useful ones. If we use all features for the classification, worse results than baseline approach can be generated. Fig. 4.6 shows two examples of pixel classification using all obtained features in single projection approach. Therefore, we use feature selection to select a



(a) (b)

Figure 4.6: Example segmentation results using all obtained features.

smaller and more helpful features for classifications. Here we use a simple yet effective model called sequential forward selection by A. Waynet Whitney [52]. The basic idea is to perform a "bottom up" greedily search for the best features.

To select the best features, a training set is needed. The training set is first divided into two subsets, one as training instances and the other as testing. We then resample them for faster speed. After feature set is selected, training set for classifying vessels is randomly sampled and only a subset is used because of the limitation in GPU memory for using CUDA k-NN. k-NN classifier with soft labels is then used to generate grey-scale probability images.

4.3 Experimental methods

For validation of the method, all procedures involving mice were approved by the Animal Care and Use Committee at the University of Iowa, and complied with the ARVO statement for the Use of Animals in Ophthalmic and Vision Research. Specifically, we use 20 SD-OCT images all from right eyes of 20 mice from Bioptigen scanner. They are 400×400×1024 in dimension and 3.5×3.5×1.53 μm size per pixel.

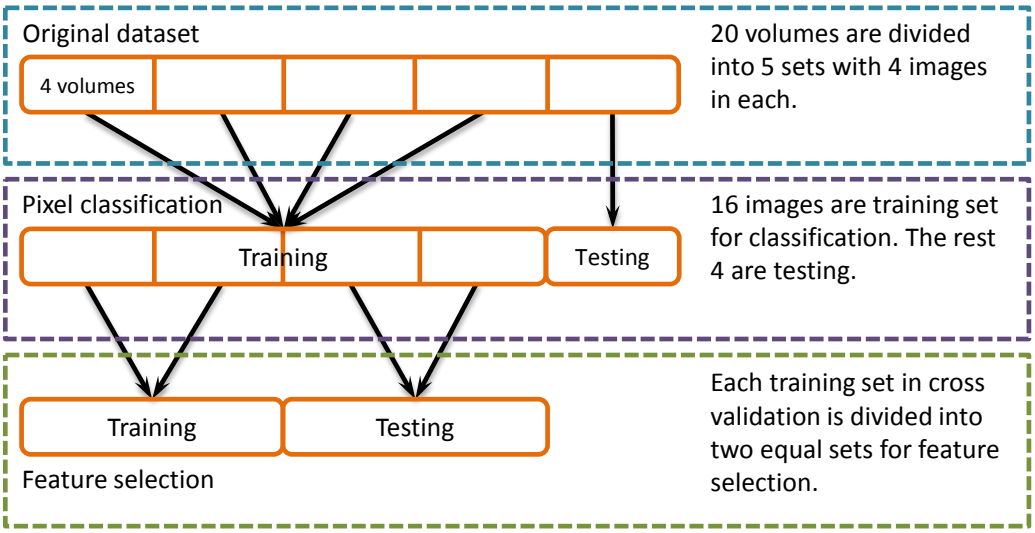


Figure 4.7: The cross-validation model.

The 20 images are divided into 5 sets with 4 imagesets in each. Then a cross validation method is used for the process. In feature selection part, rules of cross validation is also applied. The same 16 imagesets for training for k-NN are divided into 8 and 8 imagesets to find best features. A total number of 20 features are selected, and they are applied for k-NN classification in single projection and all layers approaches. An illustration of the process is shown in Fig. 4.7.

Here we use $k = 31$ according to the approach in human subjects [9]. Receiver Operating Characteristic (ROC) curve is obtained using different thresholds ($i = 0, 1, 2, ..., k$) of soft labeled image to binary image. For the imagesets we test, volumes are processed using the three methods we proposed earlier. Pixels in the 2D projection images are manually labeled as "vessel" or "non-vessel". Information from each of the layers in the images as well as the overall best single projection image are used for manually labeling the location of vessels.

We use a subset of 100,000 training and 100,000 testing points in feature selection for speed performance. For k-NN classifier, a subset of training data of 200,000 points are used as reference points. Area under curve (AUC) of ROC curves are computed

Table 4.1: First 10 features selected in feature selection for single projection † and all layers approach ‡.

Phase	Method used	Features selected
1	Single projection	31, 37, 77, 29, 12, 16, 33, 28, 25, 46, 20
	All layers	171, 237, 45, 39, 326, 321, 278, 179, 185, 30
2	Single projection	30, 24, 72, 77, 13, 36, 38, 27, 26, 31
	All layers	174, 228, 36, 30, 180, 330, 210, 263, 154, 276
3	Single projection	31, 25, 26, 71, 28, 39, 44, 77, 21, 40
	All layers	175, 25, 265, 31, 280, 274, 182, 28, 399, 282
4	Single projection	27, 33, 24, 269, 31, 34, 64, 19, 55, 14
	All layers	177, 27, 231, 45, 517, 327, 286, 160, 6, 306
5	Single projection	31, 25, 34, 77, 12, 40, 27, 37, 26, 13
	All layers	182, 32, 272, 38, 237, 326, 29, 269, 276, 191

† In single projection approach, features with indices of 0 – 47 are Hessian analysis based features, 48 – 77 are Gaussian-based features, and 78 – 269 are Gabor features.

‡ In all layers approach, indices of 0 – 335 are Hessian-based features, and 336 – 545 are Gaussian-based features.

to evaluate the segmentation results. The AUC between pairs of ROC curves of approaches we use are compared using the bootstrapping method by Carpenter and Bithell [53]. For the approach, we use pROC package in R [54], and p-values for significance is set as 0.05.

4.4 Results

For feature selection, it takes around 2 days for finding best feature combination in single projection method, and 4 days in terms of all layers method. The features chosen in the process are shown in Table 4.1. Only the first 10 selected features are shown in the table. As we can see, in cross validation, the features selected are very different in each fold.

In k-NN method, each of the dataset would take around 20 seconds to segment. ROC curves of all four methods are plotted in Fig. 4.8. In every set of projection images, the method using projection images from all layers gets a best result. The

Table 4.2: Comparison of AUC for three approaches.

	Baseline	Single projection	All layers
AUC	0.878	0.908	0.928

AUC results are in Table 4.2. Under bootstrap-based statistical tests, each pair of the methods have statistically significant difference with $p < 0.05$.

Example results can be seen in Fig. 4.9. Even without any statistical analysis, method using all layers generates the best overall vessel image. It has cleaner background, and segmented blood vessels are clear and continuous. For baseline approach, not only are more non-vessels segmented in the background, some vessels cannot be successfully segmented compared with the other two approaches.

4.5 Discussion

There are three approaches proposed in this work for segmenting mouse retinal vessels. Our results show that with baseline approach adopted from prior work, some of the vessels cannot be successfully segmented. Using the other two new approaches results in a significant improvement. In particular, using more layer information would offer more information from projection images and hence gives better results.

Features selected in cross-validation in feature selection part are not the same; one of the reason could because of the greedy selection without discarding mechanism. If different sets of features have been selected in different folds, features selected afterwards tend to remain different. Also, most selected features are Hessian-based.

We can also apply feature selections to the feature sets and get a satisfying feature collection within acceptable time (around 10hours$\times$5 for 5 fold cross-validation), using CUDA-based k-NN.

In future work we could use registration and fundus images of mice to improve the vessel segmentation. Better cleaning up strategies could be applied, because blob-like

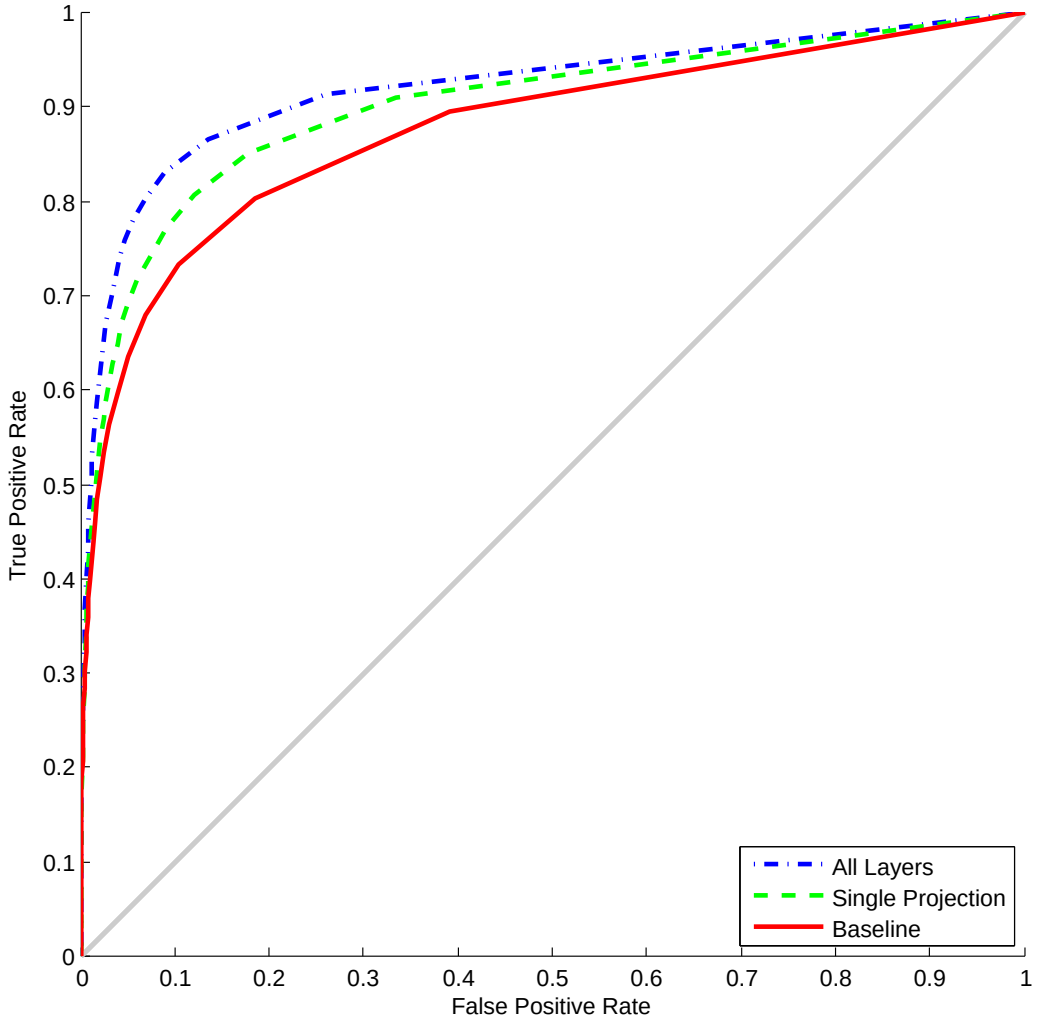


Figure 4.8: Comparison of ROC curves for baseline, single projection, and all layers approach.

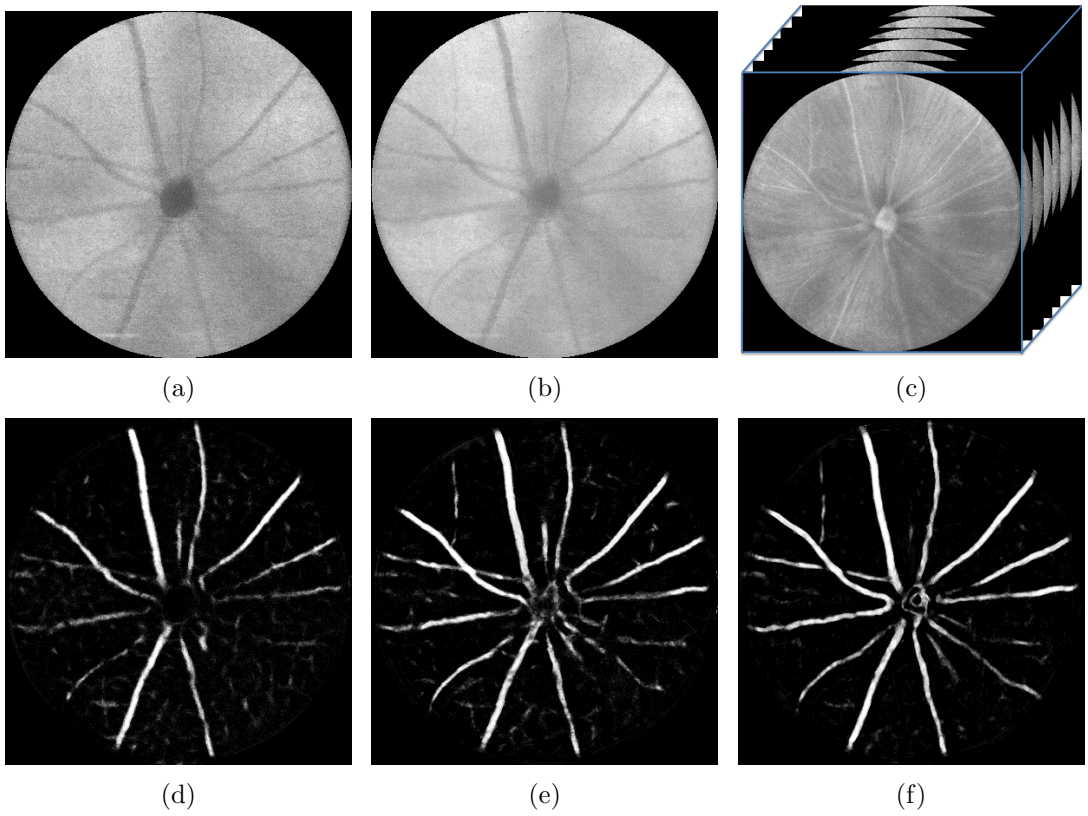


Figure 4.9: Example results of the three proposed approaches. (a) and (d) are the projection image and result from baseline approach. (b) and (e) are the corresponding for single projection approach. (c) and (f) are for all layers approach.

structures would affect results. Methods such as in [5] could also give a better result.

CHAPTER 5
CONCLUSION

In this work, a speed comparison between GPU-based CUDA k-NN implementation and the ANN implementation has been tested on three sets of medical imaging data. The results show that with higher dimensional data, CUDA-based k-NN approach could have up to two orders of magnitude of speed up. Otherwise, ANN would be a better implementation to use.

Also, based on the work of CUDA k-NN, we present two new approaches to segment vasculature in mouse retina using large set of features. They performs better than directly implementing closest previous OCT-based approach. The method using all layer projections would show overall best result with more visible vessels and higher contrast.

REFERENCES

[1] M. E. Martínez-Pérez, A. D. Hughes, A. V. Stanton, S. A. Thom, A. A. Bharath, and K. H. Parker, "Retinal blood vessel segmentation by means of scale-space analysis and region growing," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI99*. Springer, 1999, pp. 90–97.

[2] F. Zana and J.-C. Klein, "Segmentation of vessel-like patterns using mathematical morphology and curvature evaluation," *Image Processing, IEEE Transactions on*, vol. 10, no. 7, pp. 1010–1019, 2001.

[3] B. S. Lam and H. Yan, "A novel vessel segmentation algorithm for pathological retina images based on the divergence of vector fields," *Medical Imaging, IEEE Transactions on*, vol. 27, no. 2, pp. 237–246, 2008.

[4] X. Jiang and D. Mojon, "Adaptive local thresholding by verification-based multithreshold probing with application to vessel detection in retinal images," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 1, pp. 131–137, 2003.

[5] J. Staal, M. D. Abràmoff, M. Niemeijer, M. A. Viergever, and B. van Ginneken, "Ridge-based vessel segmentation in color images of the retina," *Medical Imaging, IEEE Transactions on*, vol. 23, no. 4, pp. 501–509, 2004.

[6] M. Niemeijer, J. Staal, B. van Ginneken, M. Loog, and M. D. Abramoff, "Comparative study of retinal vessel segmentation methods on a new publicly available database," in *Medical Imaging 2004*. International Society for Optics and Photonics, 2004, pp. 648–656.

[7] J. V. Soares, J. J. Leandro, R. M. Cesar, H. F. Jelinek, and M. J. Cree, "Retinal vessel segmentation using the 2-D gabor wavelet and supervised classification," *Medical Imaging, IEEE Transactions on*, vol. 25, no. 9, pp. 1214–1222, 2006.

[8] A. Klein, W. K. Renema, L. J. Oostveen, L. J. S. Kool, and C. H. Slump, "A segmentation method for stentgrafts in the abdominal aorta from ECG-gated CTA data," in *Medical Imaging*. International Society for Optics and Photonics, 2008, pp. 69 160R–69 160R.

[9] M. Niemeijer, M. K. Garvin, B. van Ginneken, M. Sonka, and M. D. Abràmoff, "Vessel segmentation in 3D spectral OCT scans of the retina," in *Proc. SPIE*, vol. 6914, 2008, p. 69141R.

[10] M. Niemeijer, M. Sonka, M. Garvin, B. van Ginneken, and M. Abràmoff, "Au-

tomated segmentation of the retinal vasculature in 3D optical coherence tomography images," *Invest Ophthalmol Vis Sci*, 2008.

[11] J. Xu, D. Tolliver, H. Ishikawa, G. Wollstein, and J. S. Schuman, "3D OCT retinal vessel segmentation based on boosting learning," in *World Congress on Medical Physics and Biomedical Engineering, September 7-12, 2009, Munich, Germany*. Springer, 2009, pp. 179–182.

[12] Z. Hu, M. Niemeijer, M. D. Abràmoff, K. Lee, and M. K. Garvin, "Automated segmentation of 3-D spectral OCT retinal blood vessels by neural canal opening false positive suppression," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2010*. Springer, 2010, pp. 33–40.

[13] D. Huang, E. A. Swanson, C. P. Lin, J. S. Schuman, W. G. Stinson, W. Chang, M. R. Hee, T. Flotte, K. Gregory, C. A. Puliafito *et al.*, "Optical coherence tomography," *Science*, vol. 254, no. 5035, pp. 1178–1181, 1991.

[14] H. Wehbe, M. Ruggeri, S. Jiao, G. Gregori, and C. A. Puliafito, "Automatic retinal blood flow calculation using spectral domain optical coherence tomography," in *Biomedical Optics (BiOS) 2008*. International Society for Optics and Photonics, 2008, pp. 68 470I–68 470I.

[15] G. Huber, S. C. Beck, C. Grimm, A. Sahaboglu-Tekgoz, F. Paquet-Durand, A. Wenzel, P. Humphries, T. M. Redmond, M. W. Seeliger, and M. D. Fischer, "Spectral domain optical coherence tomography in mouse models of retinal degeneration," *Investigative ophthalmology & visual science*, vol. 50, no. 12, pp. 5888–5895, 2009.

[16] M. E. Pennesi, K. V. Michaels, S. S. Magee, A. Maricle, S. P. Davin, A. K. Garg, M. J. Gale, D. C. Tu, Y. Wen, L. R. Erker *et al.*, "Long-term characterization of retinal degeneration in rd1 and rd10 mice using spectral domain optical coherence tomography," *Investigative Ophthalmology & Visual Science*, vol. 53, no. 8, pp. 4644–4656, 2012.

[17] B. J. Antony, M. D. Abràmoff, M. M. Harper, W. Jeong, E. H. Sohn, Y. H. Kwon, R. Kardon, and M. K. Garvin, "A combined machine-learning and graph-based framework for the segmentation of retinal surfaces in SD-OCT volumes," *Biomed. Opt. Express*, vol. 4, no. 12, pp. 2712–2728, Dec 2013.

[18] M. D. Abràmoff, M. K. Garvin, and M. Sonka, "Retinal imaging and image analysis," *Biomedical Engineering, IEEE Reviews in*, vol. 3, pp. 169–208, 2010.

[19] T. Cover and P. Hart, "Nearest neighbor pattern classification," *Information Theory, IEEE Transactions on*, vol. 13, no. 1, pp. 21–27, 1967.

[20] B. Weyn, G. van de Wouwer, A. van Daele, P. Scheunders, D. van Dyck, E. van Marck, and W. Jacob, "Automated breast tumor diagnosis and grading based on wavelet chromatin texture description," *Cytometry*, vol. 33, no. 1, pp. 32–40, 1998.

[21] M. de Bruijne, B. van Ginneken, M. A. Viergever, and W. J. Niessen, "Adapting active shape models for 3D segmentation of tubular structures in medical images," in *Information Processing in Medical Imaging*. Springer, 2003, pp. 136–147.

[22] P. Anbeek, K. L. Vincken, M. J. van Osch, R. H. Bisschops, and J. van der Grond, "Probabilistic segmentation of white matter lesions in MR imaging," *NeuroImage*, vol. 21, no. 3, pp. 1037–1044, 2004.

[23] G. P. Mazzara, R. P. Velthuizen, J. L. Pearlman, H. M. Greenberg, and H. Wagner, "Brain tumor target volume determination for radiation treatment planning through automated MRI segmentation," *International Journal of Radiation Oncology* Biology* Physics*, vol. 59, no. 1, pp. 300–312, 2004.

[24] X. Chen, X. Zhou, and S. T. Wong, "Automated segmentation, classification, and tracking of cancer cell nuclei in time-lapse microscopy," *Biomedical Engineering, IEEE Transactions on*, vol. 53, no. 4, pp. 762–766, 2006.

[25] B. J. Antony, M. D. Abràmoff, M. Sonka, Y. H. Kwon, and M. K. Garvin, "Incorporation of texture-based features in optimal graph-theoretic approach with application to the 3D segmentation of intraretinal surfaces in SD-OCT volumes," in *SPIE Medical Imaging*. International Society for Optics and Photonics, 2012, pp. 83 141G–83 141G.

[26] M. S. Miri, K. Lee, M. Niemeijer, M. D. Abràmoff, Y. H. Kwon, and M. K. Garvin, "Multimodal segmentation of optic disc and cup from stereo fundus and SD-OCT images," in *SPIE Medical Imaging*. International Society for Optics and Photonics, 2013, pp. 86 690O–86 690O.

[27] Q. Hu, M. K. Garvin, M. A. Christopher, X. Xu, T. Scheetz, and M. D. Abramoff, "Optimal filter approach for the detection of vessel bifurcations in color fundus images," in *SPIE Medical Imaging*. International Society for Optics and Photonics, 2013, pp. 866 920–866 920.

[28] J. L. Bentley, "K-d trees for semidynamic point sets," in *Proceedings of the sixth annual symposium on Computational geometry*. ACM, 1990, pp. 187–197.

[29] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu, "An

optimal algorithm for approximate nearest neighbor searching fixed dimensions," *Journal of the ACM (JACM)*, vol. 45, no. 6, pp. 891–923, 1998.

[30] V. Garcia, E. Debreuve, and M. Barlaud, "Fast k nearest neighbor search using GPU," in *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on.* IEEE, 2008, pp. 1–6.

[31] H. Yu, C. Caldwell, K. Mah, and D. Mozeg, "Coregistered FDG PET/CT-based textural characterization of head and neck cancer for radiation treatment planning," *Medical Imaging, IEEE Transactions on*, vol. 28, no. 3, pp. 374–383, 2009.

[32] K. Murphy, B. van Ginneken, A. M. Schilham, B. De Hoop, H. Gietema, and M. Prokop, "A large-scale evaluation of automatic pulmonary nodule detection in chest CT using local image features and k-nearest-neighbour classification," *Medical Image Analysis*, vol. 13, no. 5, pp. 757–770, 2009.

[33] D. R. Wilson and T. R. Martinez, "Reduction techniques for instance-based learning algorithms," *Machine learning*, vol. 38, no. 3, pp. 257–286, 2000.

[34] J. H. Friedman, J. L. Bentley, and R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time," *ACM Transactions on Mathematical Software (TOMS)*, vol. 3, no. 3, pp. 209–226, 1977.

[35] M. Bern, "Approximate closest-point queries in high dimensions," *Information Processing Letters*, vol. 45, no. 2, pp. 95–99, 1993.

[36] Q. Kuang and L. Zhao, "A practical GPU based kNN algorithm," in *International Symposium on Computer Science and Computational Technology (ISCSCT)*, 2009, pp. 151–155.

[37] V. Garcia, E. Debreuve, F. Nielsen, and M. Barlaud, "K-nearest neighbor search: Fast GPU-based implementations and application to high-dimensional feature matching," in *Image Processing (ICIP), 2010 17th IEEE International Conference on.* IEEE, 2010, pp. 3757–3760.

[38] H. KOLB, "How the retina works," *American scientist*, vol. 91, no. 1, pp. 28–35, 2003.

[39] H. Kolb, E. Fernandez, and R. Nelson, "Simple anatomy of the retina," 2005.

[40] C. Sinthanayothin, J. F. Boyce, H. L. Cook, and T. H. Williamson, "Automated localisation of the optic disc, fovea, and retinal blood vessels from digital colour

fundus images," *British Journal of Ophthalmology*, vol. 83, no. 8, pp. 902–910, 1999.

[41] C. Sinthanayothin, J. Boyce, T. Williamson, H. Cook, E. Mensah, S. Lal, and D. Usher, "Automated detection of diabetic retinopathy on digital fundus images," *Diabetic Medicine*, vol. 19, no. 2, pp. 105–112, 2002.

[42] E. Ricci and R. Perfetti, "Retinal blood vessel segmentation using line operators and support vector classification," *Medical Imaging, IEEE Transactions on*, vol. 26, no. 10, pp. 1357–1365, 2007.

[43] C. A. Lupascu, D. Tegolo, and E. Trucco, "FABC: retinal vessel segmentation using AdaBoost," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 14, no. 5, pp. 1267–1274, 2010.

[44] A. Bhuiyan, B. Nath, J. Chua, and R. Kotagiri, "Blood vessel segmentation from color retinal images using unsupervised texture classification," in *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, vol. 5. IEEE, 2007, pp. V–521.

[45] G. Kande, T. Savithri, and P. Subbaiah, "Retinal vessel segmentation using spatially weighted fuzzy c-means clustering and histogram matching," in *India Conference, 2008. INDICON 2008. Annual IEEE*, vol. 1. IEEE, 2008, pp. 1–6.

[46] D. R. Wilson and T. R. Martinez, "Instance pruning techniques," in *ICML*, vol. 97, 1997, pp. 403–411.

[47] K. Li, X. Wu, D. Z. Chen, and M. Sonka, "Optimal surface segmentation in volumetric images-a graph-theoretic approach," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 1, pp. 119–134, 2006.

[48] M. K. Garvin, M. D. Abràmoff, 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," *Medical Imaging, IEEE Transactions on*, vol. 28, no. 9, pp. 1436–1447, 2009.

[49] K. Lee, M. Niemeijer, M. K. Garvin, Y. H. Kwon, M. Sonka, and M. D. Abràmoff, "3-D segmentation of the rim and cup in spectral-domain optical coherence tomography volumes of the optic nerve head," in *SPIE Medical Imaging*. International Society for Optics and Photonics, 2009, pp. 72 622D–72 622D.

[50] B. J. Antony, M. D. Abràmoff, K. Lee, P. Sonkova, P. Gupta, Y. Kwon, M. Niemeijer, Z. Hu, and M. K. Garvin, "Automated 3D segmentation of in-

traretinal layers from optic nerve head optical coherence tomography images," in *SPIE Medical Imaging*. International Society for Optics and Photonics, 2010, pp. 76 260U–76 260U.

[51] P. Kruizinga and N. Petkov, "Nonlinear operator for oriented texture," *Image Processing, IEEE Transactions on*, vol. 8, no. 10, pp. 1395–1407, 1999.

[52] A. W. Whitney, "A direct method of nonparametric measurement selection," *Computers, IEEE Transactions on*, vol. 100, no. 9, pp. 1100–1103, 1971.

[53] J. Carpenter and J. Bithell, "Bootstrap confidence intervals: when, which, what? A practical guide for medical statisticians," *Statistics in medicine*, vol. 19, no. 9, pp. 1141–1164, 2000.

[54] X. Robin, N. Turck, A. Hainard, N. Tiberti, F. Lisacek, J.-C. Sanchez, and M. Müller, "pROC: an open-source package for R and S+ to analyze and compare ROC curves," *BMC Bioinformatics*, vol. 12, no. 1, p. 77, 2011.